DOCUMENTATION OF THE SAPRC CHEMICAL MECHANISM MODELING SOFTWARE AND FILES

Draft Report to the

California Air Resources Board
Contract No. 11-761

October 7, 2020

By

William P. L. Carter
College of Engineering, Center for Environmental Research and Technology
University of California, Riverside, California 92521

ABSTRACT

This document describes a software system and files used to carry out computer kinetic model simulations of reacting chemical systems with no mixing or transport. The system was developed for modeling atmospheric chemical reactions for simulating smog chamber or static reactor systems, simulations of ambient systems using box models, and calculation of MIR and other ozone reactivity scales, but can be used for other applications. It was used in the development of the SAPRC chemical mechanisms, the evaluation and testing of these and other mechanisms against results of environmental chamber experiments, and to carry out box model ambient air simulations for calculating VOC reactivity scales and other purposes. It consists of a mechanism preparation program with a highly flexible format for inputting reactions and model species specifications that is used to prepare the programs for model simulations, programs for converting model-independent simulation input to input for specific models, programs for running model simulations, and programs to aid in preparation of model simulation input and output. The present document describes the programs, files, and procedures needed to run model simulations using different mechanisms. The availability of additional information concerning the programs and mechanisms used in these files, and how to obtain updated information, is discussed.

ACKNOWLEDGEMENTS AND DISCLAIMERS

This work is funded in part by the California Air Resources Board through Contract 11-761. This is still a draft and has not been reviewed by the California Air Resources Board or any other persons or organizations. Mention of trade names or software products does not constitute endorsement or recommendation for use.

Please contact the author at carter@cert.ucr.edu to comment on errors, omissions, or areas where the content needs to be clarified or improved. Updated versions of this document may be available at http://www.cert.ucr.edu/~carter/SAPRC/SAPRCfiles.htm.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF EXAMPLES

1. INTRODUCTION

Chemical kinetic computer modeling is a powerful tool for the quantitative understanding and predictions of complex chemical systems. Such models involve the prediction of how chemical systems evolve over time, given the initial concentrations or input rates and the rates of chemical transformation or loss for all the chemical species involved. Using such models requires at a minimum a chemical mechanism that predicts the rates of chemical transformation for each of the chemical species given the concentrations of all the species in the model and the relevant environmental conditions and appropriate software that can solve the set of ordinary differential equations that correspond to the kinetic equations that are derived from the mechanism. The most useful software for this purpose depends on the modeling application and can also depend on the type of chemical mechanism that is used.

One important application of computer kinetic modeling is airshed modeling, where transformations of atmospheric pollutants over time are calculated given the initial and emitted concentrations, environmental conditions, and a gas-phase atmospheric chemical mechanism designed for this purpose. Such models are necessary for calculations of levels of secondary pollutants, i.e., pollutants such as ozone or secondary particle matter that are not emitted directly but are formed from the chemical reactions of the emitted pollutants. Often secondary pollutants a greater air quality concern than the primary pollutants that are actually emitted, and such models are necessary to understand or predict how emissions changes will affect the concentrations of secondary pollutants of concern. For example, ground level ozone concentrations exceed health-based air quality standards in many areas of the world, yet ozone is not emitted directly but is formed in a complex series of reactions involving emitted volatile organic compounds (VOCs) and oxides of nitrogen (NO_x) and sunlight. Use of air quality models with chemical mechanisms for ozone formation from VOCs and NO_x are necessary in order to develop and investigate effective control strategies for VOCs and NO_x for reducing ambient ozone levels.

Calculation or prediction of air quality in actual environments generally involves 3-dimensional air quality models that divide the domain being modeled into thousands of cells representing the various locations and altitudes, and represents transport of materials between the cells due to diffusion and meteorology as well as emissions of pollutants and chemical transformations within the cells. Examples of current open-source models of this type are the CMAQ model developed and maintained the U.S. EPA (http://www.cmaq-model.org/) and the CAMx model developed and maintained by Environ Corporation (http://www.camx.com). Although these 3-D models provide the best means for predicting how pollutants vary with time and location in actual airshed scenarios, because of their large computer resource and data input demands, complexity of their output, and relatively high cost and expertise required for proper usage, they are not necessarily optimum for all air quality modeling applications.

In many cases, much simpler "box models", which represent the airshed or experiment being modeled as a single well-mixed container with the chemical species, are sufficient or even preferable for the model application. Examples include:

- <u>Basic chemical mechanism development and analysis</u>. Simple box models provide the most practical means to develop new mechanisms, investigate or develop their features, evaluate which portions are or are not important, explore options for representing complex or uncertain processes, and design experiments that may be useful for reducing uncertainties. Such models focus only on the inputs and predictions that are directly relevant to modeling the chemical transformations, and complexities such as mixing and transport do not need to be considered.
- <u>Chemical mechanism development and evaluation</u>. Chemical mechanisms for modeling secondary air pollutants such as ozone are highly complex and have many uncertainties, and their

predictive capabilities need to be evaluated by comparing their predictions against experimental observations. Box models provide the most straightforward way to test effects of alternative assumptions or treatments of chemical uncertainties on model predictions under well-characterized conditions. In addition, the most straightforward test of the predictive capabilities of chemical mechanisms independent of other model uncertainties is modeling experiments in environmental chambers, where uncertainties not related to the chemical mechanism can be minimized, and where use of box models is sufficient. Box model predictions can be used not only to test the consistency of the mechanism with the data, but also for designing which experiments may be potentially the most useful for reducing uncertainties

- Modeling with large mechanisms and VOC Reactivity Assessment. Many hundreds of compounds are emitted into the atmosphere and even if only their most important oxidation processes are represented explicitly in the chemical mechanism then many thousands of reactions and species may need to be represented. Because of their large computational demands, 3-D models generally use condensed chemical mechanisms that involve a much more limited number of model species and reactions. It is much more practical to implement large mechanisms in box models and use such models to make predictions that require the additional chemical detail or to evaluate the effects of using condensed mechanisms on predictions of interest. For example, box models with large mechanisms have been used for developing reactivity scales that measure the relative ozone or other air quality impacts of the many hundreds of different types of VOCs that are emitted into the atmosphere. Examples are the maximum incremental reactivity (MIR) scales developed using the detailed SAPRC chemical mechanism and box models for 39 urban areas in the United States (Carter, 1994a) and the photochemical ozone formation potential (POCP) reactivity scales using a Master Chemical Mechanism and single cell models for various trajectories in Europe (Derwent et al, 1998).

A number of software packages exist for conducting single cell or box model chemical kinetic simulations, though most consist primarily of the model simulation program itself, and have only limited or no associated software tools to aid or simplify preparing model inputs or analyzing or presenting the results. One exception is the SAPRC photochemical modeling system that has been used for the development and evaluation of the SAPRC atmospheric chemical mechanisms and the derivation of the MIR and other ozone reactivity scales that are widely used (Carter, 1990, 1994a, 2000, 2010a,b). This software system has been developed and utilized continuously starting around the mid-1970's (Carter et al, 1979), and features have been added or improved during the almost 40 years to aid in the chemical mechanism development and VOC reactivity assessment research. It currently includes the following features:

- A flexible chemical mechanism compiler that converts a mechanism listing into computer source code for kinetic model simulations. This compiler can be used for explicit mechanisms where each reaction represents an elementary process and also for lumped mechanisms with variable kinetic and product yield parameters that depend on the compounds being represented. Input can be taken from multiple files to allow for modular mechanism construction if desired. It permits kinetic parameter input in the many formats used in current atmospheric chemical mechanisms, including the special parameters used in versions of the SAPRC-07 mechanism that are not supported by most other mechanism compilers. It also allows the use of the steady state approximation for some species where this may be appropriate, which can reduce the computer demands for simulations with the mechanism.
- Input files implementing various current atmospheric chemical mechanisms, including current and recent versions of the detailed and condensed SAPRC mechanisms.
- A model simulation program that can be used not only for simulations of reactions in static reactors or environmental chamber experiments, but also for dynamic single cell simulations with

time-varying emissions, dilution, mixing volumes, temperatures, light intensities, and representation of pollutants entrained from aloft for the purpose of approximating ambient conditions with a single cell model. The latter capability is used to conduct reactivity scale simulations using simplified models of various urban areas around the United States (Carter, 1994a,b).

- Input files for simulating the hundreds of environmental chamber experiments used to develop the SAPRC chemical mechanisms and also files with the results of these experiments for comparison with model predictions. This includes files giving the spectra and intensities of the light sources employed and the parameters used for the models for chamber effects used when evaluating the mechanisms with these data.
- Input files for the simulations of the ambient scenarios used to calculate the MIR and other Carter (1994a, 2000, 2010a,b) reactivity scales and also for the ambient scenarios used as test cases developed for mechanism comparison purposes. The latter were used when developing condensed versions of the SAPRC mechanisms based on the detailed versions (Carter, 2010c,d).
- Programs and input files needed to implement the "adjustable parameter" version of the detailed SAPRC mechanism (Carter, 2000, 2010a) for the purpose of modeling chamber experiments or conducting box model ambient simulations. This version of the mechanism represents the individual VOCs that are not in the base mechanism using generalized reactions with adjustable parameters whose values depend on the compound (or mixture of compounds) being represented. This is the version of the mechanism that us used for VOC reactivity assessment and also evaluating the mechanism when modeling the chamber experiments. The programs and files derive the parameter values in the format needed by the model simulation programs based on the compounds being simulated.
- Programs and Excel templates that can be used to manage the outputs of the simulations for various purposes. These include producing plots of the simulation results by themselves or against results of chamber experiments or other simulations, summarizing model performance statistics in simulations of large numbers of chamber experiments, and compiling results of ambient simulations, including VOC reactivity scales. The model simulation program can produce results in a CSV format that can be loaded directly into spreadsheets, or that can store the results in ASCII files that can be read by various utility programs for producing desired plots and summaries of results. (A discussion of the Excel templates is beyond the scope of the present document.)

The programs, files, and examples have been available for some time at the SAPRC mechanisms web site at http://www.cert.ucr.edu/~carter/SAPRC/SAPRCfiles.htm (See Section 4), and have been downloaded and adapted by various research groups for their use. However, until now the programs have not been documented beyond providing instructions for installation and comments in various example batch files for execution and various input files.

This document provides an initial documentation of the SAPRC model simulation programs and associated utilities, files and procedures. It does not cover all the features of the software, and does not give a complete discussion of how to use the software with adjustable parameter mechanisms such as SAPRC-07 and SAPRC-11. Additional related programs, files, and features may be discussed in future version of this document, or in separate documents (see Section 4).

2. SYSTEM OVERVIEW

2.1. Introduction

The SAPRC model simulation software system consists primarily of a series of interrelated Fortran programs and files that were developed over almost a 40-year period for chemical mechanism development, evaluation, and box modeling applications research. Later, Excel spreadsheet templates and files were developed to aid in the use of these programs. As such, this software system has a number of components and capabilities, making it relatively complex compared to other kinetic modeling software systems are available. However, most of the components and capabilities are not needed for relatively simple modeling applications, and some components are only relevant to special applications that may be of interest to only a few potential users. In the overview discussion we will consider the minimal system for basic modeling applications that do not require most of the special feature and special types of special applications that require use of additional features and programs. These are discussed below.

Regardless of the application, use of this software involves three separate steps: (1) mechanism preparation, (2) model simulation, and (3) display and analysis of the simulation results. Mechanism preparation consists of running program(s) that take inputs specifying the reactions, rate constants, and applicable mechanistic parameters and producing the computer code and data files needed to run the simulation. The model simulation step consists of using the prepared mechanism to run the simulation given inputs specifying the initial conditions and relevant parameters for the system being modeled. This could represent a model of a well-mixed air parcel in a natural or polluted atmosphere, a model of a laboratory or environmental chamber experiment, or an idealized or simplified model of a reacting air parcel in the atmosphere. The analysis of the results consists of viewing, plotting, or processing the concentration-time data output in the model simulation, as needed for the specific modeling application.

Some software packages use a single program to combine the mechanism preparation and model simulation step, but in this system the mechanism preparation must be carried out as a separate step. This is because in this system the model simulations are carried out using mechanism-specific model simulation programs that are prepared as part of the mechanism preparation process. Once the mechanism is prepared it can be used for simulations of any number of different scenarios or experiments. Mechanism-specific modeling programs can conduct the simulations with much greater efficiency than general modeling program, making them more suitable for applications where multiple simulations using a given mechanism are required. Mechanisms can also optionally be prepared with named mechanistic parameters that can be specified as inputs during the model simulations to allow effects of changes of these parameters to be calculated without having to prepare the mechanism. The use of named, variable mechanistic parameters also allows use of mechanisms whose parameter values depend on the compounds being represented, a feature that is required for most modeling applications using the SAPRC detailed mechanisms (Carter, 2000, 2010a).

The analysis of the simulation results depends on the specific application, but in most cases the last stage involves loading the results into spreadsheet programs for plotting or analysis. Some of the applications require processing the data or collecting data from multiple simulations prior to importing the data into spreadsheets, and programs that can be used for these purposes are included in the distribution. These are discussed below in conjunction with the discussion of the use of the programs for specific purposes.

The types of modeling applications where this software can be used are summarized are summarized as follows. Each of these are discussed in detail in the following sections.

- Minimal System for Basic Application. A basic model application consists of use of a single chemical mechanism of any size with no adjustable parameters to conduct static box model simulations with a constant light source if it has photolysis reactions, with the output saved to CSV files that can be read into spreadsheets for plotting or analysis. This requires the least number of programs and program options

- Minimal System for Ambient Simulations. Although the most realistic ambient simulations require 3-D, multi-cell models, this box modeling software can do single cell ambient simulations with time varying solar light intensity and time varying dilution, pollutant emissions, inversion heights (affecting box volume and entrainment of pollutants aloft) and reaction conditions such as temperature and humidity. This requires several additional steps and input options when preparing mechanisms and a number of additional input types for the model simulations. Again, the simulation results can be saved to CSV files that can be read into spreadsheets for plotting or analysis.

- Calculations with Multiple Mechanisms. Simulations with multiple mechanisms are useful for mechanism development, evaluation, and comparison. These applications consist of running basic or ambient simulations using different chemical mechanism or using similar mechanisms with different default parameters. The mechanism preparation step is the same as above, but more than one mechanism and mechanism-specific modeling program is prepared. The mechanism files and programs may be in the same or different locations or folders (which in general are different from those used for the simulation inputs and outputs) but if they are in the same folder they must have different names. A mechanism parameter file specifying the name and location of the mechanism simulation program and files must be specified when running the model simulations, and the input file for the simulation may have to be pre-processed by a "lumping" program if the different mechanisms use different model species names or lumping methods to represent the compounds being modeled. The results are stored in separate locations depending on the names of the model parameter file and the simulation input file, and the distribution includes programs that read these files given these two names and formats them for input into the spreadsheet program for output or analysis.

- Simulations of Laboratory or Environmental Chamber Experiments or Measurement Data. Such applications consist of model simulations of previous experiments or measurement campaigns for the purpose of seeing how well the model can simulate the observations. The mechanism preparation process is generally the same discussed above, except that the mechanism must include appropriate provisions for background effects such as wall losses, if applicable. The simulation process is also similar, except that the input files must be carefully prepared to be consistent with the conditions of the experiments, and additional inputs may be needed for characterization and light spectrum data if applicable. The processing of the results includes summarizing or loading experimental measurement data into the spreadsheets for comparisons with the simulated results. The distribution includes modeling input and experimental data files for all the experiments used in most recent SAPRC mechanism evaluations (Carter, 2010a,b; Carter and Heo, 2012, 2013; Carter et al 2013, and programs for inputting experimental data into spreadsheets for comparison with modeling results.

- Incremental Reactivity Simulations. The software and files in this distribution can also be used to calculate the MIR and ozone reactivity scales using the detailed SAPRC mechanisms (Carter, 1994a, 2000, 2010a,b,e).

2.2. Installation and Operating System Issues

The distributed programs in the SAPRC modeling system are all written in Fortran and compiled and tested using the GNU Fortran compiler "gfortran", running under a Windows operating system. The GNU Fortran compiler is also available for Unix systems, but all the executables would have to be recompiled, many of the programs would need small modifications[1], and all the batch files used for building the programs have to be modified to work under Unix systems. Most of the discussion in this document assumes the programs are installed on a Windows system, but experienced Unix and Fortran programmers should be able to make the necessary modifications to the build files and programs based on information given in the program details sections of this document.

2.2.1. Installing the files.

The SAPRC modeling software and mechanism files are available at http://www.cert.ucr.edu/~carter/SAPRC/SAPRCfiles.htm, or do an internet search for "saprc modeling programs" in case they have been moved. The distributed files are contained in .ZIP files and are organized into a directory structure, and should be extracted in a way that preserves this structure. (Other directory structures could be used, but changes would have to be made to many of the parameter files discussed in Section 3.1.) If the distributed programs are to be used with the files, they must be extracted with the subfolders in all .ZIP files sharing the same root folder. The directory structure for the distributed files is shown on Table 1, which also indicates the distributed zip files and the types of files in each location.

The suggested installation procedure is to copy the .ZIP files to be used to the root directory where they are to be installed, along with the distributed UNZIP program. Then open a DOS window and go to that folder and run UNZIP for each of the files (e.g., UNZIP MECH; UNZIP PGMS, etc.), which will extract the files to the proper folders. (If you are overwriting previously downloaded files with a newer version, use the "-o" option, e.g., "UNZIP -o MECH".) The ZIP files are no then longer needed and can be deleted.

2.2.2. Installing the GNU Fortran (gfortran)

In order for the distributed executable programs to prepare mechanisms conduct simulations, it is necessary that the GNU MinGW system and its associated GNU fortran compiler, gfortran be installed. This is because the distributed executables requires components of MinGW and gfortran in order to run, and changing the mechanisms requires that the simulation program be re-compiled. Installing the compiler will also allow you to modify the distributed Fortran programs, which are all compatible with gfortran. The location of the MinGW binaries need to be on the path in order for the programs to run or compile. Normally this is C:\MinGW\bin, but it does not have to be there.

GNU Fortran is public domain software, and the MinGW version that was used to compile these programs and it is recommended that this version be used. MinGW with gfortran can be obtained from the GNU web site at http://gcc.gnu.org/wiki/GFortranBinaries#Windows. Based on our tests on a Windows7 b4-bit laptop, using the latest MingGW graphical installer described in the Graphical User Interface Installer section at http://www.mingw.org/wiki/Getting_Started is a convenient way to get a version of gfortran for using the SAPRC programs described herein. Note that the tests were run without using the XP mode for Windows7 64-bit, which indicates using the XP mode for Windows7 64-bit computers is not

---

[1] Most of the changes would involve the use of the "\" character to indicate directories and in a few cases the commands used in system calls would have to be modified. Programs that use system calls are indicated in the program documentation details.

Table 1. Directory structure for distributed files, indicating the types of files at the locations.

Distribution	Directory structure			File Types
MECH.ZIP (mechanism files)	MECH	SAPRC99		*Mech*.PRP, .MOD, .PRP, .EXE, *RxnType*.RXN *PhotType*.PHF, *LightModel*.KZS, .FZS *.LPC, *.LCC, *.GNA, *.LPM, KZSPREP.BAT, PHF.LIS,
		SAPRC07		
		SAPRC07T		
		SAPRC11		
TESTCALC.ZIP (files to run mechanism test calculations)	TESTCALC	(root folder)		*MechID*.PRM, *.BAT, TESTCALCS.XLS
		INPFILES		*CalcID*.INP
		CDTFILES	SAPRC07	*CalcID*.CDA
			(other mech's)	
CHAMCALC.ZIP (all files except has INP, VSA and GDT files for only selected runs) CHAMEXPT.ZIP (has only the INP, VSA and GDT files for all experiments)	CHAMCALC	(root folder)		*MechID*.PRM, CHAMEXPT.XLS, *.XLS, *.BAT
		INPFILES		*RunID*.INP, *RunID*.VSA
		CHAR	LIGHT	*LightSource*.SDR
			SAPRC07	*ChamberSet*.CHR
			(other mech's)	
		CHDFILES		*RunID*.GDT
		CDTFILES	SAPRC07	*RunID*.CDA
			(other mech's)	
PGMS.ZIP (executable programs and program source files)	PGMS	(root folder)		*.EXE, lib*.DLL (if MinGW not installed)
		SOURCE	(root folder)	SUBS.LIB, BLDALL.BAT, ALLSOURCE.BAT
			INT	INT.LIB, *.FOR, BLDALL.BAT
			SUBS	*.FOR, BLDALL.BAT
			UTILPGMS	
			CHAMPGMS	
			INTUTIL	
			LMPPGMS	
			LMPSUBS	
			LUMPGEN	
			LUMPINT	
			PHK	
			PREP	
			REACTCAL	
	TEMPLATES			*.XLT
REACT.ZIP (files for reactivity calculations)	REACT	(root folder)		*.BAT, SCENARIO.PRM, MODELING.PRM
		CDTFILES	SAPRC07	*ScenID*.CDA, REACTCAL.CDA
		CMPFILES		*Mixture*.CMP
		INPFILES		*ScenType*.INP, *LightModel*.JZS
		SAPRC07	(root folder)	TESTHC.INS, ALOFT.LMO, ARBMIX1.RXP
			CLCFILES	*ScenID*.BAS, .CLC, .CLD
			RCTFILES	*ScenID*.RCT

required (although in some cases may be necessary). Be sure to include gfortran when prompted during the installation.

Note the location where the MinGW binary files are installed. By default this is C:\MinGW\bin, but it does not have to be there. It is a good idea to have this location be in the system path, though this is not absolutely necessary if the NEWENV.BAT file is configured and used as described in the following section. The environmental settings for MinGW are discussed at http://www.mingw.org/wiki/Getting_Started.

If the user is only interested in running the programs with the existing mechanisms, and does not want to modify any programs or mechanisms, then an alternative approach is to copy the minimum number set of dynamic link library (.DLL) files to the same folder as the executables on the path. This permits the executables to run without MinGW and gfortran being installed. These are in the file MINGF.ZIP. Copying MINGF.ZIP to the root folder and running "UNZIP MINGF" will copy these three .DLL files to the PGMS folder where they will be needed. However, DON'T do this if you have MinGW and gfortran installed, or there may be an incompatibility if you install a different version that we used when compiling the programs for the distribution. If you decide to install MinGW and gfortran later, then all the .DLL files in PGMS should be deleted.

2.2.3. Running the Programs

The following discussion covers running the programs on a Windows-based PC system. The programs have been tested on Windows 98, 2000, XP, and Windows 7 systems. They will probably run under Vista or Windows 8, but that has not been tested.

In order for the programs to run, whether in a DOS window or using macros in the distributed Excel files and templates, it is necessary to configure the NEWENV.BAT file so that it gives the appropriate locations of the distributed files and the location of the gfortran binaries if they are not on the system path (see above). Example 1 shows the portion of the NEWENV.BAT file that will need to be edited, indicates the items that may need to be changed. In almost all cases, the line "SET TMPENV=" will need to be changed to the root of the distribution of files (i.e., the root of the directory structure shown on Table 1), since as distributed it references a temporary location used when testing the installation. No other edits should be needed if the binaries for MinGW and gfortran are C:\MinGW\bin, but the line "SET TMPGF=" would need to be changed if it is something different.

The distributed programs can be run in a DOS window or using macros in the example Excel spreadsheets and templates that are included with the distribution. These require that Microsoft Excel be installed and have been developed using Excel 2000 and tested using Excel 2003 and 2007. (The macros to run the model and load the data will probably not work on non-Microsoft spreadsheet programs, but this has not been tested.) One way to assure that the necessary environment exists when running the programs in a DOS window is to run NEWENV.BAT immediately after opening the DOS window, or configuring the DOS window so it runs automatically when it is opened. NEWENV.BAT is also used by the macros in the distributed Excel spreadsheets for running the calculations to assure that the necessary environment exists when the programs are run, so NEWENV.BAT needs to be configured before the macros also can be used. The Excel files and templates also need to be configured before they can be used; this is discussed in the "Excel Files for Running Calculations and Displaying Results" section, below.

The procedures to run the examples in the various distribution sets are discussed in conjunction with the descriptions of the individual programs and files, later in this document.

Example 1. Portion of NEWENV.BAT as distributed that need to be edited. The portions that may need to be changed are indicated by bold underline font.

```
:       Sets the environment for SAPRC modeling programs
:       Use:
:               NEWENV      ... Adds the needed additional elements
:                               to the present system path
:               NEWENV TEST ... Uses the minimal system path for
:                               testing the distribution
:
:       IMPORTANT! The first two commands need to be edited for your system!
:
:       Change following to root of the location of the distributed
:       files.  ("I:\TMP\SAPRC" was used for testing, but is not
:       suitable for most systems.)
@SET TMPENV=I:\TMP\SAPRC
:
:       Change the following if the location of the MinGW binaries
:       is something other than C:\MinGW\bin.  It can also be blank
:       (set to "") if this is already on the system path, but not
:       if the TEST option is used.
@SET TMPGF=C:\MinGW\bin
:
: ------------------------------------------------------------------
:       The remainder of this file should not need to be edited.
```

2.3. Conventions Used in this Document

2.3.1. Examples of Commands and Inputs

Throughout this document we will be giving examples of commands needed to run the programs with various options or contents of files read by the programs, or information output by the program. Unless noted otherwise, the following conventions will be consistently used.

- Quote characters (") are used to indicate example commands or input when they are included within the text and not on a line by themselves. The other conventions discussed below are used within the quotes. The quotes should not be used in the command or input unless indicated otherwise. An example is: 'Use the "DIR" command to get a directory listing of all files and "DIR *name*" to list a particular file that you name'.
- Text given in UPPERCASE with no special formatting (i.e., not italics and not surrounded by brackets) must be exactly as shown. Almost all of the programs are case sensitive and expect the commands and options input to them to be upper case. Windows DOS boxes are not case sensitive when giving file names (including program names), but the discussion here gives them in upper case as well.
- Text in *italics* (regardless of case) are variables indicating names or options that the user must specify. Generally the description of the command, input, or output will indicate what the variable or name refers to. File names are not case sensitive in Windows DOS systems but may be in other systems. Options or variables read by the programs are generally case sensitive; upper case is usually required for options unless indicated otherwise. Variables (e.g., species names, reaction labels, or coefficient names) can optionally be lower case or include lower case

characters. An example is "DIR *name*", where *name* is the name of a file that you want a directory listing for.

- Brackets ("[" and "]") are used to indicate optional input that can be omitted or included, depending on the user's intentions. The brackets themselves must not be included in the input. An example is "INTRUN *runID model* [*option*]", where the names *runID* and *model* must be specified, while *option* does not need to be specified, but if so is given without the brackets, e.g., "INTRUN EC340 SAPRC07 CHECK".
- Spaces must be included as shown. In some cases, such as DOS command lines and some program inputs, multiple spaces or tab characters can be used when a single space is shown, but this may not be true in all cases. This may be clarified in the discussion of specific cases, but if not use of a single space is recommended.
- The terms "folder" and "directory" are used to refer to a file location on the computer. The character "\" is used throughout this document to separate file names and folder names, as used in Windows DOS systems. If the programs are converted to run on Unix systems, then these need to be replaced by the "/" character in all cases. A discussion of considerations for converting the programs to run on Unix systems is beyond the scope of the present document.
- Much of the discussion in this document assumes that the programs and files are installed in a particular directory structure, as discussed on Table 1 in Section 2.2.1. The programs all allow for use of different file structures, but the examples will not run without modifying some of the input files as discussed later in this document (see, for example, Table 3). The root of the installation can be in any directory, but the locations of the other files relative to each other should be as given in Table 1. In this document, the locations of the installed files are given without giving the root location name. For example, if the files are installed in the root location C:\SAPRC, according to Table 1, the location of the file INT.LIB would be C:\SAPRC\PGMS\SOURCE\INT \INT.LIB. If this is referenced in the text, it will be as "PGMS\SOURCE\INT\INT.LIB". The location of the root of the distribution is not given because it depends on the user's choice when installing the files.
- Special characters other than brackets ("[" and "]") and the "\" character on Unix systems, must be entered exactly as shown unless indicated otherwise. The description of various types of variables and names indicate what special characters, if any, are acceptable in these variables or names. But these are generally not used in the examples in most cases. For example, a possible reaction string might include "*#coef* {*species1* + *species2*}", and an instance of this with specific species and coefficient names is "#2 {HCHO + HO2}".

This document includes a number of figures with example input files or portions of such files. In those cases the text is exactly as shown, though comments may be included in a separate column. In some cases special font, such as **bold** or <u>underline</u> is used for emphasis or to point out portions of the files for discussion, but the files themselves are all plain text and do not have fonts.

2.3.2. Special Terms Used

The following terms have specific meanings throughout this document that may not all be exactly the same as when they are used in other contexts. To avoid possible confusion these are defined and discussed below.

<u>Model and Model Preparation Input and Output</u>. Throughout this document, the terms "model" or "kinetic model" are used to refer to a set of reactions, model species, and parameters that are used for a particular set of model simulations. These "models" are defined in model preparation input (.PRP) files

that are described in Section 3.2.3, and the contents of these files, and files referenced by them, are referred to as "model preparation input". These models can also be thought of as chemical mechanisms, though some complex mechanisms, such as SAPRC07, may use different model preparation input files with different sets of reactions or parameters for different applications. Therefore, we do not use the term "mechanism" to describe the contents of model preparation input files. A "prepared model" is the output of the model preparation process that is required before simulations can be done using the model.

Model Species. The term "model species" is used to refer to entities within the model that refer to chemical species or entities (e.g., tracer species or chemical operators) whose concentrations can be specified and calculated. Depending on the mechanism and model used they can refer to real chemical compounds (.e.g, "O3"), groups of chemical compounds assumed to react similarly (e.g., "OLE1"), portions of molecules that are treated as if they reacted like compounds (e.g., "PAR" in Carbon Bond mechanisms), parameters that are treated as if they were compounds within the model (e.g., "HV"), counter species (e.g., "XC" for "lost carbon" in SAPRC mechanisms), or chemical operators (e.g., "xHO2" in SAPRC07, meaning "formation of HO2 following peroxy radical reactions"). They are not necessarily real chemical compounds unless a chemically explicit model is used. There can be different types of model species in terms of how they are treated by the software, as discussed in Section 3.2.5.

Reactions and Kinetic Parameters. The term "reaction" refers to a single process represented within the model by which one or more model species ("reactants") are converted to a set of other model species ("products") (which may be empty) at a rate that is given by a "rate constant" multiplied by the concentrations of all the reactants. It may refer to an elementary chemical process or group of elementary reactions that are treated in the model as if they are one reaction. The rate constant can be given explicitly in the model preparation input file or can be derived from "kinetic parameters" that are given in the file. The various types of kinetic parameters that can be used are discussed in Section 3.2.4.

Coefficients and Parameters. The term "coefficient" is used to refer to a type of constant or variable parameter defined within the model, They are most commonly used to refer to yields of product model species within a reaction, but can be used for a number of other purposes within the model as discussed in Section 3.2.6. There are two types of coefficients: "constant coefficients" whose values consist of specified real number and cannot be changed once defined, and "variable coefficients" that have both a name and a value, and which can be changed after the model is prepared by giving the name and the new value. They should not be confused with "parameters", whose names and meanings are built into the software and whose names and meanings (but not necessarily values) depend on the program but not the model being used. The names and meanings of variable coefficients can vary depending on the model used, but as discussed in Section 3.2.6 certain special coefficient names are reserved for particular purposes by the software.

Simulation, Simulation Program, and Simulation Input and Output. Throughout this document, the term "simulation" refers to the calculation of how the concentrations of the model species change with time using a prepared model. The program that does this simulation is referred to as the "simulation program", and each prepared model has its own separate simulation program that is created as part of the model preparation process. The file that contains the inputs to the simulation program, giving the initial concentrations or input rates of the model species and other parameters controlling the simulation, including non-default values for kinetic parameters or coefficients, is referred to as the "simulation input". Simulation input files are discussed in Section 3.3.1. "Simulation output" consists of files containing the concentrations of the model species as a function of time that are produced by the simulation programs.

Fortran format codes. Fortran format codes are used as a shorthand to indicate the formats of input files that are in fixed width format, i.e., where data are expected to be in certain columns. The following codes are used in this document, where "*n*" refers to the number of columns and "*m*" refers to the number of digits after a decimal point that normally (though not necessarily for input files) is used: A*n* = character data, left justified; I*n* = integer data, right justified; F*n.m* = floating point data (for input *m* can be ignored), E*n.m* = floating point data in scientific format, and *n*X = ignored columns. For example, (I5,2X, A8, F6.1) means that the first 5 columns contain an integer, the 2nd two columns are ignored, the next 8 columns contain a character string, and the next 6 columns contain a floating point number that usually will have one digit after the decimal point. Formats of the form (*codes1*/(*codes2*)) mean that the first line uses the format indicated by *codes1*, while all the subsequent line uses the format indicated by *codes2*.

2.3.3. Concentration, Time, and Rate Parameter Units Used

Although model can be prepared and simulations can be conducted using any units for concentration, times, and rate constants (with certain limitations, discussed below), the software is designed for simulations using concentrations in ppmV, times in minutes, and rate constant units that correspond to these (referred to as ppm-minute units)[2]. Therefore, the documentation is based on the assumptions that these are the units that are being used in the simulations. These are common units used in simulations of ground-level air pollution and of environmental chamber experiments. The model preparation program has options to convert kinetic parameters from molecules, cm, sec units[3], but these options cannot be used if something other than ppm-minute units are used in the simulations.

The programs have been thoroughly tested using ppm-minute units, but the success of the simulations cannot be assured if the magnitudes of the concentrations or times are many orders of magnitude different than one. Test calculations have shown that the simulations will fail or the time data will not be properly output if the time units are orders of magnitude different than unity. Simple test calculations using concentrations, times, and rate constants in molecule, cm, sec units seem to work properly, though this has not been tested with complex models.

As discussed in the descriptions of the input files, temperature units must always be in degrees K, and kinetic parameters that refer to activation energies must be either kcal/mole or degrees K, depending on the input options used. This is built into the software. It may be possible to use other temperature units if appropriate units are used for activation energies, but this has not been tested.

2.4. Summary of Programs and Files

The purpose of this section is to give an overview of the programs and files needed for various types of kinetic modeling applications, and how they are related, so potential users can understand what programs and files are involved for those applications. The steps involved in running the programs for different purposes are summarized, but details on how to prepare inputs for specific programs for specific purposes are given separately in the "Program Details" section of this document. This is because not all of the details and features are needed in most cases, and the overview information in this section should be useful for directing users to the portions of the program details sections that are most relevant to the specific applications of interest.

The relationships between the various programs and files for various modeling applications other than those for incremental reactivity calculations are shown on Figure 1 and Figure 2, and Table 2 lists

---

[2] Rate constant units depend on the order of the reactions, i.e, the number of model species reactants. For ppm-minute units the units for 1st, 2nd, and 3rd order reactions are min^{-1}, ppm^{-1} min^{-1} and ppm^{-2} min^{-1}, respectively.

[3] Concentrations in $molecules/cm^3$, times in seconds, and rate constant units of sec^{-1}, cm^3 $molec^{-1}$ s^{-1}, and cm^6 $molec^{-2}$ s^{-1} for 1st, 2nd, and 3rd order reactions, respectively.

and briefly describe the files and programs involved. Table 1, above, shows the directory structure and locations of the files as distributed. An overview of use of this software and files for the various modeling applications is discussed below. The programs and files for incremental reactivity calculations are discussed separately in Section 3.6 because they are more complex and may not be of interest to most users.

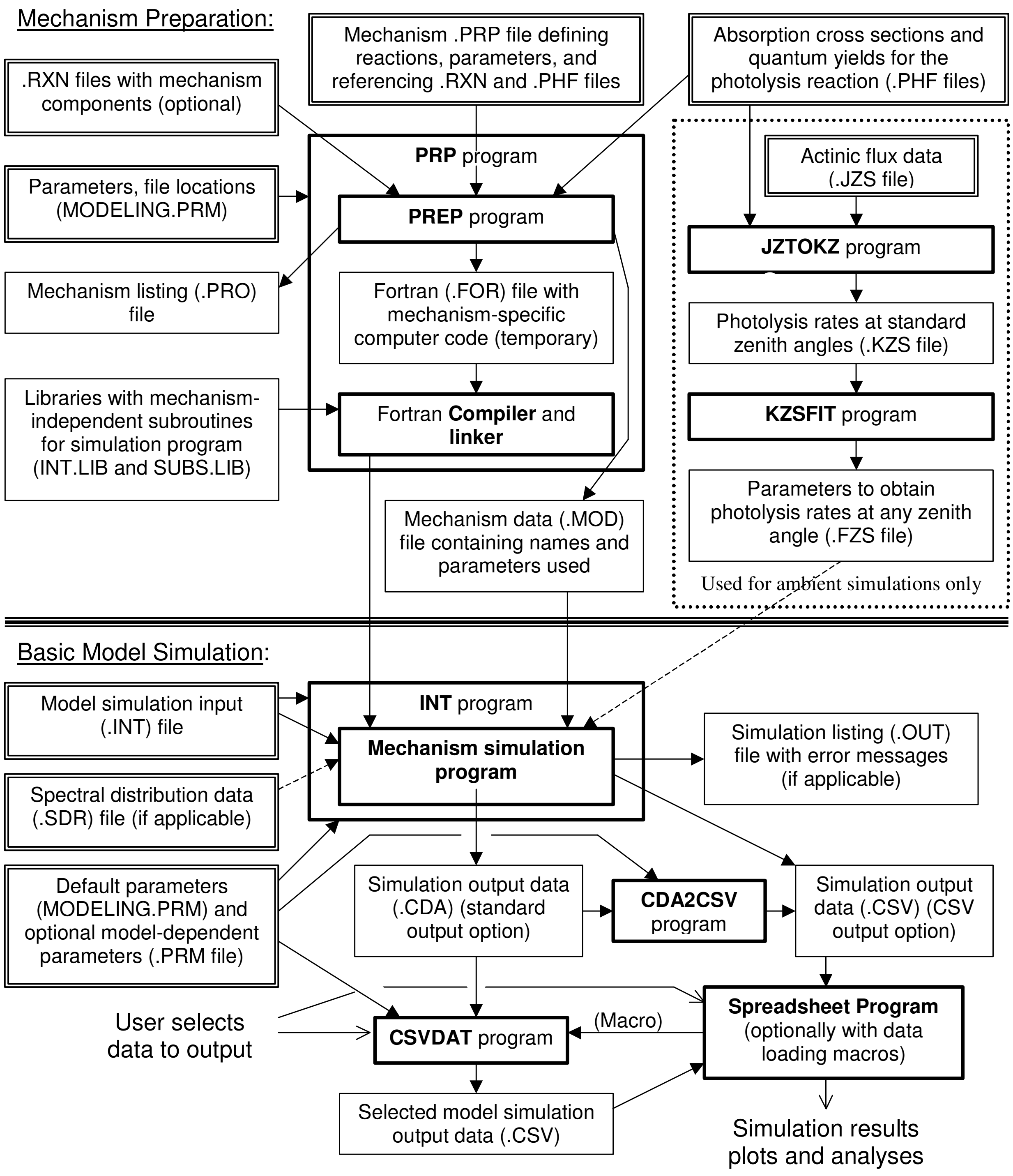
2.4.1. Model Preparation

The relationships between the various programs and files involved in basic model preparation are shown on the top part of Figure 1 and Table 2. The main model preparation input is in the model's PRP file, whose name also gives the name of the model-specific model simulation program (.EXE file) to be prepared and also the model data (.MOD) file, and whose contents give the reactions, and parameters and files that define the model. The model preparation input includes a list of all the reactions in the models, listing the model species reacting and being formed in each reaction, and the yields of the model species formed in the reactions. The reactions can represent elementary chemical processes or lumped overall process that can form many products with non-integer yields. The parameters used for defining the rate constants as a function of temperature, pressure, and optionally other parameters are also given, and a number of different types of temperature and pressure dependences are supported. If the model has photolysis reactions then the name of a .PHF file containing the absorption cross section or quantum yields must be given or otherwise the values of all the photolysis rates would have to be input in the model simulation step. Parameters for defining emissions rates and emissions or inputs of groups of compounds for reactivity simulations can also be specified if applicable -- this is used primarily for simulations of the more complex ambient scenarios used for reactivity calculations, The model input .PRP file can optionally refer to various .RXN files containing portions of the model, but this option is useful primarily when it is convenient to organize modules of large and complex models into separate files. (The use of .RXN for the file type is by convention; any file extension name can be used for this purpose.) For simpler models it is usually more convenient to have all model definition input other than absorption cross section and quantum yield data in a single .PRP file. The input formats are the same regardless of whether the data are in the main .PRP file in or a referenced .RXN file. The input data and formats are described in Sections 3.2.3 through 3.2.11.

The model preparation programs and also most of the other programs in the SAPRC modeling system also read a file called MODELING.PRM to obtain parameters that control the programs. In the case of basic model preparation, MODELING.PRM is only used to obtain names of folders or directories for input and output files. In most cases the only required input is the SOURCE parameter, which tells the PRP program where to find the INT.LIB and SUBS.LIB object libraries containing the model-independent portions of the model specific model simulations being prepared. In addition, MODELING.PRM can be used to give the locations of the .PHF files used for input, and the model.EXE and .MOD files to be output, if they are different than the default directory when the program is run, which must be the location of the .PRP file. This is not the case for the example model files in the distribution. The data and options specified in MODELING.PRM are described in Section 3.1.

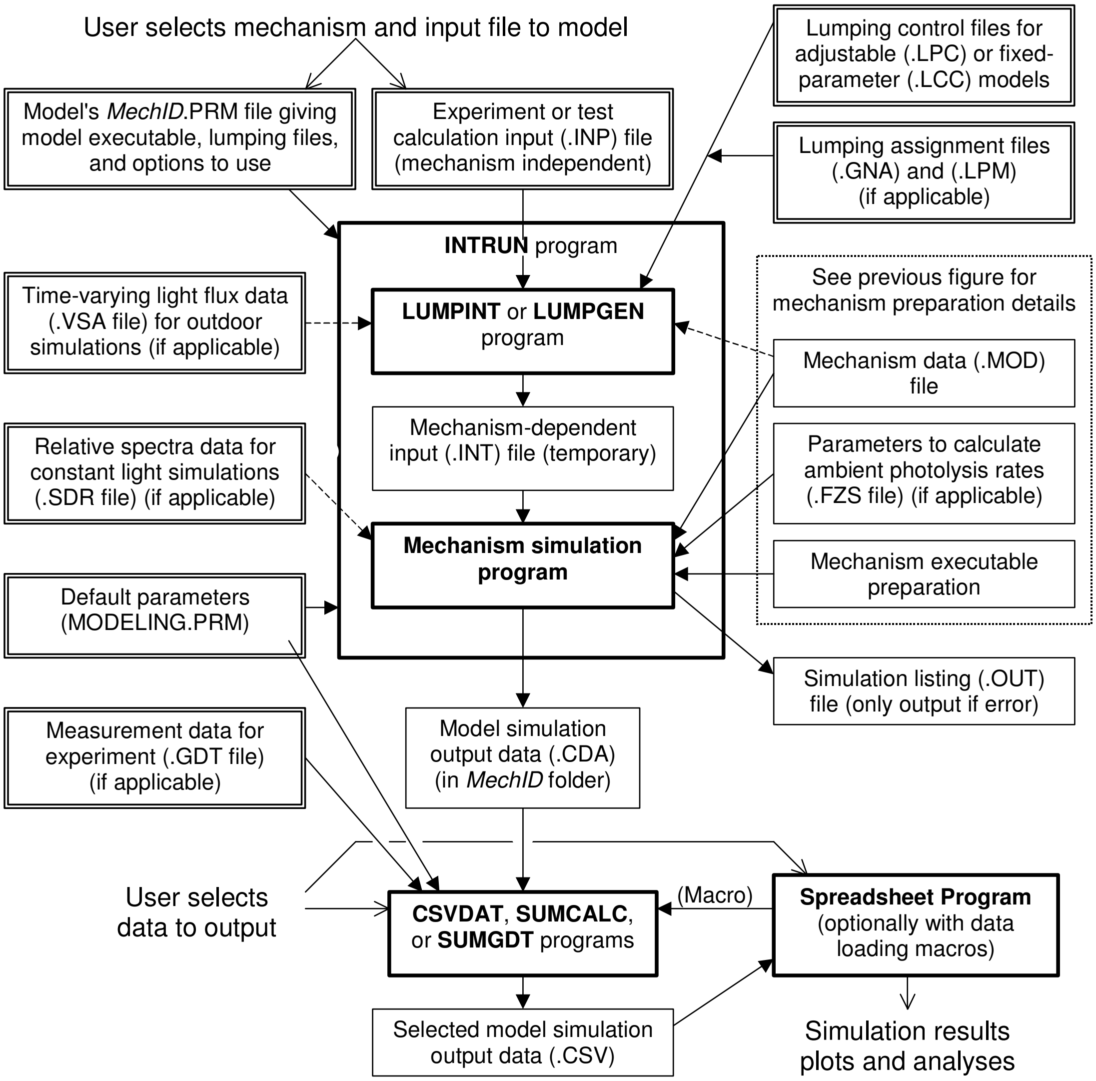
The PRP program is generally the one invoked by the user to prepare and compile the model. It is described in more detail in Section 3.2.1. It must be called by the following procedure.

1. Make sure that the .PRP file and any .RXN and .PHF files it references have the desired model input information and that MODELING.PRM refers to the correct location of the distributed program source libraries and the locations of the .PHF files if different from the location of the .PRP file.



Note: Bold boxes and font indicates programs. Programs called by other programs and temporary files are shown as boxes within boxes. Input files that the user can modify are shown as double-line boxes. Dashed arrows indicate inputs that may be needed depending on how the light source is represented.

Figure 1. Relationship between programs and files in the SAPRC model simulation software system: 1. Model preparation and basic model simulation applications.



Note: Bold boxes and font indicates programs. Programs called by other programs and temporary files are shown as boxes within boxes. Input files that the user can modify are shown as double-line boxes. Dashed arrows indicate inputs that may be needed depending on how the light source is represented.

Figure 2. Relationship between programs and files in the SAPRC model simulation software system: 2. Model simulations of chamber experiments or mechanism test calculations.

Table 2. Summary of programs, templates, and files in the SAPRC model simulation software system, excluding files and programs used only for incremental reactivity simulations.

Name [a]	Notes [b]	Description
		<u>Files and programs used for model preparation</u>
PRP.EXE	P, A, Sec.3.2.1	Prepares a model for model simulations. Runs PREP and then compiles the model simulation program
PREP.EXE	P, A, Sec.3.2.2	Reads model input file and outputs model-specific subroutines and data for the model simulation program. This is generally run using PRP rather than run directly by the user.
GFORTRAN.EXE LIB*.DLL [c]	P, A, Sec.2.2.2	This is the compiler and linker used with the system and is called by PRP to compile the model-dependent subroutine and link it with the rest of the modules to create the model-specific executable. Other compilers could be used but modifications to the PRP program may be required, and the INT.LIB and SUBS.LIB files (discussed below) would need to be rebuilt. The LIB*.DLL files are necessary for all the executable programs in the current distribution to run, and must be on the system path.
Mechname.PRP	I, A, U, Sec.3.2.3	Main input file used for model preparation. Gives the species, reactions, rate parameters and other model-specific parameters and data. Can optionally give files where portions of the model are input (generally .RXN files for SAPRC mechanisms).
Setname.RXN	I, s, Sec.3.2.3	Portions of a model preparation input that can optionally be included in a separate file and referenced in the .PRP file. Used to separate portions of the SAPRC mechanism for different types of simulations
Photname.PHF	I, A, Sec.3.3.4	Absorption cross section and quantum yield data for a photolysis reaction in the model. Name is referenced in the .PRP or .RXN file when photolysis reactions are specified.
Mechname.PRO	L, A	Model listing and results summary file output by PREP
Mechname.MOD	O, A	Model data file output by PREP. Read by the model simulation program to obtain lists of names and parameters in the model
Mechname.FOR	O, A	Model-specific FORTRAN subroutine output by PREP and compiled and linked into the model-specific model simulation program as part of the operations of the PRP program. Normally this is deleted by PRP if the model preparation is successful.
INT.LIB SUBS.LIB	A	Library of object files for model-independent FORTRAN subroutines that are linked into the model-specific model simulation program as part of the operations of the PRP program. Sources for these subroutines are included in the distribution. INT.LIB contains the subroutines only used for the simulation programs, while SUBS.LIB include subroutines that are also used for other programs in the distribution.
MODELING.PRM	I, A, Sec.3.1	Parameters used to control the operations of the programs. Used to indicate file locations and options. Generally the distributed file does not need to be changed unless the relative file locations are changed.

Table 2 (continued)

Name [a]	Notes [b]	Description
		<u>Used for model preparations for ambient simulations only</u>
KZSPREP.BAT JZTOKZ.EXE KZSFIT.EXE	P, r, Sec.3.3.5	KZSPREP.BAT is used to create input files used to calculate solar photolysis rates as a function of zenith angle for the purpose of ambient simulations and reactivity calculations. It invokes JZTOKZ to calculate the photolysis rates given the .JZS file with the actinic fluxes at a set of standard zenith angles and the list of the photolysis (.PHF) files in PHF.LIS, then KZSFIT to calculate fitting parameters used by the simulation program to get the photolysis rates at any zenith angle from those at the standard angles.
PHF.LIS	I, r, Sec.3.3.5	Contains the list of all the photolysis (.PHF) files in the model. This is used by JZTOKZ (called by KZSPREP) to determine which photolysis rates need to be calculated.
Lightmodel.JZS *Lightmodel*.KZS *Lightmodel*.FZS	I, r O, r O, r, Sec.3.3.5	*Lightmodel*.JZS contains the actinic fluxes at specified zenith angles calculated by a particular solar light model. This chemical mechanism-independent file is used as input to create solar photolysis rates as a function of zenith angle for the chemical mechanism for the purpose of ambient or reactivity simulations. The .KZS file contains the photolysis rates at the same zenith angles as those in the .JZS file and the .FZS file contains fitting parameters that can be used by the model simulation to get the photolysis rate at any zenith angle.
		<u>Used for lumping input processing where applicable.</u>
Lumpmethod.LCC	I, c, t, r. Sec.3.4.4	Lumping control file for fixed parameter models to give the model species used to represent various compounds. Used as input to LUMPGEN (see programs for model simulations) to create model-specific simulation input files from model independent inputs.
Lumpmethod.LPC	I, c, t, r, s X, Sec.3.4.5	Lumping control file for adjustable parameter detailed mechanisms such as detailed SAPRC, where the mechanistic parameters for some model species depend on the compounds being represented in the simulations. Used as input to LUMPINT (see programs for model simulations) to create model-specific simulation input files from model independent inputs. References various .GNA and LPM files for various types of lumping information and parameters.
Speciesgroup.GNA	I, c, t, r, s, Sec.3.4.5	Model parameter files for various groups of compounds giving the rate constant and product yield parameters to be used in adjustable mechanisms for the various compounds. Referenced in the .LPC file and used as input to LUMPINT.
LUMPMOLE.LPM	I, c, t, r, s. Sec.3.4	"Lumped molecule" assignments used for detailed, adjustable mechanisms such as detailed SAPRC. Gives names of compounds and the compounds with assigned mechanisms used to represent them. Also used to give compositions of simple isomeric mixtures.
OLDNAMES.LPM	I, c, Sec. 3.4	Referenced by .LPC files used for chamber simulations to translate discontinued species names used in input files for some older chamber runs into species names currently in use.

Table 2 (continued)

Name [a]	Notes [b]	Description
		<u>Files and programs used for model simulations</u>
Mechname.EXE	P, O, A, Sec.3.3	Model-specific model simulation program. Produced by PRP and invoked by the INT, INTRUN, REACTCAL, MROPT, or MROPTN programs depending on the modeling application. Generally not invoked directly by the user.
INT.EXE	P, g, Sec.3.3.1	Runs a model simulation with minimal options and no lumping or other pre-processing. The model used (the model executable) must be specified as part of the input file.
INTRUN.EXE	P, c, t, Sec.3.4.3	Runs a model simulation with pre-processing of the input for lumping and other options. Used for chamber simulations and test calculations. The parameter (.PRM) file used to determine the model and other options must be specified.
LUMPINT.EXE LUMPGEN.EXE	P, c, t, r, s, Sec.3.4	Used to pre-process model simulation files to derive lumped species and parameters corresponding to input species and for other pre-processing purposes. Called by INTRUN to produce input to model-specific simulation program, and called by the batch files that prepare models for reactivity simulations. Parameter file specifying the model and the lumping control files must be specified. LUMPINT is used for adjustable parameter mechanisms such as detailed SAPRC and LUMPGEN is used for fixed parameter mechanisms such as CSAPRC or CB05.
Calcname.INT	I, O, g, U, Sec.3.3.1	Input file to model-specific simulation program. Gives initial concentrations and simulation program control parameters. Can also be used to change some model-dependent parameters to non-default values. All species and parameter names must be defined for the model. This can be created by the user or is created from the *calcname*.INP file by LUMPINT or LUMPGEN.
Calcname.INP	I, c, t, U, Sec.3.4.3	Input file to the LUMPINT or LUMPGEN programs giving the initial concentrations other simulation program control parameters, including some that may be model independent. This input is used by LUMPINT or LUMPGEN to create the corresponding model-dependent .INT file that is used as input to the model-specific simulation program.
Modelname.PRM	I, c, t, Sec.3.1	Model-dependent parameters for model simulations. Specifies the model-specific model simulation to use and the model-dependent parameters such as lumping control files and options. Also specifies file locations for model-dependent parameters. Required input for INTRUN and some of the utility programs used to collect and summarize the model simulation results. The name of this file is generally also used to determine where the simulation output files for the model are located.

Table 2 (continued)

Name [a]	Notes [b]	Description
MODELING.PRM	I, A, Sec.3.1	Default parameters for the model simulations. Used to indicate file locations and options. For chamber and test calculation simulations it contains only model-independent parameters since the model-dependent parameters are read from the *modelname*.PRM file. For incremental reactivity simulations it also contains the model dependent parameters since otherwise .PRM files are not used by REACTCAL and the other reactivity programs.
Calcname.OUT	L, A, U, Sec.3.3.6	Listing output file created by the model simulation program, summarizing inputs, results, and errors. This output file can be suppressed using the "NOOUT" option in the input (.INT or .INP) file, but is still produced if an error occurs so the user get information on the source of the error. The distributed chamber simulation and test calculation input files have this NOOUT option so this file is not produced unless an error occurs.
Calcname.CSV	O, g, U, Sec.3.3.6	The calculated concentration-time data can optionally be output as CSV files that can be directly imported into spreadsheet programs. This option may be the most convenient option for the general user with relatively small numbers of simulations. This option is not used in the distributed input files for chamber and mechanism comparison simulations; those input files cause the calculation results to be output in .CDA files instead. This CSV option should not be used when more than 255 species are output.
Calcname.CDA	O, c, t, r, Sec.3.3.6	The calculated concentration-time data can optionally be output in a fixed-width ASCII format that can be read by the distributed utility programs that collect the data for plotting or analysis. If INTRUN is used, these files are located in subfolders named by the name of the *modelname*.PRM file that specifies the model used when the simulation is run. If INT is used and no parameter file is specified, they are located in a subfolder called CALCS.
		<u>Input files for modeling chamber experiments only</u>
RunID.INP	I, c	Model simulation input data for a chamber experiment, giving conditions and initial concentrations. Reactants are given in terms of actual compounds and LUMPINT or LUMPGEN is required to convert this in a model-specific format for model simulations. One such file is included for each chamber experiment in the distribution, but others can be created for new experiments.
RunSet.CHR	I, c	Characterization input for sets of chamber experiments assumed to have similar chamber-dependent parameters. There is at least one such file for each chamber for which experiments are modeled, and for some chambers there are several such files if conditions are believed to change. LUMPINT or LUMPGEN takes this input as part of processing the *RunID*.INP files for model simulations.

Table 2 (continued)

Name [a]	Notes [b]	Description
LightSource.SDR	I, c	Relative spectral distributions for each light source used for indoor chamber experiments that are modeled. The distribution includes at least one such file for each type of light source employed. This is required input to the simulation program for all indoor chamber experiments.
RunID.VSA	I, c	Absolute actinic fluxes at specified times for the outdoor chamber runs (that use natural sunlight) in the distributed database of chamber experiment. One such file for each experiment, though for dual chamber experiments one file is used for modeling both reactors. These are used as input for LUMPINT or LUMPGEN when it is pre-processing outdoor chamber experiments for modeling. LUMPINT uses these data (together with the action spectra data for the photolysis reactions in the model taken from the .MOD file) to calculate photolysis rates as a function of time for inclusion in the input to the model simulation program.
		<u>Programs and files for processing mixtures</u>
RENORCMP.EXE	P, r, c, Sec.3.4.1	Utility program that normalizes the compositions of complex mixtures in mixture .CMP files to 1 ppmC total. This normalization is necessary if the same mixture is used in more than one experiment, with the amount added in ppmC being given in a #FAC record in the input file. This is not necessary for the mixture .CMP files in the distribution, but should be used for new mixtures. The molecular weight and carbon number information is obtained from DMS.PRM.
DMS.PRM	I, r, s, Sec.3.4.1	Contains list of names used for organic compounds for which mechanisms have been developed and simple isomeric mixtures for which compositions have been assigned, along with their carbon numbers, molecular weights and other information. Used by RENORCMP and DMSRCT.
Mixture.CMP	I, c, r, Sec.3.4.1	Compositions of complex mixtures used in some experiments. The compositions should all be normalized to 1 ppm Carbon. The experiment's input (.INP) file references this mixture, and has a #FAC record indicating the carbon number, which is multiplied by all the concentrations in the mixture to give the initial concentrations. This is then converted into moles of model species when the input file is pre-processed by LUMPINT or LUMPGEN.
		<u>Files and programs for presentation and analysis of results</u>
CDA2CSV.EXE CDA2CSV2.EXE	P, A, Sec.3.5	Produces a CSV file containing all the concentration-time data for a calculation using a CDA file, referenced by giving the calculation and model name, as input. CDA2CSV produces a file with the channel names in rows and times in columns, and does not include all the data if there are more than 255 species calculated. CDA2CSV2 produces a file with the channel names in rows and the times in columns and does not include all the data if there are more than 255 times output.

Table 2 (continued)

Name [a]	Notes [b]	Description
CSVDAT.EXE	P, A, X	Produces CSV files with selected data for import into spreadsheets. Calculation results are taken from the CDA files giving the calculation input file name and model parameter name. This can also be used obtain selected experimental results data from experimental .GDT files (see below) for comparison of experimental with calculation results. Used by the distributed Excel templates for presentation or results to obtain the data for the spreadsheets.
SUMCALC.EXE SUMCALC2.EXE	P, c, t, X	Produces a summary of calculation results given a list of calculations (or experiments modeled) and the model (parameter file name) used. The results summarized can be hourly (for SUMCALC) or half-hourly (for SUMCALC2) data, interpolated if needed, or initial, final, or integrated values for selected species. Outputs a .CSV file that can be imported into spreadsheets
INTRATE.EXE INTRAT2.EXE	P, A, X	Produces a CSV file that can be imported into spreadsheets for analysis of reaction rate or integrated reaction data if the option to output these data is selected when the model simulation is run. Used by spreadsheets created using the INTRATE or INTRAT2 Excel templates to import these data. The calculation ID, model parameter file name, and optionally the time for the rate or integrated rate data must be specified. If the time is not given the final rate or integrated rate data are used.
RUNCALC.XLT	T, U, X	Excel template for creating spreadsheets to aid in running model simulations of chamber experiments and producing plots of concentration-time data for selected species. Template includes plots of up to 6 species for comparing results of up to 4 model calculations with each other and optionally with results of up to 2 chamber experiments on the same plot. All the experimental and calculated results are loaded in using the CSVDAT program, and once the data are loaded the names of the species plotted can be changed without re-loading the data. The macros in the template can be used to run the simulations for specified models as well as retrieve data from previously run simulations.

Table 2 (continued)

Name [a]	Notes [b]	Description
CALCWORK.XLT	T, U, X	Excel template for creating spreadsheets to aid in running model simulations for comparing and analyzing simulations or mechanisms. Three model simulations, with different inputs, models, or parameters, can be compared, plotted, and analyzed. Similar to RUNCALC except it does not have provision for experimental data but allows for more options for analysis of results. This includes calculations and analyzes of instantaneous or integrated rates of all reactions, determining the relative importances of reactions in contributing to consumptions or formations of species or groups of species, and comparing calculations. All the calculated results are loaded in using the CSVDAT program, and once the data are loaded the names of the species plotted can be changed without re-loading the data. The macros in the template can be used to run the simulations for specified models. The inputs can be specified in the file or taken from the input file for an experiment.
		<u>Programs and files for analysis of chamber simulations</u>
RUNSFIT.XLT RUNFITPLT.XLT	T, c, X	RUNSFIT.XLT is an Excel template for creating spreadsheets to aid in running model simulations of multiple chamber experiments and producing plots of concentration-time data for selected species. Data from up to 127 experiments and multiple channels can be plotted, which each plot showing data for up to three calculations and one experiment. The template as distributed contains no plots but the plots for the desired number of experiments and channels are created by a macro, using RUNSFITPLT.XLT to make the plots (this template is not accessed by the user directly). The macros in the template can be used to run the simulations for specified models as well as retrieve data from previously run simulations.
GDT2CSV.EXE	P, c, X	Converts experimental data read from an experiment's .GDT file to CSV format for import into spreadsheets.
SUMGDT.EXE SUMGDT2.EXE	P, c, X	Outputs hourly (for SUMGDT) or half-hourly (for SUMGDT2) experimental data for selected measurement channels for comparison with model calculation results. Interpolated data are used if there are no measurements on the hourly or half-hourly interval. Gets the experimental data from the .GDT files (see below) for the experiments.
SUMINP.EXE	P, c, X	Summarizes the model simulation input data for a group of selected experiments. A summary of the initial concentrations are output in a .CSV file that can be loaded into a spreadsheet. Not needed for modeling but useful for obtaining summaries of inputs for runs that are modeled.
RunID.GDT	I, c, X	Experimental measurement data from an experiment for comparison with model calculation results. There is one such file for each experiment for which input data are included for modeling. These files are read by the CSVDAT, GDT2CSV and the SUMGDT programs for importing the data into spreadsheets.

Table 2 (continued)

Name [a]	Notes [b]	Description
CHAMEXPT.XLS	c, X	List of all chamber experiments that are included in the distribution and summary of information about the experiments. Indicates the types of experiments and has a summary of the input data.
RUNCALCEX.XLS RUNSFITEX.XLS	c, X	Example spreadsheets of calculation results based on RUNCALC.XLT or RUNSFIT.XLT for representative experiments.
		Input and output files for calculation of incremental reactivity scales are given in Table 14.

[a] Names in *italics* are variable depending on the model or the simulation. Standard file types included in the distribution are as follows: EXE = executable program; LIB = library of compiled subroutines; BAT = DOS command-line batch file; FOR = Fortran source file; XLS = Excel file; XLT = Excel template file; CSV = comma-separated data file; TXT = text file used for information purposes; PDF = Adobe PDF file. All other file types are specific to the SAPRC modeling system and are described in the body of the table.

[b] Codes indicating use and type of file are as follows:

P	Executable Program. Included in the distribution unless indicated otherwise.
I	Input file prepared by user and/or included in the distribution.
O	Output file produced by a program that can also be used as an input file for another.
L	Listing file for output for information purposes. Not used as an input file for any program
S	Program source or compiler output file included with the distribution
U	This type of input file is the most likely to be created, modified or examined by the user
T	Excel template.
X	A complete discussion of this file or program is beyond the scope of this document.

Codes indicating types of model simulations where the program or file may be used

A	Used for all or most types of simulations
g	Used for general or simple model simulations
c	Used for simulations of chamber experiments
t	Used for test calculation simulations
r	Used for incremental reactivity simulations (Most of the reactivity programs and files are listed in Table 14.)
s	Used for SAPRC mechanisms primarily, though could be used for other mechanisms.

[c] This public domain software is not included with the distributed SAPRC files but is available at the GNU web site at http://gcc.gnu.org/wiki/GFortranBinaries#Windows

2. Open a DOS window and make sure that the environment is such that preparation programs and other executables in the distribution, and also the Fortran compiler and linking programs are on the path. For the distributed files running the NEWENV.BAT file and installing MINGW with gfortran should be sufficient for this purpose (see installation instructions, above for details).
3. Navigate to the folder or directory containing the .PRP file for the model being processed. The distribution includes distributed files for several mechanisms, though more complex mechanisms (such as detailed SAPRC) may contain more than one .PRP file for different versions of the

mechanism for use in different modeling applications, such as ambient or environmental chamber simulations.

4. Invoke the PRP program by giving the command "PRP *model*", where *model* is the name of the version of the mechanism being prepared. The input is taken from *model*.PRP and the model listing, executable, and data files output are *model*.PRO, *model*.EXE and *model*.MOD, respectively. Normally these are in the same folder as the .PRP file.

The PRP program reads MODELING.PRM to get file locations, then calls the PREP program to read the .PRP, .RXN, and .PHF input files and output the model data (.MOD) file and a temporary Fortran source (.FOR) file containing the model-specific subroutine(s) needed for the model simulation program. If this is successful, PRP then runs the gfortran compiler to compile the model-specific subroutines(s) and link it with the model-independent modules (INT.LIB and SUBS.LIB) to produce the model-specific executable. Optionally this can be done as two steps, first by giving the command "PREP *model*" and then by giving the compile and linking commands, which depend on the locations of the libraries and modules used. The intermediate .FOR file is deleted if PRP is used, but is saved if PREP is called explicitly. The PREP program (whether called explicitly or by PRP) also outputs the *model*.PRO listing file containing information about the model and any error messages. PRP does not prepare a model executable if PREP encountered sufficiently severe errors.

Figure 1 shows that an additional model preparation step is needed if ambient simulations are to be carried out using appropriate photolysis rates for the time-varying solar zenith angles. However, this step is only needed once for each set of actinic flux values as a function of zenith angle and each set of photolysis (.PHF) files with the wavelength-dependent quantum yields for the photolysis reactions, even if multiple versions of the mechanisms with different .PRP files are used. The JZTOKZ program takes as input a .JZS file containing actinic flux values (in terms of quanta per unit wavelength at specified wavelengths) at a standard set of zenith angles (0, 10, … 70, 78, and 86^o) and outputs a .KZS file in the local folder containing the photolysis rates at those zenith angles calculated from each of the .PHF files in the local folder. Since .JZS files are model-independent they are generally in different folders than the model-specific files containing the .PHF input and the .KZS output files, so the locations of these files are specified in MODELING.PRM. The KZSFIT program is then run to find parameters that fit these photolysis rates as a function of zenith angle so that smoothly-varying photolysis rates can be calculated for any zenith angle. These parameters are output in a .FZS file in the same folder, and the data in this .FZS file can be read by the model simulation program when conducting ambient simulations using the actinic fluxes in the corresponding .JZS file as part of the inputs of the scenario being modeled.

The JZTOKZ and KZSFIT programs are invoked by steps 1-3 given above for running the PRP program, followed by the commands "JZTOKZ *lightmodel*" and then "KZSFIT *lightmodel*", where *lightmodel* is the name of the model-independent *lightmodel*.JZS file used as the input and the model-dependent *lightmodel*.KZS and .FZS files that are output. Because the actinic fluxes in the .JZS files refer to scenario conditions and not the chemical mechanism, the distributed .JZS files are in the same folder as the scenario input files, while the .KZS and .FZS for the mechanism are in the folder with the model files (see Table 1). But it is important that the users understand that the ambient photolysis rates reflected in the .KZS and .FZS files are strictly speaking *not* part of the mechanism or *not* part of the scenario conditions, since they are dependent on the combination of both.

2.4.2. Basic Model Simulations

The bottom part of Figure 1 shows the relationships between the programs and files involved with basic model simulations using the minimal sets of programs and options for that purpose. Summaries of the programs and files involved are also included on Table 2. This is not the only way the model simulations can be run, but may be the most straightforward for basic applications.

The main input to the simulation program is the .INT file, which for basic applications indicates which model simulation program to use and gives the initial reactant concentrations and other simulation conditions, the time period for the simulation, and how often concentration-time data are output. For basic applications these files are prepared manually by the user, and the required and optional inputs and formats are discussed in Section 3.3.1.

If the model has photolysis reactions then spectral distribution data for the light source and a measure of the overall light intensity must be input so the program can calculate the photolysis rates, or the relevant photolysis rates need to be included explicitly in the .INT file. For basic applications with a constant light source, the recommended procedure is for the .INT file to reference a .SDR file that contains the relative spectral input, and the rate coefficient for a selected photolysis reaction (usually the NO_2 photolysis rate for air pollution simulations) must also be included in the .INT file. The relative spectral data in the .SDR file is then used to calculate the ratios of rate coefficients for the other photolysis reactions relative to the input rate constant, from which the absolute photolysis rates can be calculated. There are other ways to input photolysis rates for ambient simulations or simulations of outdoor chamber experiments, and these are discussed in the following sections.

The simulation program also takes input for default program control parameters from the file MODELING.PRM. For basic model simulations this would generally contain the location of the .SDR files used as input and the base location for the output files, as discussed below. Additional parameters, generally those that depend on the model used, can also be given in MODELING.PRM if those are to be used as defaults. However, a separate model parameter (.PRM) that can optionally be specified when the model simulation program is run to contain parameters that depend on the model used. Parameters in MODELING.PRM are still read if a model's .PRM file is used, but parameter values in MODELING.PRM will be over-written if they conflict with those given in the model's .PRM file. The model's PRM file is only needed if more than one model is used, and then only if the parameters and the file locations are different for the different models. These .PRM files generally do not need to be modified once they are set up appropriately for the application and file locations.

By default the program will output only an .OUT file that contains a listing output that contains the a listing of the model and the concentration-time data in a format designed for printing but not for serving as inputs into spreadsheets or other programs. Generally the .INT file should contain a "SAVE" or "SAVE.CSV" command as part of its inputs, and these will cause the program to output the results in formats that are more useful for subsequent data analysis. The contents of the .OUT files are generally not of interest for subsequent analysis if a SAVE command is used, and a "NOOUT" option can be used to suppress producing these files unless an error occurs. The "SAVE CSV" command results in the concentration-time data to be output in a .CSV file in the local folder, with the same name as the .INT file. The "SAVE" command by itself results in the concentration data being output in a .CDA file that can be converted to a .CSV file containing all or selected subsets of the programs by other programs, as discussed below. This option is generally useful when calculations are to be conducted and compared using different models, or data from multiple calculations are to be used in subsequent analyses, and is required if more than 255 concentrations are to be output at each time interval. These CDA files are saved in a folder named by the name of the model parameter file if one is used in these calculations, or in a folder called "CALCS" if not. These folders for CDA files are either below the root folder where the program was run or in a different location specified in MODELING.PRM.

If the program outputs a .CSV file directly then it can be loaded into a spreadsheet program for subsequent analysis or plotting. CDA files cannot be read directly by spreadsheet programs, but they are fixed-width ASCII files that can be easily read by Fortran programs, including distributed programs that can convert them to .CSV files with all or selected data. Figure 1 shows that the CDATOCSV program can be used to produce .CSV files with all the data in a .CDA file (if it has 255 or fewer output concentrations per time step), and the CSVDAT can be used to produce .CSV files for selected data. Both of these are used by macros in excel files included in the distribution to aid in the use of this software for modeling applications.

Although the model's simulation program prepared by PRP as discussed above can be invoked directly, the most convenient way to invoke it for basic applications is to use the INT program as shown on Figure 1. This program takes the input from the main simulation input (.INT) and/or the parameter (.PRM) file(s) to determine which model simulation program to run and where it is located. The steps to invoke it are similar to those discussed above for PRP, and are as follows:

1. Make sure that the .INT file to be used has the desired input for the simulation and make sure that the MODELING.PRM and the model's .PRM file (if used) contain the appropriate file locations.
2. Open a DOS window and make sure that the environment is appropriate as discussed above in conjunction with preparing models.
3. Navigate to the folder or directory you will use for carrying out the simulations, which will contain the .INT and .PRM files used as inputs.
4. Invoke the INT program by giving the command "INT *calcname* [*modelname*]", where *calcname* is the name of the INT input file and *modname* is the name of the model's .PRM file, if used (i.e., input from *calcname*.INT and *modelname*.PRM). If *calcname*.INT contains the "SAVE CSV" command the results will be in the file *calcname*.CSV in the current folder. If it contains the "SAVE" command by itself the output will be in the file *calcroot*\CALCS *calcname*.CDA or *calcroot**modname**calcname*.CDA, depending on whether a model parameter file is specified, where *calcroot* is the location of the calculation outputs as specified using the CAL= option in MODELING.PRM.

If the data are output as CDA files the user can create .CSV files with some or all the results using the CDA2CSV or CSVDAT programs. If there are not more than 255 species to be output a .CSV file with all the results can be produced by giving the command "CDA2CSV *calcname*" if a model parameter file is not specified when INT was run, or "CDA2CSV *calcname modname*" otherwise. Selected data can be obtained using the CSVDAT program for output into CSV files, where input to the program includes the calculation name (*calcname*), the model name (*modname*) or "CALCS" if a model name was not specified, and the names of the model species to output. This program is discussed in Section 3.5.

2.4.3. Ambient Sunlight Simulations

Additional processing is required to carry out simulations of scenarios using ambient sunlight or outdoor chamber or other experiments where the spectrum of the light varies with time. For ambient simulations the recommended procedure is to start with a .JZS file containing solar actinic fluxes at a standard set of solar zenith angles (0, 10, …, 70, 78, 86) and use the distributed JZTOKZ and KZSFIT programs to produce photolysis rate input for the simulations, as shown on Figure 1. The actinic flux data that go into .JZS files are independent of the chemical mechanism and can be created using light models such as the TUV model (TUV, 2019), the unpublished light model developed by Jeffries (personal communication, 1989-1990) that was used to prepare the actinic flux inputs for the reactivity scenarios developed by Carter (1994a), or those given by Peterson (1976). The calculation of such fluxes is beyond the scope of this document, but examples of such files are included in the distribution.

As shown on Figure 1, the JZTOKZ program calculates the photolysis rates given the actinic fluxes that are in the .JZS file, using the absorption cross section and quantum yields in the .PHF files used by the chemical mechanism, and outputs them in a .KZS file. The KZSFIT program then processes these data to calculate fitting parameters that are stored in a .FZS file for input into the model simulation program. Since the actinic flux data are part of the specification of the conditions of the ambient scenario being modeled and the absorption cross sections and quantum yields are part of the chemical mechanism, the rates should not be considered to be part of any chemical mechanism or ambient scenario, but are hybrids of both. However, for the purpose of file organization using this software they are kept in the same folders as used for the chemical mechanism. This is because the names used for KZS and FZS are based on that for the scenario-dependent .JZS file, so it is the file location that is used to determine which model was used to calculate them.

If .JZS files are used, the photolysis rates need to be calculated for each model that will be used to simulate scenarios with these actinic fluxes. The procedure to do this is as follows:

1. Open a DOS window and make sure that the environment as needed as discussed previously.
2. Navigate to the folder or containing the .PRP files for the model(s) to be used. It may not be the same as that for the model's .PHF files if the file MODELING.PRM file in the folder with the .PRP files indicates that the PHF files are elsewhere, but the PHF and PRP files are in the same folder for the example models in the distributed files.
3. Give the command "JZTOKZ *fluxfile* [*file*]", where *fluxfile* is the name of the .JZS file (i.e., input from *fluxfile*.JZS) and *file* is optional and if given is the full name of a file containing a list of .PHF files to be processed. If *file* is not given the program gets a directory listing to find all the .PHF files in the local folder, and processes them all. This program then outputs *fluxfile*.KZS in the local folder, giving the photolysis rates for the zenith angles specified in the .JZS file. The command "KZSFIT *fluxfile*" is then used to produce the *fluxfile*.FZS file in the same folder that is used as input to the ambient model simulations.

The procedure for carrying out model simulations is similar to that discussed above for basic model simulations, except that additional input is needed in the simulations .INT file. Instead of specifying a .SDR file to use, the parameter section of the input file must contain the line "P(Z)=*fluxfile* to indicate which .FZS file to use. The simulation programs look in the same place for these .FZS files as it does for the model's .MOD and .EXE files, so no additional input is needed in the parameter input (.PRM) files if the .FZS files in the model preparation folders as recommended. The latitude, solar declination for the time of year, and the difference between solar and simulation time also need to be included in the input .INT file because solar time is required to calculate zenith angles and therefore actinic fluxes as a function of time.

An alternative approach can be used to calculate time-varying photolysis rates for modeling outdoor chamber experiments, which, because of wall or local environment effects, may have different actinic fluxes as a function of time than open environments. This was used by Carter (1990) and Carter and Lurmann (1990, 1991) when modeling experiments for University of North Carolina (UNC) outdoor chamber experiments because the UNC investigators did not recommend modeling these experiments using standard ambient solar fluxes, but instead provided us with recommended actinic flux data as a function of time (Jeffries and Sexton, personal communication). This approach was extended to our modeling of some of our outdoor chamber experiments (Carter, 2000), and is used when modeling some of the outdoor chamber experiments in the current distribution set. In this approach the input data for each experiment include a .VSA file that contains the actinic fluxes at given times throughout the experiment, and the and the photolysis rates at these times are calculated by the programs that pre-process the chamber data input for modeling as discussed in the following section. They are essentially the same as .JZS files except they the fluxes at given times rather than zenith angles, and they are processed by different programs. The photolysis rates at intermediate times are calculated by linear interpolation.

VSA files could also be used to calculate photolysis rates for ambient simulations, but it requires different VSA files for each location and time of year because the dependence of zenith angles and therefore actinic fluxes on time depends these factors. Also this is less efficient computationally because the actinic fluxes and photolysis rates do not change smoothly with time if VSA input is used, and this results in more time steps required for the simulations to complete with the desired accuracy.

The model simulation programs also provide for simulations of complex ambient scenarios with time-varying emissions, dilution, and reaction conditions such as temperature, as well as varying light intensity. This generally requires additional input at mechanism preparation time as described in Section 3.2.9, and additional input at model simulation time as described in Section **Error! Reference source not found.**. Such scenarios have been used primarily for conducting incremental reactivity calculations, and are discussed in Section 3.6.

2.4.4. Mechanism Evaluation or Model Comparison Simulations

Use of the INT program such as discussed above and shown on Figure 1 may not be the optimum approach when the objective is to evaluate or compare models, when input data need to be pre-processed to derive initial concentrations of model species from concentrations of measured species. Input data also need to be pre-processed if model uses model species whose parameters depend on the compounds being represented, such as the adjustable parameter versions of detailed SAPRC. A better alternative for these applications is to use the INTRUN program, as shown on Figure 2. This differs from INT in that main simulation data are in a model-independent input (.INP) file, and that a "lumping" program is called to pre-process the data to produce the model-dependent .INT file that the model simulation program needs. Use of the INTRUN is also required if the light source spectral flux data are to be input from .VSA files, as discussed in the previously. Note that INTRUN requires that a model parameter (.PRM) file be specified, unlike INT where its use is optional.

The format of the .INP files that are the primary inputs for model simulations where INTRUN is used is the same as that for the .INT files discussed above, which is discussed in detail in Section 3.3.1. The main differences are that it can include ".LUMP" input that means that the subsequent inputs include species that may need to be converted to model species depending on the model used, the specification of the model to be used is ignored (because it is given in the model .PRM file). It can also include "VARYSD" input that refers to a .VSA file containing time-dependent spectral data, where applicable. Also, the .INP files can optionally be located in different folders than where INTRUN is run and the .PRM files are located, as specified in MODELING.PRM. The distributed data sets include .INP files for each of the chamber experiments model test calculations that can be simulated, and the user can create additional such files for such simulations.

The other inputs are similar to those discussed above for INT, except that the use of model parameter (model .PRM) files is not optional, and lumping control input, discussed below, is also required. The model's .PRM file must be specified because the name of this file controls where the time-concentration data are stored, so they can be kept separate from those calculated using other models. The model's .PRM file must contain the model-dependent location such as the locations of the model preparation .MOD and .EXE files, as well as the lumping control parameters and files. The MODELING.PRM file should have model-independent information such as the locations of the .INP, .SDR, or calculation output files, but that information can be over-written by data in the model's .PRM file. However, if only one model is to be used then the model-dependent information can be put in MODELING.PRM and INTRUN can be run using "MODELING" as the model's .PRM file.

As shown on Figure 2, the INTRUN program operates by first running a lumping program, either LUMPINT or LUMPGEN depending on the type of model, to pre-process the .INP file and create the

.INT file that is used as input to the model-dependent simulation program. This .INT file is temporary and is deleted after the simulation, unless an error occurs or the "DEBUG" option is used when running the program. The LUMPINT program is used if the model has adjustable parameters that depend on the compounds being represented (e.g., detailed SAPRC mechanisms) and the LUMPGEN is used for fixed parameter mechanisms such as CB05, CSAPRC07 and SAPRC18.

The model's .PRM file must include input that indicates which of these programs is used (see Section 3.1), and for use with fixed parameter models it should include the line

LMPPGM=LUMPGEN or LUMPINT

It also must specify the lumping control files that are used, which for LUMPGEN is discussed in Section 3.4.4. The lumping control file for LUMPGEN is a .LCC file that indicates which model species are used for each compound that may be specified in the .INP files, which may be more than one model species for mechanisms such as CB05. The lumping control file for LUMPINT is a .LPC file that performs functions similar to .LCC files, but for lumped parameter mechanisms. In addition to giving model species used to represent specified compounds using the "lumped molecule" approach, it also gives adjustable parameter species concentrations and parameters based on the compound(s) they represent. Normally these lumping control files do not need to be modified once a mechanism is developed, unless new species are added whose model species assignments or parameters need to be specified.

The steps involved for running the INTRUN program are the same as those discussed above for .INT, except that it is invoked by using the command "INTRUN *calcname modname*" where *calcname* is the name of the INT input file and *modname* is the name of the model's .PRM file. As with INT, the simulation will by default output a simulation listing output (.OUT) file, and will also output concentration-time data as .CSV or .CDA files if "SAVE" or "SAVE CSV" is included in the input (.INP) file. The distributed .INP files for chamber simulations include the "NOOUT" option to suppress creation of .OUT files unless errors occur, and all contain the "SAVE" (not "SAVE CSV") command. The primary output for successful simulations then go to .CDA files in subfolders determined by the name of the model's .PRM file, i.e., *calcroot\modname\calcname*.CDA, where *calcroot* is the location of the calculation outputs as specified using the CAL= option in MODELING.PRM.

As shown on the bottom part of Figure 2, the results of the calculations can be loaded into spreadsheets after running the CDA2CSV or the CSVDAT program to use the data in the .CDA files to create .CSV files with all or selected calculation data. The distribution includes spreadsheet templates with macros to use these programs to load the data for selected calculations and models. CSVDAT and the spreadsheet macros can also be used to load experimental data for comparison with the model calculation if the experimental data are in .GDT files such as those given for each chamber experiment in the distribution that can be modeled. The distributed spreadsheet templates also include macros to run INTRUN with selected calculations prior to loading the data.

2.5. Simple Model Preparation and Simulation Example

2.5.1. Model Preparation

Example 2 shows an example of simple model preparation input and output. It illustrates use of each of the various types of model species in a very simple two-reaction model system, one of which is a photolysis reaction, the other of which has a constant rate constant. Example 2b shows the photolysis file used, which is a simplified approximation of the absorptions and cross sections for the photolysis of NO_2 to form NO and O^3P. The model species "A", "B", and "P1" are declared as active, though this is not necessary for "A" and "B" because they appear as reactants. The model species "S" is declared as steady-state and "HV" is constant with a default value of 1. The model species "P2" is formed in a reaction but is

Example 2. Example of simple model preparation input and output

Example 2a: File = EX1.PRP
Model preparation input

```
Simple example 1. A + HV -> S; S + B -> P1 + P2
! Declare species
.ACT
= A + B + P1
.STS
S
.CON
HV  1.0
.DUM
= Test-ACT + Test-BO
! Compute test sums to compare active and Buildup-only results
.INS SAVE
[Test-ACT]=[A]+[P1]
[Test-BO]=[A]+[P2]
! Enter reactions
.RXN
R1) PF=SIMPNO2  ;A + HV = S
R2)   1.0E+4    ;S + B = P1 + P2
.END
```

Example 2b: File = SIMPNO2.PHF
Simplified approximation of NO2 photolysis to NO + O3P

```
Simple approximation of NO2 + HV = NO + O3P (wl, abs, qy)
0.250 0.00e+00 1.0
0.300 1.30e-19 1.0
0.350 4.50e-19 1.0
0.400 6.00e-19 1.0
0.415 6.00e-19 0.0
```

Example 2c: Portions of EX1.PRO
Model preparation listing file

```
REACTIONS INPUT:                                 A        EA      B
                                                 (IN PPM-MIN UNITS)
 R1  ) A + HV = S                                  PHOT. = SIMPNO2          1   5   0 =    4
 R2  ) S + B = P1 + P2                        1.000E+04   0.00   0.000     4   2   0 =    3    8

LIST OF REACTANTS

    3 ACTIVE
       5) A                     6) B                     7) P1

    1 CONSTANT
       1) HV                 1.00

    1 BUILDUP-ONLY
       2) P2

    2 DUMMY (SAVED)
       3) Test-ACT              4) Test-BO

    1 STEADY STATE
       8) S

    4 VARIABLE COEFFICIENTS:
       1) DILUTION  0.00        2) LITTLE   1.000E-30    3) TEMPR     298.
       4) SIMPNO2   0.00
```

not declared as active, so it is buildup-only by default. The dummy species "Test-ACT" and "Test-BO" are added for the purpose of testing the effects of using the more approximate method used for calculating concentrations of buildup-only species. Because of the nature of the model, if dilution is zero then the sum of the concentrations of "A" and "P1" or "A" and "P2" should be constant at the initial concentration of "A", because "S" is held in steady state. The inserted instructions input calculates "Test-ACT" as the sum of "A" and "P1", both of which are active, while it calculates "Test-BO" as the sum of "A" and "P2", where "P2" is buildup-only. Selected portions of the listing output file (with some lines edited so they fit on the page without wrapping) are shown in Example 2c. Examples of calculations using this madel are discussed in the following section.

Note that Example 2 does not show the simplest possible example of a model input file, since use of constant, steady-state, buildup-only and dummy species is not always necessary, nor are photolysis reactions or inserted instructions. However, this example is useful to show useful features of the software with minimum complexity.

2.5.2. Model Simulation

Example 3 shows an example of a simulation with a photolysis reaction and output to a CSV file. This simulation uses the mechanism prepared using the model EX1, whose input and output files are shown and discussed in Example 2 in Section 3.2.11. This simulation is run using the command

```
INT EX1
```

Where all the input files are assumed to be in same directory where the command is run. The input file, EX1.INT, is shown in Example 3a. Note that this refers to the model EX1, has the results saved in a CSV file, runs for 10 minutes simulation time, and outputs data every 2.5 minutes. Since this model has a photolysis reaction, a spectral distribution needs to be specified (see Section 3.3.4), and in this example the spectral distribution is given in the file SIMPSUN.SDR, which is shown in Example 3b. The resulting CSV file containing the results of the simulation is shown in Example 3c. A simulation listing file, EX1.OUT, is also produced.

Note that no model parameter file is used in this example simulation, so the location of the prepared model files (EX1.EXE and EX1.MOD) and the spectral distribution file (SIMPSUN.SDR) must be in the same folder from which the simulation is run (the folder with EX1.INT) unless different folders are specified for them in MODELING.PRM. This is assumed not to be the case in this example.

Example 4 shows an example of a simulation with a very simple model where a parameter file is used and where the simulation output goes to a CDA file, so the data can be processed by other programs provided with this distribution. The simulation is run with the command

```
INT EX2 MODEL2
```

where model parameters are given in the file MODEL2.PRM (Example 4a). This file specifies that the model files are in the subfolder EX2 and the model to be used is EX2MOD, i.e., the prepared model files are EX2\EX2MOD.EXE and EX2\EX2MOD.MOD. The input file used to prepare these model files, EX2/EX2MOD.PRP, is shown in Example 4b. The simulation input file, EX2.INT, which specifies a 5 minute simulation time with output every minute and initial model species "A" concentration of 10 ppm is shown in Example 4c. This has the option "SAVE", which means that the simulation output data goes to a .CDA file with the same name as the input .INT file in a subfolder with the same name as the parameter file, i.e., MODEL2\EX2.CDA. This file is shown on Example 4d. It can be converted to CSV files for input into spreadsheets or read for analysis using various programs discussed in 3.5.

Example 3. Example of simple model simulation input and output files (EX1).

Example 6a: File = EX1.INT. Model simulation input

(See Example 2 for mechanism preparation input and output and Section 3.2.11 for discussion)

```
Simple example 1: A=1, SIMPSUN (PHOTK=0.6771 MIN-1)
MODEL=EX1
SAVE CSV
.
.SD-SIMPSUN
.DPRN 2.5
.TEND 10.0
A 1.0
B 10.0
.INT
```

Example 3b: File = SIMPSUN.SDR

Example spectral distribution file (See Section 3.3.4)

```
Simple approximation of Solar Z=0 Spectral Flux
0.300 0.0E+00
0.450 3.5E+19
1.000 3.5E+19
```

Example 3c: File = EX1.CSV

Model simulation output produced using EX1.PRP and EX1.INT

```
      7,EX1       ,CALCS     ,EX1       ,10/09/12 (11:08),
Minutes,A         ,B         ,P1        ,P2        ,Test-ACT  ,Test-BO   ,S         ,
    0.0, 1.000E+00, 1.000E+01, 0.000E+00, 0.000E+00, 1.000E+00, 1.000E+00, 6.771E-06,
    2.5, 1.840E-01, 9.184E+00, 8.160E-01, 8.177E-01, 1.000E+00, 1.002E+00, 1.356E-06,
    5.0, 3.387E-02, 9.034E+00, 9.661E-01, 9.685E-01, 1.000E+00, 1.002E+00, 2.539E-07,
    7.5, 6.235E-03, 9.006E+00, 9.938E-01, 9.962E-01, 1.000E+00, 1.002E+00, 4.687E-08,
   10.0, 1.148E-03, 9.001E+00, 9.989E-01, 1.001E+00, 1.000E+00, 1.002E+00, 8.639E-09,
```

Example 4. Example of simple model simulation input and output files (EX2).

Example 4a: File =MODEL2.PRM
Model parameter file

```
MOD=EX2\
DFLMOD=EX2MOD
```

Example 4b: File = EX2\EX2MOD.PRP
Model preparation input

```
Simple example 2. A -> B; B -> C
.RXN
1)  1.0  ;A = B
2)  1.0  ;B = C
.END
```

Example 4c: File = EX2.INT
Model simulation input.

```
Simple example 2
SAVE
.
.DPRN 1.0
.TEND 5.0
A 10.0
.INT
```

Example 4d: File = MODEL2\EX2.CDA
Model simulation output data

```
A8    3 EX2      CALCS    EX2MOD   08/20/13 (10:07)
A          B          C
      0.00
 1.000E+01  0.000E+00  0.000E+00
      1.00
 3.679E+00  3.679E+00  2.636E+00
     2.00
 1.353E+00  2.706E+00  5.930E+00
      3.00
 4.979E-01  1.493E+00  8.000E+00
      4.00
 1.832E-01  7.325E-01  9.078E+00
      5.00
 6.741E-02  3.369E-01  9.591E+00
```

3. PROGRAM AND FILE DETAILS

3.1. Parameter Files

Most of the modeling programs discussed here read a parameter file called MODELING.PRM to obtain default file locations and options that control the operations of the programs. This is expected to be in the local folder or directory where the programs are run, so different defaults and parameters can be used when running the programs from different locations. This file is always read by programs that use it, regardless of what options are given on the command line. Many of the model simulation programs can also obtain parameters and file locations from a model parameter (.PRM) file that is named on the command line. This contains parameters and file locations for specific models, but has the same specifications and format as MODELING.PRM. If a model parameter file is specified on the command line (or is required, depending on the program), then MODELING.PRM is read first and then the model parameter file is read and processed using the same procedure. Because the model parameter file is read last, any parameters specified there over-writes those input previously from MODELING.PRM if there is a conflict. Therefore, MODELING.PRM can give default parameters and file locations that can be over-written by the model parameter file if desired.

Parameter files are ASCII files that can be created and modified using any text editor, and the distributed files include a number of parameter files whose contents reflect the directory structure and model options used. Table 3 lists the options that can be specified in these parameter files and also gives the default options if they are not specified in either MODELING.PRM or a model .PRM file. Except as indicated on Table 3, the format is "*code*=*value*", where *code* is the option code (generally all caps) and *value* is the parameter value, and there are no spaces between the code, "=", and value. In a few cases a different format is used as indicated on the table. There is one such code and value pair for each line (record) in the file. Blank lines or lines that start with a "!" character are treated as comments. Different programs use different options, and the programs or types of programs that use each are indicated in Table 3. Parameter codes that are not recognized by the program (i.e., intended for another program or incorrectly entered) are ignored.

An examination of the source code will indicate that some programs can also obtain other parameters from these parameter files. These options are not required for the applications documented here and most have not been tested with the current version of this software. Therefore, use of parameters not listed in Table 3 is not recommended.

Examples of parameter files for chamber simulations are shown on Example 5, which shows the MODELING.PRM file and a model-specific parameter file used for model simulations of chamber. An example of a MODELING.PRM for reactivity calculations using the SAPRC-07 mechanism is shown on Example 10 in Section 3.6. All of these are included in the distributed files as of the time this document was created, but may be updated in later versions of the mechanisms.

3.2. Model Preparation

An overview of the model preparation process is given in Section 2.4.1. As shown on Figure 1, the major programs involved with model preparation are PRP, PREP and the Fortran compiler and linker that that creates the executable model simulation program from the preparation program's output. The operation of these programs and their file formats are described in the following sections.

Table 3. Program parameters and data that can be specified in MODELING.PRM or model parameter (.PRM) files

Code	Description, discussion, and defaults	Programs [a]
	File location specifications [b]	
SOURCE=	Path where to find the SUBS.LIB and INT.LIB program when compiling the model. The path specified is the location of SUBS.LIB and INT.LIB is assumed to be in a subfolder called "INT" below it. Generally this is required for model preparation.	PRP
MOD=	Path where the model's .MOD and .EXE files are located. Generally not required for model preparation because these files go in the local folder for the model, but usually required for model simulations. Also, KZSFIT uses this to determine the location for the .KZS input and .FZS output files, but it usually is not needed for this. Usually required for model simulations if they are to be done in different folders than model preparations. Model dependent.	PRP, INTs, INTRUN, REACTCAL, MROPT, INTRAT, KZSFIT
PHF=	Location of the model's photolysis (.PHF) files. Only needed if they are not in the same location as the model preparation files. Model dependent.	PREP, PHK, JZTOKZ
SDR=	Location of the .SDR files giving relative spectral distributions. Generally used when simulating indoor chamber irradiations.	PHK, INTs, LUMPINTs
JZS=	Location of the .JZS files giving solar actinic flux data as a function of zenith angle.	JZTOKZ
CAL=	Root folder for calculation results (.CDI) files. The files are located in subfolders to this named for the model's parameter file used, or in the subfolder "CALCS" if no model parameter file is specified. For example, if CAL=D:\CALFILES\ and the model .PRM file used in "CB4", then the .CDI files would be located in D:\CALFILES\CB4\. Should be model independent since the subfolder names indicate the model.	INTs, INTRUN, CDA2CSV, SUMCALCs,
INP=	Location of model-independent model simulation input (.INP) files. These are used by INTRUN to create .INT input files for the model specific model simulation program. Generally used for input files for chamber experiments or model comparison calculations.	INTRUN, SUMINP REACTCAL
PGMS=	Location of executable files called by the program. May not be needed if the programs folder (PGMS in the distribution) is on the system path, but this may not be the case if the programs are invoked using an Excel macro.	INTRUN
VARYSD=	Location of .VSA files containing spectral flux data at various times as needed for simulations of some outdoor chamber experiments.	LUMGEN, LUMPINT
MECH=	Locations of .LPM, GNA or other files referenced in the lumping control (.LPC or .LCC) files read by LUMPINT or LUMPGEN. Model dependent.	LUMGEN, LUMPINT, DMSRCT

Table 3 (continued)

Code	Description, discussion, and defaults	Programs [a]
CMP=	Location of mixture composition (.CMP) files giving relative amounts of compounds in mixtures.	LUMGEN, LUMPINT, RENORCMP, DMSRCT, REACTTAB
GDT=	Location of experimental data (.GDT) files for comparison with modeling results.	SUMGDTs, GDT2CSV
GN1=	Location of single VOC model parameter (.GN1) file, if input from such a file is specified on the command line.	LUMPINT
SAVELOC=	Location where the calculation results (.CDI) file is to be located if different than where specified in the model's .PRM file. This is ignored if specified in MODELING.PRM.	INTs
	Program control parameters	
DFLMOD=	Name of the model-specific simulation program to use. The path for this program is as specified by the MOD=parameter. This is required by INT only if it is not specified in the simulation input file, but is always required by INTRUN, LUMPGEN, and LUMPINT because they ignore the model specification in the input file. (REACTCAL uses other parameters to determine the model simulation program to use.). No default. Model dependent.	INT, INTRUN, LMPGEN, LUMPINTs
TRY2= TRY3=	Name of the model-specific simulation program to use if the one specified in DFLMOD= or (for TRY3) TRY2= fails because of input errors. Generally used to allow for certain species to require special mechanisms with additional reactions (e.g., chlorine-containing species for the SAPRC07 mechanism) without having to use different model parameter (.PRM) files for them. No defaults. Model dependent	INTRUN
LMPPGM=	Name of the lumping processing program to use when INTRUN is processing model-independent input in the .INP file to create a .INT file for the model simulation program. Can be either LUMPGEN or LUMPINT. Default is "LUMPGEN". Depends on the type of model.	INTRUN
LCC=	Name of the fixed parameter lumping control (.LCC) file used by LUMPGEN to determine model species corresponding to the compounds specified in the model-independent simulation input (.INP) file. The .LCC extension is not included in this specification.	LUMPGEN
LPC=	Name of the adjusted parameter lumping control (.LPC) file used by LUMPINT LUMPGEN to determine model species and mechanism parameters corresponding to the compounds specified in the model-independent simulation input (.INP) file. The .LPC extension is not included in this specification.	LUMPINT
DMS=	Full or relative path name of the DMS.PRM file containing molecular weights, carbon numbers, and other information on individual VOC model species. Default is DMS.PRM in local folder.	RENORCMP, DMSRCT, REACTTAB, RCTAVG

Table 3 (continued)

Code	Description, discussion, and defaults	Programs [a]
	Parameters or input used to modify or insert content into the model simulation input (.INT) files when these are being created by LUMPINT or LUMPGEN from the model-independent input (.INP) file.	
@=	Format: "@=*text newtext*" where "*text*" and "*newtext*" are character strings, and there is a single space between them. When processing .INP input files to create .INT files for model simulations, LUMPINT and LUMPGEN replace lines "@*text*" with "@*newtext*". For many of the distributed programs, including the model simulation programs created by PRP, if the input file has a line with the format "@*filename*", the input is taken from the file named by *filename* and input from the main input file is resumed once the end of file in *filename* is reached. These "@=" parameters in the .PRM files are used to control the folders where these files are located. For example, if the .INP file has a record @CMP*mixture*.CMP, and MODELING.PRM has the record "@=CMP\ D:\CMPFILES\\", then the record is output to the .INT file as "@D:\CMPFILES*mixture*.CMP". This can be model dependent, but usually is not.	LUMPINT, LUMPGEN
INTIN: *Line 1* *Line 2 (etc)*	This is optional and is used by the LUMPINT and LUMPGEN programs to add input for the model simulation input (.INT) file that is being created. This generally us used to specify non-default parameter values for simulations using a particular model parameter (.PRM) file, and is not appropriate for MODELING.PRM. This consists of at least two lines, the first being "INTIN:" and the subsequent lines being input records that are to be inserted into the .INT files. The first character in the .PRM file input lines following the INTIN: must be a space, and a record starting with a non-blank character in terminates this input. The record terminating the input can be another parameter specification for the program as listed on this table, or it could be a comment (line starting with "!"). Almost always model dependent.	LUMPINT, LUMPGEN, REACTCAL
	Miscellaneous	
!	Comment record. The rest of the input line is ignored. Note that blank records or records with unrecognized codes are also ignored.	All programs
MDC=	A charactFer string (40 max) that is included in the calculation label included in the output .CDI or .CSV files that can be used to identify the model or for other purposes. Does not affect the calculation results. Generally model dependent.	INTRUN REACTCAL

Table 3 (continued)

Code	Description, discussion, and defaults	Programs [a]
	Parameters used for reactivity calculations [c]	
MOD-BS= MOD-G= MOD-*XX*=	Specifies which mechanism executable file to use when conducting various types of reactivity simulations. Parameter value is the first part of the file name, without the extension (8-character max). MOD-BS refers to the mechanism file when conducting the base case simulations, and should be AIRLMPSB for SAPRC-07 reactivity calculations. The others refer to codes given for the test compound in the TESTHC.INS file (see Section 3.6.1).	REACTCAL
SCENPARM=	Name of the SCENARIO.PRM file which gives file names and various input parameters for the reactivity calculation scenarios. SCENPARM =SCENARIO.PRM. in this distribution.	REACTCAL
MAXCAL=	List of model species whose reactivity results are to be output, which must include ozone ("O3"). If this input is missing then O3 is the only such species. Other model species can be included, separated by spaces (no space around the "="). The distributed file include PAN and HO2H (H_2O_2) as well as O_3 in this list.	REACTCAL
RXP=	Location of the folder where input files giving the model species used for base case ground level mixture and the TESTHC input file giving model species and (if applicable) parameters for test compounds (see Section 3.6.1). Normally this is given a name that indicates the chemical mechanism used, such as "SAPRC07".	REACTCAL
MODELSET= or BASEXT=	Location of the folder where reactivity results are saved. Either of these input codes could be used. Normally this is the same folder as specified by the RXP= input, and designates the name of the chemical mechanism used.	REACTCAL
TESTHC=	Location of the file giving the processed input defining mechanisms for the test compounds whose reactivities are to be calculated (see Section 3.6.1). For SAPRC07 this is .\SAPRC07\TESTHC.INS.	REACTCAL
CLC=	Location of folder for primary reactivity output. For SAPRC07 this is .\SAPRC07\CLCFILES.	REACTCAL DMSRCT
RCT=	Location of folder for processed reactivity output. For SAPRC07 this is .\SAPRC07\CRCFILES.	DMSRCT
EXPLICIT-KOH=	Location of a file giving the OH radical rate constant for all test compounds for which reactivities are to be calculated, except for compounds whose rate constants are given in the TESTHC input file (e.g., SAPRC07\TESTHC.INS) if this is a lumped parameter mechanism (as is SAPRC-07). Must be given for all compounds if this is not a lumped parameter mechanism. This information is used to compute how much test compound to add in the incremental reactivity simulations.	REACTCAL (not needed for all VOCs for SAPRC-07)

[a] Programs where this input must be specified are underlined; otherwise this is optional. "INTs" refers to INT and any of the model-specific simulation programs created by PRP and run using INT,

Table 3 (continued)

INTRUN, or REACTCAL. "INT" means it is used only by the INT program itself, and not the model-specific programs. "*name**", refers to any program whose name starts with *name*.

[b] The default for options that refer to paths is the local directory, i.e. ".\". Except as indicated, options that refer to paths must end with a "\" character. Note that these file locations are generally independent of the model used unless indicated otherwise in the comments column. Generally the parameters that are independent of the model are more appropriately given in the MODELING.PRM file rather than the model-specific .PRM files, though this is not required.

[c] The examples of parameter values those used for calculating reactivity scales for the SAPRC-07 mechanism, for which the necessary files are all included in the distribution. Parameter values for other mechanisms would be analogous or similar.

In the subsequent discussion, the term "kinetic model" or (if obvious from the context) just "model" will refer to a specific set of model species, reactions, and related parameters that must be specified as inputs to the PRP and PREP program to create the program needed to carry out kinetic model simulations for a specific type of application. For simple chemical mechanisms and applications the kinetic model can be thought of being the same as the chemical mechanism, but complex mechanisms, such as SAPRC07, or complex applications, such as reactivity simulations, may have several kinetic models depending on the types of model species or special parameters or options needed for the application. Specific examples of different kinetic models for various applications within the SAPRC framework are given later in this document, and these can be extended to other chemical mechanisms by analogy as appropriate.

3.2.1. PRP program

The PRP program is not strictly speaking required for model preparation but it provides the most convenient way for the necessary steps to be carried out. As shown on Figure 1, it first calls the PREP program to process the input files for the kinetic model and prepare the model data (.MOD) and Fortran source (.FOR) file for the model, and then calls the Fortran compiler and linker to prepare the model's executable (.EXE) file that actually does the kinetic simulations. The details and inputs for the PREP program are discussed in the next section. The details specific to the PRP program itself and compiling and linking the model are described in the remainder of this section.

As indicated on Table 3, the PRP program reads the "SOURCE" and "MOD" parameters from the file MODELING.PRM to obtain the location of the files it needs or creates. The SOURCE parameter, which gives the location of the libraries of compiled model-independent subroutines that are linked with the output of PREP to create the model-dependent simulation program, is required, and PRP will fail if this parameter is not found in MODELING.PRM. The "MOD" parameter that gives the location of the model-specific input and output files is not needed if that location is the same as where the PRP program is run. If the users install the programs and files as recommended in Table 1, then the MODELING.PRM file in the various distributed mechanism folders will have the appropriate values and will not need to be modified. If users want to create their own mechanisms in separate folders, then simply copying the MODELING.PRM file from any distributed folder to the new mechanism folder should be sufficient, at least for the purposes of PRP and PREP.

The PRP program outputs the commands needed to compile the model-dependent subroutine created by PREP and link it with the other modules to create the model's executable program using the GNU gfortran compiler and linker. The file PRP.FOR in the SOURCE\PREP folder will need to be edited and re-compiled if a different compiler and linker is used. The commands built in to the distributed version of PRP.FOR is as follows:

Example 5. Examples of parameter files for chamber simulations

File=CHAMCALC\MODELING.PRM
Default parameters for chamber simulations

```
!Default modeling parameters
!
!Calculation results
CAL=CDTFILES\
!Characterization files
SDR=CHAR\LIGHT\
!Input files
INP=INPFILES\
CHD=CHDFILES\
GDT=CHDFILES\
!Location of executables required for simulations
PGMS=..\PGMS\
!Location of solar fluxes as a function of time for outdoor runs
VARYSD=INPFILES\
!Location of mixture files for runs with complex mixtures
@=CMP\ INPFILES\
```

File=CHAMCALC\SAPRC07.PRM
Parameters for chamber simulations using the SAPRC07 mechanism

```
!          Parameters for running simulations with detailed SAPRC07
!
! Mechanism ID added to title string for chamber simulations
MDC=SAPRC07
!
! Location of model executable and other files
! (e.g. the, .EXE and .MOD files produced when compiling the model.PRP file)
MOD=..\MECH\SAPRC07\
!
! Model executable file used by default
! (Simulation uses the mechansim in CHAMLUMP.PRP in this case)
DFLMOD=CHAMLUMP
!
! Model executable file to be used if simulation fails with the default
! model (not applicable for most mechanisms. In this case, the model
! in CHAMCL.PRP has chlorine reactions that are not in CHAMLUMP.PRP)
TRY2=CHAMCL
!
! Lumping control files used (LPC for adjustable-parameter mechansims)
LPC=..\MECH\SAPRC07\CHAMLUMP
!
! Location of lumping parameter files used by lumping control file
MECH=..\MECH\SAPRC07\
!
! Lumping program used (LUMPINT for detailed SAPRC-07, LUMPGEN for
! condensed mechanisms with non-adjustable model species)
LMPPGM=LUMPINT
!
! Location of characterization input files used for chamber simulations
@=CHAR\ CHAR\SAPRC07\
```

```
gfortran model.FOR sourceINT\INT.LIB sourceSUBS.LIB -o model.EXE
```

where *model* is the name of the kinetic model and *source* is the value of the SOURCE parameter in MODELING.PRM, which must end with a "\" character. For example, if *model* is MECH1 and *source* is "..\..\PGMS\SOURCE\", the command is

```
gfortran MECH1.FOR ..\..\PGMS\SOURCE\INT\INT.LIB ..\..\PGMS\SOURCE\SUBS.LIB -o MECH1.EXE
```

To run PRP, open a DOS window as discussed in Section 2.4.1 and give the command

```
PRP model [SAVE]
```

where *model* is the kinetic model input (.PRP) and output (.PRO, .MOD, .FOR, and .EXE) file name. The contents and format of the input (.PRP) file is discussed in Section 3.2.3, below. If the SAVE option is specified then the .FOR file is retained, but if it is omitted then the .FOR file created by PREP is deleted if the .EXE file is successfully created. The .FOR file is retained if the compile and linking command fails.

Programming considerations. The source code for the PRP program is PRP.FOR located in the PGMS\SOURCE\PREP folder in the distributed files (see Table 1). You will need to edit and recompile it if you need to change it to support the commands needed for another compiler. To do this, edit Line 119 (or the line after the comment "compile and link") in PRP.FOR as needed. Note that FILNAM(1:LFIL) refers to the model's .PRP file name, and SRCUIC(1:LSRCUC) refers to the value of the SOURCE parameter in MODELING.PRM.

3.2.2. PREP program

The PREP program is the primary program used for preparing kinetic models for simulations. As indicated on Figure 1, it takes its primary input from .PRP files (which may refer to other files for additional input), and produces the model's data (.MOD) and Fortran (.FOR) file as well as a listing output file. The format of the preparation program input files are discussed in Section 3.2.3, below. Other aspects of the use of this program are also discussed below.

The PREP program is invoked by the command "PREP *model*", where *model* is the name of the input .PRP file. The .PRP extension is not included. If *model* is not specified on the command line, the program prompts for it and it quits if a blank line is provided in response. It also quits if the named .PRP file does not exist. As indicated on Table 3, the PREP program reads the "MOD" and "PHF" options from MODELING.PRM to determine the location of the .MOD output file (MOD) or photolysis action spectra input (.PHF) files (PHF). These are not normally required since they are usually in the same folder as where PREP (or PRP, which invokes it) is run.

The PREP program does not produce any output to the user if it executes without error. This is because it is normally invoked using the PRP program, which does produce appropriate output to the user if PREP executes successfully.

Programming considerations. The source code for the PREP program is also located in the PGMS\SOURCE\PREP folder in the distributed files. The main program is PREP.FOR, and there are a number of subroutines as indicated in the file BLDALL.BAT that has the commands to compile and link it. Note that this is one of the first programs written when the programming of this software system was started in the mid-1970's, the program has been modified over the years to run on a number of different computers, and there is a certain amount of obsolete code in the source files. Note also that the program reads other parameters in MODELING.PRM besides those listed in Table 3, but they are no longer used and have not been tested in the current version of the software.

If very large kinetic models are to be processed it may be necessary to change some array dimensions for both the PREP program and also the model simulation program. The relevant array dimension specifications are in the file PSPECS.FOR, which is referenced by most of the PREP subroutines using the Fortran "INCLUDE" statement. Comments in PSPECS.FOR indicate which maximum array size parameters need to be changed for the various purposes. It is important that if any of these are increased the corresponding array size parameters for the model simulation programs to also be increased. Otherwise the prepared kinetic model executable will be unusable. These are in the file MAINSPC.FOR in the folder PGMS\SOURCE\INT. Once these specifications in PSPECS.FOR or MAINSPC.FOR are modified, all the subroutines in the programs need to be re-compiled. This can be done by running BLDALL.BAT in the respective source folders.

3.2.3. Format of Model Preparation Input (.PRP) Files

The preparation program input (.PRP) file is a text file that consists of lines that are read sequentially by the preparation program in order to obtain the parameters, commands, and input data it needs to prepare a model. The following applies not only to preparation program input files but also input files for the model simulation programs as discussed in Section 3.3.1.

- Lines should be no longer than 80 characters. Longer lines may be accepted in some instances but this is not documented so longer lines should be avoided.
- Lines starting with the "!" character are treated as comments and are ignored. This is the case for input files for almost all programs discussed here, except as indicated otherwise.
- If a line starts with a "@", the rest of the line is expected to be a file name from which subsequent input lines are read. This is referred to as the "indirect input file". The full name of the file (with extension) must be given and if it is in a different folder than where the command is given then the absolute or relative path name must also be given. The format of the data in the indirect input file is the same as in the main input file, and once the end of the file of the indirect input file is reached the input from the main input file resumes. For example if the .PRP input file has a line "@BASEMECH.RXN", the file BASEMECH.RXN in the same folder as the .PRP file is read for the subsequent lines. Once the end-of-file for BASEMECH.RXN is reached, then the next record in the .PRP input file is read. Indirect input files can also have @file lines to take input from other indirect files, though there is a limit to how many levels of indirect files can be used, and use of more than two levels is not recommended.
- The line "@#SH" by itself will cause subsequent input lines to be echoed to the user as the line is read, together with a prompt indicating which type of input is being expected. This is not normally used but sometimes is useful for debugging.
- Except as indicated for certain types of inputs, blank lines are not accepted and will generally cause the program to fail.
- Except as indicated, codes or parameter names are case-sensitive. Use of upper case is required for codes that control program operation or input. Model species or coefficient names or reaction labels can have lower case characters but these are also case-sensitive. For example, the model species "xC" is treated as a different model species than one named "XC".

For SAPRC mechanisms we use the convention of giving indirect input files used in preparation program input (.PRP) files the extension ".RXN". However, this is just a convention and any extension can be used for indirect input files, since the extension must be explicitly given when these files are used. Simple models generally don't need indirect input files, and their use is not required even for complex models, though often they are useful for organizing portions of the mechanisms.

The input expected by the PREP program consists of four groups of input lines, each of which is expected. These are as follows:

- The title line. This consists of a single line that can be used to identify the model. Its contents are ignored by the program except it is output in the listing (.PRO) and data (.MOD) file. A blank line is acceptable here.
- Options and parameter Input. These consist of any number of lines (zero is acceptable) containing options or parameters that control the operations of the program. The parameters that can be input are listed in Table 4. Some parameters consist of a single keyword, while others consist of a keyword and a value, separated by one or more space characters or a tab character. The keywords must be upper case and the parameter values must be valid integers or floating point numbers, depending on the type of parameter. These are terminated by a line that start with a "." character, which is treated as the first line of the model input, discussed below.
- Model input. These consist of any number of groups of lines (at least one or no model is produced) that define the reactions, species, and other aspects of the model being prepared. Each model input group consists of a ".*code*" line, which starts with a "." character to indicate that it is the start of a new group, and *code* is a code (upper case required) indicating the type of group. The input following it depends on the type of group, and in most cases can consist of any number of lines. This group input is terminated by a line that starts with a "." character, which can either a "." by itself, another group code starting a new model input group, or the .END line. Table 5 summarizes the codes that are recognized and the types of input that follow them. Specifications for reactions, model species, and coefficients or parameters are discussed in more detail below.
- The .END line. This consists of a single line ".END", which terminates the input. Any lines following that are ignored.

3.2.4. Reaction and Kinetic Parameters, and Species Specifications

Table 5 includes a summary of inputs specifying the reactions and rate constants in the model. The main specification of this is in the .RXN inputs, which consists of at least one, and sometimes more depending on the types of kinetic parameters and numbers of products formed in the reaction. The format of the reaction input is as follows:

Label) *kinetic parameters* ;*Reactants* = *products*

Where the characters ")", ";", and "=" are required to delineate between the label, kinetic parameters, reactants, and products. The input lines should be no longer than 80 characters, so the product listing can be continued in subsequent lines if necessary. Some types of kinetic parameters also require parameters be specified in subsequent input lines. This is discussed further below.

Label. The reaction label is used to identify the reaction in cases where it needs to be referenced in kinetic parameter input for another reaction or when the capability of changing kinetic parameters during model simulations is desired. It can optionally be blank, in which case PREP generates a reaction label. Note that the ")" character is required even if the input label is blank -- in that case the ")" is the first nonblank character in the reaction input line. The label must contain no blank characters or a ")" or ";" and be no more than 4 characters long. Duplicate labels result in a warning message, but the model is usable as long is there is no need to reference the affected reaction labels in the simulation input. It is recommended that either all labels be unique, or that blank labels be used when the reaction label is not considered an important part of the model, and kinetic parameters are not to be varied from their default values during the simulations.

Table 4. List of options and parameters that can be specified in the parameter input group in model preparation (.PRP) input files.

Code [a]	Description
TEMP *Value*	Default temperature for model simulations, where *value* is a temperature in degrees K. The default is 298.12 K (25 C). This has no effect if the kinetic model has no rate constant expressions that depend on temperature.
TREF *Value*	Reference temperature for certain types of temperature-dependent rate constant expressions, such as $k(T) = A \cdot \exp(-Ea/RT) \cdot (T/TREF)^b$, where T is the temperature. Default is 300 K, and this is usually used by most rate constant evaluations. This has no effect if the model has no rate constant expressions that use this parameter. Note that although it is allowed to use TREF=1, this should not be done when the units conversions option "UNITS=PPM", discussed in Table 5, is used.
MAXVCO *Value*	Maximum number of variable coefficients. *Value* must be an integer between 1 and the maximum number of variable coefficients, which is set in the program's dimensions. Coefficients are parameters in the model that can be used for product yields and other purposes, and can be either variable (named coefficients that can be changed by appropriate inputs to the simulation programs) or constant (input as numbers that can't change). See Section 3.2.6 for a discussion of the various types of coefficients. The default and maximum number of variable coefficients are currently set at 400 and 8000, respectively. This default is satisfactory for most models, but higher defaults are sometimes needed for kinetic models with large numbers of generalized reactions with variable product yield parameters, such as AIRLMPMX in the SAPRC07 mechanism folder.
VARYHV	Allows for the light intensity to be varied in the simulations. The species "HV" is added as a parameter with a default value of 1, and this parameter can be varied to allow for light intensity of less than 100%, i.e., HV=0 means dark. For this to work, photolysis reactions must be specified as a reaction with HV, i.e. "NO2 + HV = (products)". This does not need to be specified if the ".AIRSHED" option is specified. If this is not specified, the species "HV" is treated as a constant species. It can be given a value that is different than 1 during the model simulations, but it cannot be varied with time during the simulation, as it could if the VARYHV option were specified.
AIRSHED	Adds additional parameters and features needed so the kinetic model can be used in ambient simulations with emissions input and variable mixing heights. Also allows light intensity to be varied in the simulations (implies selection of the .VARYHV) parameter.
SAVEPHF	Saves any photolysis action spectra (absorption cross section and quantum yield) data that are given in the .PRP file using .PHOT input in photolysis .PHF files. Only applicable if the .PRP file (or indirect input files referenced by it) contain .PHOT input. This is not applicable to any models in the distributed files. The default is not to output .PHF files if .PHOT input is used. This does not affect the use of the kinetic model since the action spectra data are saved in the .MOD file in any case, and that's where the simulation program obtains these data.

Table 4 (continued)

Code [a]	Description
	Undocumented or unsupported parameters.
FORMAT=*any* SAVEPHF DEBUG	These parameters were used in previous versions of the program and are still in the source file PREP.FOR. However, the non-default options of the FORMAT parameter and the SAVEPHF parameter are not compatible with the distributed version of the program, and should not be used, and the DEBUG parameter has no useful effect.

[a] Upper case required.

Kinetic parameters. The rates of the reactions in the model simulations are obtained by multiplying the rate constant for the reaction times the concentrations of all the reactants and the values of any parameters specified in the list of reactants (see below). The kinetic parameter input provides for multiple means to derive the rate constants given the temperature, pressure, light characteristics, or derived parameter values depending on the type of reaction. The types of kinetic parameter inputs, and their format requirements, are summarized in Table 6. As indicated there, in most cases the rate constant or the parameters used to derive the rate constant are given on the first line of the reaction input, between the ")" character that signifies the end of the reaction label and the ";" character that signifies the start of the reactant list. The various parameters can be separated by spaces (any number) or commas, and their meanings depend on the type of reaction as indicated on Table 6. Temperature and pressure-dependent rate constants require too many parameters to be included in the space between the label and the list of reactants, and in those cases the parameters are input in a separate line following the reactant and product list. In those cases, the first line contains the codes "FALLOFF", "K1+K2[M]", or "K0+K3M/1+K3M/K2" (case sensitive, no embedded spaces), to indicate falloff kinetics or other types of special temperature and pressure dependences, with their parameters given in subsequent two or three input lines following the reactant and product specification.

Note that the rate constants are pressure dependent if "FALLOFF", "K1+K2[M]", or "K0+K3M/1+K3M/K2" codes are used. The pressure is specified using the constant model species "M", which must be specified, which is the concentration of all species present, including air if applicable. Normally it should be given a default value of 1.0e+6 for 1 atmosphere if ppmV units are used. If there is no model species "M" or if it has a concentration of zero, then the rate constant for FALLOFF reactions are always zero, and those of the other types are not calculated assuming [M]=1 in the concentration units used, which usually is not appropriate.

The default the rate constants (k) and Arrhenius A factors (A) on Table 6 are in ppm, minute units but they can be in molecule, cm, second units if the UNITS=PPM option (see Table 5) is used. In the latter case, the input units are converted to ppm, minutes units for the with the simulation program. The exact units depend on the order of the reaction, which is determined by the number of reactants that are model species and not coefficients as discussed below. The units of activation energies (Ea's on Table 6) are kcal/mole by default but can be deg K if the UNITS=EA:DEGK option is specified. Note that the units of k and A are not affected by the UNITS=EA options and the units of Ea are not affected by UNITS=PPM option. The parameters B on Table 6 are unitless, but if TREF=1 then the units of A are changed depending on the value of B. For example, if TREF=1 and B=1, and the reaction is first order, default units are min^{-1} $degK^{-1}$. However, if TREF=300^{o}K (or some other temperature, which is not recommended), then the units of A are the same as the units of the rate constant.

Table 5. List of codes and types of input that can be specified in the model input portion of the model preparation (.PRP) input files.

Code [a]	Description
.	Null input. Used to terminate the previous model input set and put the input process in a known state. All subsequent lines are ignored until a line that starts with a "." is encountered.
.RXN (input lines follow)	Reaction input. At least one such input group must be included in the input file or the model will not be processed. Subsequent lines consist of *Label) Kinetic parameters* ;*Reactants = Products* [+ *more products (if applicable)*] [*Additional kinetic parameters (if applicable)*] The reaction label is optional (but the ")" is required, the kinetic parameter input is summarized in Table 6, the reactants and products consist of species names separated by "+"'s, and products can be optionally specified with yield specifications. See the discussion in Section 3.2.4 for a more complete discussion of this input.
.ACT .BLD .CON .STS .DUM (input lines follow each)	Used to indicate the types of model species in the kinetic model, if default types are not to be used. As discussed in Section 3.2.4, these refer to active, buildup-only, constant, steady-state, or dummy species, respectively. For constant species (.CON input) the subsequent input consists of the species name and its default concentration, separated by spaces or a tab. For the other types, the subsequent input lines consist of list of model species, either one per line or in lines such as "= *species1* + *species2* + …, where *species1*, *species2* are model species names. It does not matter if these input lines come before or after the .RXN inputs giving the reactions where they are involved.
.COE (input lines follow)	Used to give default values to variable (i.e., named) coefficients. Subsequent lines are "*name value*", where *name* is the coefficient name and *value* is its numeric (real) value. It does not matter if the value of the coefficient is given before or after the reaction(s) using it are input, and coefficients do not necessarily have to product yield parameters. Certain special coefficients are built into the software, as discussed in Section 3.2.6, and the .COE input can be used to specify their values if the defaults are not to be used.
.UNITS=PPM .UNITS=OK	Used to specify the units for the kinetic parameters that will be input in subsequent (but not preceding) .RXN input groups. No subsequent lines are associated with these, and any lines following them that do not start with a "." are ignored. See the discussion of kinetic parameter input in Section 3.2.4. Briefly, the default is for concentrations to be in ppmV units, and .UNITS=PPM means that the subsequent rate parameter inputs are consistent with concentrations being in molecules/cm^3 units, and need to be converted with ppmV-compatible units before being input to the simulation program. .UNITS=OK means that the units are already in ppmV compatible units and do not need to be converted. Do not use this input if TREF=1.
.UNITS=EA:DEGK .UNITS=EA:KCAL	Changes the units of the activation energies in the kinetic parameter inputs in subsequent .RXN input. See the discussion of kinetic parameter input in Section 3.2.4.

Table 5 (continued)

Code [a]	Description
.UNITS=KNOTA .UNITS=ANOTK	Changes the way that Arrhenius temperature-dependence parameters are specified in subsequent (but not preceding) .RXN input. See the discussion of Arrhenius parameter input on Table 6.
.INS INIT .INS DIFF .INS SAVE (input lines follow each)	These are used to insert Fortran code into the various model-specific subroutines to perform operations or calculations that are beyond the scope of the automatically generated code. This is described in more detail in Section 3.2.11. The subsequent input is assumed to be Fortran code, with special codes used to designate concentrations or parameter values as given in Table 8. This is terminated by an input line with a "." as the first character, which can be the code for the next type of input.
.SUM *name1 name2*	Creates the code so the concentration of the dummy species *name1* is set equal to the concentration of the active species *name2* + the concentrations of all the other active species in .SUM *name1* ... inputs. The species *name2* must be active, not constant, buildup, steady-state, or dummy. Note that *name1* must be specified as a dummy species using separate .DUM input (either before or after the .SUM input) -- the .SUM command is not automatically added to the dummy species list. There must be one .SUM input line for each of the active species whose concentrations are summed up.
	Note also that if there is a reaction of a member of the sum group with the sum group (as with a peroxy + peroxy reaction represented by peroxy + total peroxy), the rate of the reaction is calculated from rate (member+sum) = k x [member] x ([sum] + [member]), rather than k x [member] x [sum]. This is because of the nature of the differential equations for self-reactions.
.NODIL (input lines follow)	The program creates a DILUTION or .DIL parameter to account for first order loss of all species by dilution during the simulations, if a nonzero dilution is specified in the input. This .NODIL option can be used if it is desired that the simulations to include model species that are not lost due to dilution. The following lines should consist of lists of model species in the same format as used for the .ACT, etc, inputs. This is useful for some types of counter species such as those used for calculating integrated concentrations.
	The following input is only valid if the AIRSHED option is specified in the parameter input field (see Table 4).
.DEP (input lines follow)	Used to input names of species that are assumed to undergo deposition. The following lines should consist of lists of model species in the same format as used for the .ACT, etc, inputs. A "D#*species*" coefficient is created to specify the deposition rate of the listed species. See Section 3.2.9 for more details.
.EMI (input lines follow)	Used to input names of species that are assumed to undergo emissions, but are not part of an emissions group specified using the .GROUP command (see below). The following lines should consist of lists of model species in the same format as used for the .ACT, etc, inputs. A "E#*species*" coefficient is created to specify the emissions rate of the listed species. See Section 3.2.9 for more details.

Table 5 (continued)

Code [a]	Description
.GROUP *name* (input lines follow)	Used to input names of species that are treated as being emitted together, with the same emissions schedule and fraction present initially. The total initial + emitted amounts of species in this group are given in a single parameter, with other parameters specifying the relative concentrations of individual species in the group, how much is present initially, and its emissions schedule (see Table 7). See Section 3.2.9 for more details. The following lines should consist of lists of model species in this group in the same format as used for the .ACT, etc, inputs.

Note that the conversions of units using the UNITS=PPM option depends on the order of the reaction determined by the number of reactants that are model species (as opposed to coefficients -- see below), and does not work correctly for the current version of the software for zero order reactions or when TREF=1. The failure to convert units for zero order reactions is a bug in the PREP program thatmay be corrected in future versions. The incompatibility with TREF=1 is because the units conversion changes the "B" parameter depending on the order of the reaction because the conversions of concentrations and rate constants from molecular to ppmV units depends on temperature based on the assumption that the "B" parameter does not affect the units of the A factor, which is not the case if TREF=1.

The correct conversion of the units depends on the order of the reaction, which is determined by the number of reactants that are model species. As indicated in the discussion of the .RXN input in Table 5, the reactants in a reaction are specified by the list of names between the ";" and "=" characters in the .RXN input, separated by "+"'s. Those that are indicated by "#*name* are coefficients whose values are multiplied by the rate constant derived by the kinetic parameter input to determine the rate constant actually used, while those without the "#"'s are model species reactants, the number of which determines the reaction order. Reactant input can contain anywhere between 1 and 4 coefficients or model species, of which zero to 4 can be model species. This means that reaction orders can be anywhere between zero and 4, though zero and 4th order reactions are not included in any of the example input files. Note, however, that the model preparation listing output (.PRO) file only lists the index numbers of the first 3 reactants or coefficients (with negative index numbers being used for coefficients), though the computer code in the .FOR file uses all 4 reactants when calculating reaction rates where applicable.

Reactants. The reactants input consists of the list of reacting model species or coefficients, terminated by a "=", and separated by a "+". The name of the first reactant should immediately follow the ";", without spaces. Single spaces are permitted around the "=" or "+" symbols, but multiple spaces are not allowed within the reactant (or product) lists. There must be at least one reacting model species but no more than three. A coefficient (model parameter) can be used instead of a model species, in which case it is indicated by "#*name*", where *name* is the name of the coefficient. In this case, the rate constant expression derived from the kinetic parameters is multiplied by the coefficient value to give the rate constant used. This allows use of a single named coefficient to change rate constants for multiple reactions.

Products. The products input follows immediately after the "=" that terminates the reactants input. It consists of a list of model species names and optionally product yields separated by a "+". The model species and yield input can have one of the following formats:

Specification	Meaning
+ *species1* + *species2* + …	One mole of a products named *species1*, etc is formed per reaction.

Table 6. Types of rate constant or kinetic parameter input that can be specified for reactions in the .RXN input.

Type	Format [a]	Discussion [b]
Constant	*k*	The rate constant is the constant and not dependent on temperature or pressure. The rate constants are in ppm, minute units by default unless the UNITS=PPM option is in effect (see Table 5), in which case they are in molecule, cm, second units. The exact units depend on the number of non-coefficient reactants as discussed in the text.
Standard extended Arrhenius temperature dependence	*A, Ea, B* or *k, Ea, B*	The rate constant, $k = A \cdot \exp(-Ea/RT) \cdot (T/TREF)^{B}$ By default this input is A, Ea, and B, but it can be k, Ea, B if UNITS=KNOTA (see Table 5Table 5) is in effect. In that case, A is calculated from the other input parameters. The units of k and A are the same, as discussed above and in the text. B can be omitted and is zero by default. Ea is in units of kcal/mole by default and R=0.0019872 kcal $mole^{-1}$ $degK^{-1}$ by default, but if UNITS=EA:KCAL (see Table 5) is in effect, then the units of Ea are degK and R=1. Ea can also be omitted if B is omitted, but in that case it is the same as a constant rate constant as discussed above. The reference temperature TREF is specified in the parameter input as shown in Table 4. It is normally 300K or one, and is 300K by default. If TREF=1 then $k = A \cdot \exp(-Ea/RT) \cdot T^{B}$
Falloff (depends on temperature and pressure)	FALLOFF *A0, Ea0, B0* *AI, EaI, BI* *F, N*	$k = \{k0 \cdot [M]/(1 + k0 \cdot [M] / kI)\} \cdot F^{Z}$, where $k0 = A0 \cdot \exp(-Ea0/RT) \cdot (T/TREF)^{B0}$, $kI = AI \cdot \exp(-Ea0/RT) \cdot (T/TREF)^{B0}$, and $Z = \{1 + [\log10\{k0 \cdot [M])/kI\}/N]^{2} \}^{-1}$ The B's and Ea's can be omitted and are zero by default. See above for information on the units of the Ea's and TREF. The model must have defined a model species named "M" that gives the total bath gas concentration.
Special T, P dependence #1	K1+K2[M] *A1, Ea1, B1* *A2, Ea2, B2*	$k = k1 + k2 \cdot [M]$, where $k1 = A1 \cdot \exp(-Ea2/RT) \cdot (T/TREF)^{B1}$, and $k2 = A2 \cdot \exp(-Ea2/RT) \cdot (T/TREF)^{B2}$ See the comments above for "falloff" reactions.
Special T, P dependence #2	K0+K3M/1+K3M/K2 *A0, Ea0, B0* *A2, Ea2, B2* *A3, Ea3, B3*	$k = k0 \cdot [M] + \{k3 \cdot [M] \cdot (1 + k3 \cdot [M] / k2)\}$, where $k0 = A0 \cdot \exp(-Ea0/RT) \cdot (T/TREF)^{B0}$, $k2 = A2 \cdot \exp(-Ea2/RT) \cdot (T/TREF)^{B2}$, and $k3 = A3 \cdot \exp(-Ea3/RT) \cdot (T/TREF)^{B3}$ See the comment above for "falloff" reactions.

Table 6 (continued)

Type	Format [a]	Discussion [b]
Rate constant is the same as that for another reaction	SAMEK *label*	The rate constant is the same as that calculated for a previously input reaction, where *label* is the label for that reaction. It cannot refer to a reaction that has not yet been input. The kinetic parameters for that reaction must be of a type listed above on this table (not below).
Rate constant is the same as the rate of another reaction	CASEOF *label*	The rate constant is the same as the rate (rate constant x reactant concentrations) calculated for previously input reaction, where *label* is the label for that reaction. It cannot refer to a reaction that has not yet been input. The kinetic parameters for that reaction must be of a type listed above on this table, except SAMEK. This is not used in the distributed model preparation files.
Photolysis reaction	PF=*PFname* or PF=*PFname* QY=F or PF=*PFname* FAC=F	The rate constant for this photolysis reaction is calculated using the absorption cross sections and quantum yields in *PFname*.PHF and the light intensity and spectral distribution specified in the model simulation input, and is zero by default. "QY=F" or "FAC=F" is optional and if given then F is the overall quantum yield that is multiplied by the photolysis rate calculated using the data and the .PHF file and the light intensity and spectrum input to give the photolysis rate. The default for F is 1. See Section 3.3.4 and 3.3.5 for a discussion of how these data are used to calculate photolysis rates in model simulations.
Rate constant is coefficient	COEF=*coef* or COEF=*coef* FAC=*n*	The rate constant is taken from the value of the coefficient or parameter with the name *coef*. This can be useful because coefficients can vary with time or be manipulated in other ways. If "FAC=*n*" is given the rate constant is the floating point number *n*.
No longer supported	LTEMP *n*	This type of kinetic parameter input is included in the program but is not in current mechanisms and is not supported.

[a] Codes that are underlined must be entered as exactly shown (case sensitive, no internal spaces). "*Label*" refers to a reaction label; *PHFname* refers to the name of a photolysis (.PHF) file and "*Coef*" refers to a coefficient or parameter name, and any other symbols refer to numeric parameters that are used to calculate the rate constant as indicated in the discussion. Commas are used here to separate the numeric parameters, but spaces could also be used. The first line consist of parameters on the first line of the reaction input between the ")" and ";", and the subsequent lines, if present, occur after the last line giving the reactant and product input.

[b] The following symbols are used in the equations o calculate the rate constant given the parameters given in the "Format" column: k is the rate constant that is being calculated; T is the temperature in degrees K; [M] is the concentration of the model species called "M", which is expected to be a constant species giving the total bath gas concentration in the concentration units being input (i.e., $[O_2]+[N_2]$ for atmospheric simulations); TREF is the TREF parameter specified as given in Table 4,and is 300K by default, and R is the gas constant in units consistent with kcal/mole and degrees K (R=0.0019872). If the UNITS=EA:DEGK option is specified (see Table 5), then R=1; i.e., R should be removed from the equation. "Log" means natural logarithm, and log10 means log to the base 10.

+ #*yield1 species1* + #*yield2 species2*+ …	The number of moles of the products named *species1*, etc, are given by the number or coefficient named *yield1*, etc.
+ #*yield* {*species1* + *species2* + …}	The number of moles of the products named *species1, species2*, etc are given by *yield.*
+ #*yield1* {+ … + #*yield2 species3* + …}	Similar to the above, but the product yield for the product named *species3* is given by *yield1*yield2*.

Note that there must be a single space between the yield specification and the species name. Single spaces around the "+" symbols are optional. These types of specifications, separated by "+"'s, can be combined.

The #*yield* specifications can be either a number (e.g, "#2", "#.5", etc.) or a valid coefficient name (up to 8 alphanumeric characters or "_" or "-"). If the former it is a fixed yield and cannot be changed without re-preparing the model. If it is not decodable as a number then it is a coefficient with this name, whose value can be specified using .COE input (Table 5). If no .COE input is given for this coefficient name, its value is zero by default. Note that negative yields are allowed by the PREP and model simulation software, though simulations may fail if computed concentrations become negative.

If the list of products is too long to fit on one 80-character line, then the line can end with a "+" character, and the list of products can continue on the next line(s). These reaction continuation lines must start with one or more blank characters. If the first or continuation line does not end with a "+" character then that is the last line listing the products.

3.2.5. Model Species

As indicated on Table 5, above, model species can be either active, buildup-only, constant, steady-sate, or "dummy". These species types can be specified using the .ACT, .BLD, .CON, .STS, or .DUM commands, respectively as shown on Table 5, but these specifications are not needed if the default species types are acceptable. If no species type specifications are given, then the species within the various types are output in the order they are input in .RXN inputs in the model preparation input files. The species specifications can be used to control the order they are output if a different ordering is desired. The various types of model species, and the defaults for these types, are summarized below.

Species Names. Species names can be 2-8 characters long (no longer), contain only alphanumeric characters and the additional characters "_", "-", and ".". Embedded blanks are not permitted, and use of lower case characters, though allowed, is discouraged. Although the Fortran programs are case-sensitive, it is recommended that only upper case characters be used so there is no ambiguity if programs that are not case sensitive are used. Although species names can be up to 8 characters, it is recommended that no active model species have names longer than 5 characters if the AIRSHED option is used. This is because this option creates parameters named "A#*name* for all active species, and can create parameters named "D#*name*" or "R*n*#*name*" if the species are named in .DEP or .GROUP lists, and the names of these parameters can be no longer than 8 characters. This would result ambiguities if more than one active model species had the same first 5 or 6 characters in their names. In such cases, only the parameters corresponding to the species that is first input during model preparation can be modified at simulation time.

Active Species. Active species are model species whose changes in concentrations over time are computed with the greatest accuracy that the solver can provide. By default any species that is a reactant in some reaction is treated as active. The amount of computer resources required for the simulations increases greater than linearly with the number of active species in the model.

Buildup-Only Species. Buildup-only species are species that are not reactants in any reaction but are formed in some reaction, and where it is acceptable that their concentrations be simulated with lower levels of accuracy in order to reduce simulation times. By default any species that is not a reactant in any reaction but is formed in some reactions is treated as buildup-only. If users requires accurate simulations of concentrations of such species, they need to specify them as active using the .ACT command. The .BLD input should only be used to change orderings of buildup-species that are output, never to force species that react to be buildup-only. If this is done, the simulation results will not be valid. To avoid this possibility, use of .BLD input in model preparation input files is not recommended.

Constant Species. Constant species are species whose concentrations are either treated as constant during simulations or whose concentrations at various times are given explicitly in the input files. No species are constant by default, so if constant species are desired for a model then they must be specified using .CON input in the model preparation input file. The default concentrations of constant species can also be specified in the preparation input file. This is the only type of model species where the default for the model simulations can be something other than zero. Constant species can appear as reactants or products in reactions, but they only have effects on simulations as reactants. Typically constant species that are present in such high concentrations that they do not change during the simulation, such as O2, M, or H2O in ambient simulations, are treated as constant to reduce the number of active species required, while allowing them to be specified and participate as reactants.

Steady State Species. This software allows for using the steady-state approximation to calculate the concentrations of certain species, provided that they are specified as such using the .STS input and other conditions, discussed below, are met. In this approximation, the species concentration is calculated by assuming that the total rate of formation of the species due to reactions where it is a product is equal to its total rate of consumption. If a species is specified as steady state but is not a reactant in any reaction, the program changes it to buildup-only, and the approximate method used for such species is used to calculate its concentrations. If it is not formed in any reaction then it is changed to constant and is essentially ignored in the model simulation. The steady state approximation is only valid for very rapidly reacting species whose rates of formation or loss greatly exceed their rates of change. When valid it can speed up the simulations by reducing the number of active species and also might reduce the mathematical "stiffness" of the simulation if applied to the most rapidly reacting of the species. No species are steady state by default.

The software imposes other limitations on types of species where the steady state approximations can be used because of limitations in the procedure it uses to calculate steady state concentrations. The procedure cannot handle situations where a steady state species reacts with itself or with another steady state species, so this approximation cannot be used for such species. When the code to calculate steady state concentrations is to be executed in the model simulation program, the concentrations of all relevant types of species must be known except for steady state species. To derive code to calculate steady state concentrations, the preparation program orders the steady state species so that the concentration of the first can be calculated independently of any other steady state concentration, the second only depends on the first, if any, etc. This is not possible in situations where there are cyclic dependences among steady state species, such as, for example, S1 → S2; S2 → S3; and S3 → S1, etc, where the steady state approximation is to be used for all three. In this example situation the steady state approximation can be applied for any two of them but not all three. Otherwise, the preparation process will fail because the steady state species cannot be ordered.

If the user is unsure whether the approximation can be used for selected species in their model, the easiest approach is to try it first then remove selected species from the .STS list if the model preparation process fails. It is also advisable to compare test calculations with and without using this approximation to assure that its use for the chosen species is not affecting the results.

Dummy Species. Dummy species are similar to constant species except their concentrations are output in the simulation output files, they cannot appear as a reactant in any reaction, and their formation as a product is ignored when processing .RXN input. Their concentrations can be only changed only by using "inserted instructions" (see Section 3.2.11) where the user can manually enter Fortran-like code to calculate their "concentrations" or values from concentrations of other species or coefficient values. These calculated dummy species concentrations or values are included in the simulation output files where they are available to the user. This differs from constant species whose concentrations as a function of time are not output. As with constant species, dummy species can be used as reactants or products in reactions, though they only affect the simulations when they are used as reactants. Using them as products in a reaction does not affect the concentrations of dummy species that are output, so although this does not result in an error it has no effect on the simulation. No species are dummy by default.

Species Names. Species names can be no longer than 8 characters and can include any alphanumerical character (upper or lower case, but case-sensitive) and the special characters "_", "-", or ".". The software allows for a few other special characters, but their use is not recommended for compatibility with future versions of the software. Spaces, double or single quotes, commas, and the characters "=", "#", "{", "}" should definitely never be used, and a period "." should never be the first character. It is legal for the first character to be numeric, but it is not recommended that the entire name be decodable as a number since this may cause problems in certain circumstances and generally will cause confusion. Some model simulation software systems do not permit the first character to be numeric and do not allow non-alphanumeric characters in species names, so this is not recommended if the intention is to use this software to develop mechanisms that will be used by other modeling systems.

If one examines the source code, one may find that the specifications of arrays of species names in some of the older programs permit for names of up to 16 characters. But this is not supported by all programs and newer options or modules added to the older programs, so more than 8 characters should not be used.

Species names should also not include any of the reserved or special coefficient names discussed in the following section. These are listed in Table 7, and the names are case sensitive. The species names "M" and "HV" (also case sensitive), if used in the model, should be reserved for use as constant species to represent total bath gas concentration (M) and fraction of maximum light intensity (HV). It is also a good idea to reserve the name "H2O" for water vapor, though this is not used as a special type of species in the current version of the software.

3.2.6. Types of Coefficients

One feature of the SAPRC modeling software system that may not be present in other systems is the use of arrays of "coefficients" to specify model-dependent parameters. These are most commonly used to represent product yields in reactions in cases where they are something other than one, but can have other uses as discussed in this section. As indicated in Section 2.3.2, there are two types of coefficients, constant and variable. Constant coefficients refer to real numbers that appear as product yields in reaction input (see discussion of .RXN input in Table 5) as numbers, and cannot be changed. Variable coefficients have both names and values, and although their default values are defined in the model preparation input, the names can be used to change their values in the model simulation input files. The names and values associated with them can be specified in the model preparation input using the .COE input as discussed in Table 5. They are also defined, with default values of zero, when they appear in reaction listings in .RXN input, using the term "#*name*, where *name* is the coefficient name and cannot be decoded as a real number. If it decodable as a real number, then the coefficient is constant and it does not have a name.

Table 7. Special variable coefficients created by the model preparation program.

Name	Default	Description
		Special parameters that are always created
DILUTION	0. min^{-1}	Dilution rate in min^{-1}. Dilution is represented by active model species being consumed with this unimolecular loss rate coefficient.
TEMPR	298. ^{o}K	Simulation temperature in degrees K.
LITTLE	10^{-30} ppm	A parameter used by the simulation program to indicate a negligible concentration for the purposes of error control.
		Following created when the <u>VARYHV</u> or <u>AIRSHED</u> option is specified.
HV	1.	Light intensity as a fraction of the maximum. All photolysis rates are multiplied by this value during the model simulations. HV=0 can be used to specify dark conditions. If there is a constant model species called "HV", it is converted to a variable coefficient. See Section 3.3.5.
		Following created when the <u>AIRSHED</u> option is specified. Section 3.2.10.
EMIT.FAC	0.082056 *TEMPR	Factor used to convert emissions input, normally given in flux units (moles per unit area) to concentration units. The default is based on flux units in millimoles $meter^{-2}$ and concentration units in ppm.
HEIGHT	1.	Inversion height. Units depend on EMIT.FAC but is meters by default. Used in treatments of deposition, to convert emissions given in flux units to concentration units and to determine input of aloft pollutants (if any) and/or dilution as the inversion height increases.
(E.RATE)	0.	Used internally in the simulation program for calculation of dilution or entrainment of aloft pollutants due the inversion height (HEIGHT) increases. Should not be specified as an input in simulation input files.
E#*species*	0.	Emissions rate of species named *species*, in units that depend on EMIT.FAC (millimoles $meter^{-2}$ min^{-1} by default). Created if the species is named after an .EMI command (Table 5).
A#*species*	0.	Concentration of species named *species* aloft, which is entrained into the simulated mixture when HEIGHT increases. Created for all species if AIRSHED is specified, but ignored if HEIGHT does not increase.
DEP.FAC	1.	Conversion factor for calculation of deposition rates given the inversion height (HEIGHT). Created only if .DEP input is specified. Default is 1, meaning that the unimolecular loss rate due to deposition is given by the deposition rate input (D#*species*) divided by HEIGHT .
D#*species*	0.	The deposition rate of a species named in the .DEP input. Units depends on DEP.FAC and units of HEIGHT.
		Parameters created when a .GROUP name command is input (see Table 5), where *name* is the name of the group.

Table 7 (continued)

Name	Default	Description
TOT.*name*	0.	Total inputs of all pollutants in group *name*. Units are emissions units, depending on EMIT.FAC (millimoles meter^{-2} by default). This can include initial concentrations (depending on the INI.*name* parameter) as well emissions during the simulation, but does not include aloft entrainment. See Section 3.2.10.
INI.*name*	1.	Fraction of pollutants in group *name* that is present initially.
EMR.*name*	0.	The hourly emissions rate in relative units. This is used to specify how emissions change with time needs to be properly normalized based on the values of INI.*name* and the amount of time in the simulation.
R*n#species*	0.	Moles of species named *species* in a group per mole input specified for the group. The number *n* is the group number, where *n*=1 refers to the group named in the first .GROUP command, *n*=2 the second, etc. The species are listed following the .GROUP command in the PREP input.

Variable coefficient names can have no more than 8 characters and can include any alphanumeric characters plus the characters "_", "-", "(", ")", ".", and a few others, whose use is not recommended. They can definitely not contain a space, quotes, or the characters "#", ",", "+", "=", "}", "{", and "&", and a "." cannot be the first character. Lower case characters are permitted in coefficient names, and these names are case-sensitive. It is permissible for the first character to be numeric, but this is not recommended. The coefficient name should not be decodable as a number unless it is a constant coefficient in a product list in the .RXN input for a reaction.

There are several variable coefficients that have special meanings as defined by the software, and there are other special variable coefficients that are created by the PREP program when various options and inputs are processed. These are listed on Table 7, along with a brief discussion of how they are used. In addition, the PREP program creates a variable coefficient for each photolysis file that is referenced in the preparation input (named in the "PF=*PFname*" kinetic parameter input as indicated on Table 6) with the same name of as the photolysis file (e.g., coefficient name is *PFname*). These special coefficients are used to allow photolysis rates to be explicitly input (optionally as a function of time) in the model simulation input files, as an alternative to specifying the actinic flux and light intensity input, and having the program calculate the photolysis rates from the data in the photolysis files. Therefore, coefficients used for other purposes in a model cannot be the same as the names of any of the photolysis files used in the model.

3.2.7. Photolysis Reactions and Photolysis Data Files

A photolysis reaction is any reaction given in the model preparation input file with "PF=*phfname*" included in its kinetic parameter input (see Table 6), where the absorption cross section and optionally quantum yield data are in the file *phfname*.PHF. Although this is not required by the software, it should be given either as a simple first order reaction (e.g., "NO2 = O + NO") or as a pseudo second-order reaction with "HV" (e.g., "NO2 + HV = NO + O"), where "HV" is a scaling factor used to represent overall light intensity. The use of "HV" is optional but is useful because it allows for easily turning lights off, simply by specifying "HV 0.0" in the model simulation input. But if "HV" is used, it must also be specified in the model preparation input as a constant (.CON) species with a default value of 1, so it's "concentration" is simply a unitless scaling factor. If it is not specified as a constant species, "HV" would be treated as a chemical reactant that is consumed as photolyses occur, which is not a physically realistic situation. The mechanism preparation software does not do this by default.

The recommended procedure for processing photolysis reactions is to give the absorption cross section and quantum yield data in photolysis data (.PHF) files during model preparation, and specify the light spectrum and intensity in the simulation program input files. The specification of the light spectrum is discussed in Section 3.3.4 for artificial light sources with constant spectra and in Section 3.3.5 for time-varying solar light. The simulation program then uses these data to calculate the photolysis rates for the given intensity and spectrum. The format of the photolysis data files as read by the PREP program is discussed in this section. The formats of the spectral distribution data files and actinic flux data used as input to the model simulation program are discussed in Sections 3.3.4 and 3.3.5, respectively.

Photolysis data (.PHF) files are plain text files with lines no longer than 80 characters. The first line is a comment or documentation line that is output in the .PRO and .MOD files but is otherwise ignored. The subsequent lines can be as follows:

- A comment line, which is any line that starts with a "!". This line is ignored.
- A line with "FAC=*factor*" where *factor* is a number greater than zero. All absorption cross sections input after this line are multiplied by this number. Absorption cross sections input before this line are not affected. The default value is 1, so this input is not needed if the absorption cross section data are in units of cm^2 $molec^{-1}$ (base e).
- A line with two or three numbers, separated by spaces, commas, or tab characters. The first number is a wavelength in microns, the second is the absorption cross section in units of cm^2 $molec^{-1}$ (base e) (or some other units that are converted to these units when multiplied by the number given in the "FAC" input line), and the third is optional and if given is the quantum yield at that wavelength. If the quantum yield is not given, then the quantum yield for that wavelength is assigned a value of one.
- A blank line. This terminates all input and all subsequent lines are ignored. This is not necessary and the input can also be terminated by the end of the file.

The absorption cross section and quantum yield data must be given in order of ascending wavelength. Note that if the wavelength range for the spectral distribution or actinic flux input to the simulation program exceeds the wavelength range of the absorption cross section and quantum yield data in the .PHF files, the absorption cross section and quantum yield for the lowest wavelength is used with spectral data for lower wavelengths, and those for the highest wavelength is used with spectral data for the highest wavelength when calculating photolysis rates. Therefore, the photolysis data files should always have data for wavelengths lower than the low wavelength limit of spectral distributions or actinic fluxes to be used in any of the calculations, and the highest wavelength in the photolysis data file should be the high wavelength cutoff for the reaction, and have an absorption cross section or quantum yield of zero. Otherwise the simulation program will not calculate correct photolysis rates.

The current version of the program allows for no more than 1000 wavelengths to be input in a .PHF file, but this can be increased by editing the parameter MAXAQY in pgms\source\prep \SAVERX.FOR. The simulation program calculates the photolysis rates by assuming that the actual action spectrum consists of straight lines between each of the wavelengths given (sometimes called the "points" method). The PREP program saves the (wavelength, absorption cross section x quantum yield) data in the .MOD files for each of the wavelength input.

The use of the absorption cross section and quantum yield data to calculate photolysis rates for model simulations is discussed in Sections 3.3.4 and 3.3.5.

3.2.8. Preparing Photolysis Data for Ambient Solar Simulations and Files Used

The model simulation programs that are prepared by the PRP program include a capability of calculating photolysis rates for solar simulations using actinic fluxes provided as a function of solar zenith angle by Peterson (1976). If use of this built-in capability is satisfactory, then no additional model preparation processing is needed to conduct ambient solar simulations.

However, additional model preparation processing is required if it is desired to conduct model simulations of ambient sunlight scenarios using actinic flux data produced by other light models, or specified by the user. This is discussed in Section 2.4.3, and the programs and files involved are also shown on Figure 1. For this purpose, the actinic fluxes as a function of wavelength and zenith must be given in a .JZS file, whose format is discussed below. The JZTOKZ and KZSFIT programs are then used to calculate the photolysis rates and prepare them for the model simulations as discussed in Section 2.4.3. The files involved are discussed below.

Lightmodel.JZS Files. These files that give actinic flux values as a function of wavelength and solar zenith angle. These files are independent of the chemical mechanism used but can depend on the scenario being modeled. Generally they are produced by some sort of light model, a discussion of which is beyond the scope of this document. These are input files that must be supplied by the user, though several examples are provided among the distributed files with this software system. These are in the REACT/INPFILES folder as indicated in Table 1, along with other input related to ambient scenario conditions. The distributed files include the following:

- PETERSON.JZS … These are the actinic fluxes given by Peterson (1976) for his "best estimate" surface albedos. These are the basis of the built-in photolysis rate calculation procedures included in the simulation programs, though the photolysis rates calculated using the built-in procedures are slightly different than calculated using this file because of the somewhat different calculation procedures used (see Section 3.3.5).
- STDZA640.JZS … These are the actinic fluxes provided by Jeffries (unpublished data, 1991) for the purpose of the calculations of the Carter (1994a) reactivity scales. They are calculated using an unpublished light model (Jeffries, personal communication, 1991) for an altitude of 640 meters. This was also used for calculating updated scales using the same ambient scenarios with newer versions of the mechanism (Carter, 2010, 2010b). Note that the output files used for the reactivity calculations prepared using STDZA640.JZS are renamed STD640.
- STDZA000.JZS … These are the actinic fluxes provided by Jeffries (unpublished data, 1991) using the same light model as employed to prepare STDZA640.JZS but for ground level.

These are all text files that have four sets of input lines, as follows:

1. The first set of lines consist of comments that are ignored by the program, and optionally a correction factor line. If there is a correction factor, the first four characters must be "FAC=" (case sensitive) and the remaining characters must be decodable as a positive number. If this is given, all the actinic flux values are multiplied by this number.
2. A line with a "." in column 1 is used to indicate the end of the comments and factor lines.
3. The next line or set of lines gives the number of zenith angles and the zenith angles, separated by spaces. The first number is the number of zenith angles, which is normally 10. It can be less than 10 but it is recommended that 10 zenith angles be used for sufficient resolution. If more than 10 is desired then the program dimensions will have to be changed and JZTOKZ, KZSFIT, and the model simulation programs would have to be re-compiled. The next numbers are the zenith angles, which typically are 0, 10, …, 70, 78, and 86. These can be on any number of lines; the

program reads lines until the specified number of zenith angles are input. There can be more than zenith angles input than the number specified, but in that case the additional angles are ignored and there can be no more input lines number required to input the specified number of zenith angles

4. The next set of lines give the wavelengths (in microns, not nm) followed by the actinic fluxes for each of the input zenith angles. There can be more than one line for each wavelength, and lines for each wavelength are read in until the specified number of actinic fluxes are input for the wavelength. There can be more than this number of actinic fluxes as long as they are not on additional lines, and these data are ignored. There is one set of these input lines for each wavelength whose actinic fluxes are to be input. The wavelengths must be in ascending order, and there can be no more than 400 without re-compiling JZTOKZ. The range of wavelengths input must range from the lowest wavelength that has non-negligible fluxes up to the highest wavelength with nonzero action spectra, which are 1000 nm or 1 micron for most models used with this software.

The actinic flux values that are input is the photons/cm^2 in the wavelength range between the input wavelength and the wavelength input before it. For the first wavelength input, the wavelength interval is assumed to be the same as for the 2nd wavelength input, i.e., if the first and second wavelengths input are 300 and 305 nm, the interval for the first is assumed to be 295-300.

Lightmodel.KZS files. These files contain photolysis rates as a function of zenith angle, with the photolysis reactions being identified by the name of the .PHF file that contains the absorption cross section and quantum yield data, as discussed in Section 2.4.3. These are normally output by the JZTOKZ program and are given the same name as the .JZS file used as input, but it should be noted that they depend on the chemical mechanism as well as the light model or scenario. Therefore these files would normally be in the same folder as the .PHF files used to create them, to avoid confusion with .KZS files prepared using the same light model but for different chemical mechanisms.

The format of these files is essentially the same as that for the .JZS files discussed above except that there is no "FAC" input, names of the photolysis files (which can be no more than 8 characters long) are given instead of wavelengths, and photolysis rates in units of min^{-1} are given rather than actinic fluxes.

Normally these would be output files that would not needed to be prepared or edited by the users. However, they can be used in applications using incomplete models where absorption cross section and quantum yield data are not specified.

Lightmodel.FZS files. These files contain the parameter used to calculate photolysis rates as a function of zenith angle for a given light model and set of photolysis reactions. These are used as inputs to the simulation programs using models with these photolysis reactions and this light model. These are normally output by the KZSFIT program and are given the same name as the .KZS file used as input (which usually is the same as the name of the corresponding .JZS file), and like .KZS files depend on both the chemical mechanism and the light model or scenario. Therefore these files would normally be in the same folder as the .KZS and PHF files used to create them, to avoid confusion with .FZS files prepared using the same light model but for different chemical mechanisms.

These files are generally not intended to be edited or modified by the user. Lines starting with "!" are comments that are not used by the simulation program. The comments include comments copied from the input files, the zenith angles and photolysis rates in the input KZS file, the photolysis rates calculated using the fitting parameters, and the percentage errors in the calculations. The first non-comment record gives the optimization method, which is always "ABEF78" with the current version of KZSFIT.

Following that there is one non-comment record for each reaction, giving the name of the PHF file, the A, B, E, and F parameters used to calculate the photolysis rate, the cosine and degrees of Z^{max}, the zenith angle above with a different method is used to calculate the photolysis rate, and a measure of the quality of the fit. The method used to calculate the photolysis rates from these parameters is described in Section 3.3.5.

3.2.9. Treatment of Dilution and Reactant Injections without the AIRSHED option

Although the AIRSHED option, discussed in the following sections, gives the most flexibility and options for representing ambient scenarios, it is possible to represent both dilution and emissions in simpler models without the complexities of this option. This is discussed below.

Dilution. All prepared models have a parameter called DILUTION that gives the pseudo-unimolecular rate constant of reactants by dilution. The default value of this parameter is zero (unless it is explicitly given in the .COE input in the mechanism preparation file), and it is usually specified as part of the simulation input. Dilution is represented for all active and buildup-only species, except those listed in the .NODIL input (Table 5). (It is not used for steady-state species, based on the assumption that other loss rates for such species are much faster than dilution.) The units are $time^{-1}$, or min^{-1} by default. If the AIRSHED option is specified, this is in addition to any dilution due to inversion height increases. However, if the AIRSHED option is specified, dilution can also be represented increasing the inversion height during the simulation, as discussed in Section 3.2.10 below. If this approach is used, the value of the DILUTION parameter should be kept at zero.

Reactant Injections. There are no provisions for specifying emissions as such unless the AIRSHED option is used. However, for simple cases, such as modeling the continuous injection of reactants during an environmental chamber experiment or modeling flow experiments, injections of selected species can be added as a pseudo-reaction, where the species is represented as being formed from nothing in a zero-order reaction, with a rate coefficient that has units of concentration per unit time. The system does not permit inputting reactants without anything in the reactants field (see Section 3.2.4), but zero order reactions can be input using a coefficient as a reactant. Therefore, one way to represent a continuous NO injection, with the injection rate in units of ppm/minutes (the default concentration and time units) being specified by a parameter we'll call "*EMITNO*" (any name can be used), this can be representing by including the following in the PREP input:

label) 1.0 ;#*EMITNO* = NO

The rate coefficient of 1.0 means that the value of the value of the coefficient *EMITNO* is the emissions in the appropriate zero order units, but a different value could be used if different time units are desired. For example, if it is preferred to give inputs in concentration/hour units, then the "rate constant" should be given a value of 1/60 so that the rate of the reaction would be in the minute time units used in the simulation. Note that the units of injection rates in this example is different than that used for emissions input if the AIRSHED option is specified, since it refers concentration changess in a volume and not fluxes through an area.

3.2.10. Treatment of Emissions, Deposition, and Entrainment with the AIRSHED Option

Although this software cannot be used for multi-cell, 3D model calculations, it can be used to represent a two-layer "floating box" model with a variable inversion height, emission and deposition of pollutants, and entrainment of pollutants from aloft if the AIRSHED option is specified. If this is specified, the conceptual model is a reacting parcel consisting of fixed surface area on the ground going up to the inversion height, and an unreacting "aloft" layer that may contain pollutants that would be entrained into the reacting ground layer if the inversion height increases during the simulation. PREP generates the parameter HEIGHT to give the inversion height, but there is no parameter specifying the area of the ground surface of the parcel because it is not needed in the simulation. It can be thought of having unit area, which means that the volume of the reacting parcel would be given by HEIGHT parameter. It also creates parameters for specifying emissions and deposition rates, as described below.

Emissions. Reactant concentrations that are simulated in the model give moles per unit volume, but the conceptual model behind the AIRSHED option requires specifying emissions in terms of moles per unit ground surface area per unit time. This is because emissions are thought of coming from the surface or items fixed distances from the surface, and are generally not affected by the inversion height (i.e., volume). Given constant emissions rates, the moles of pollutant injected into the reacting parcel will be independent of the mixing height, the concentrations will go down as the inversion height increases because of the increased dilution. This is a more realistic conceptual model for ambient simulations than specifying emissions in concentration/time units. However, the model simulation software requires changes in emissions be given in concentration/time units, as discussed below.

The changes in concentrations due to emissions are calculated from the emissions rates as follows. If Moles is the number of moles of the pollutant in the reacting parcel, Conc is the concentration in moles per unit volume, HEIGHT is the inversion height, and A is the ground (and top) area of the parcel, E is the emissions rate of a reactant in moles per unit area per unit time, and D is the deposition velocity in units of distance per unit time, and t is simulation time, then

$$\text{Conc} = \text{Moles} * A^{-1} * \text{HEIGHT}^{-1}$$

$$\text{d(Moles)/dt}^{\text{Emit}} = E * A$$

$$\text{d(Conc)/dt}^{\text{Emit}} = \text{d(Moles)/dt}^{\text{Emit}} \cdot A^{-1} * \text{HEIGHT}^{-1} = E / \text{HEIGHT}$$

Note that the area of the parcel on the ground, A, cancels when computing the change on concentration due to emissions, if emissions are quantified as moles per unit area. Normally this software uses ppm for the units of the simulated concentrations, while emissions fluxes are millimoles per meter^2 per unit time. If 1 ppm is thought of as 10^{-6} atm, then ppm = 10^6 R·T·C, where R is the gas constant (8.206e-5 meter^3 atm degK^{-1} mol^{-1}), T is the temperature, and C is the moles per liter. If E is in units of millimoles meter^{-1} per unit time, then

$$\text{d(ppm)/dt}^{\text{Emit}} = \text{EMIT.FAC} * E / \text{HEIGHT}$$

where EMIT.FAC is a conversion factor 10^6 R·T, or 24.63 at 300°K. This is how the code generated by PREP calculates the change in concentrations due to emissions. The default conversion factor generated by PREP is 24.63, which is appropriate for 300°K, but is generally used regardless of the temperature of the scenario being simulated. A different value of EMIT.FAC can be specified at simulation time if different units are used for emissions rates or simulated concentrations.

Emissions rates for single species are given by the parameters "E#*species*", where *species* is the species name. Such a parameter is created for each active species if the AIRSHED option is specified, with a default value of zero (species not emitted). These parameters should be given in units of millimoles meter^{-2} min^{-1} at simulation time if the default value of EMIT.FAC is used. Note that HEIGHT must also be defined and be at a representative value for the scenario being simulated, or the concentrations resulting from the fluxes that are specified will not be correct.

Group Emissions. Simulations of ambient scenarios require emissions of many pollutants, which are often grouped together. For example, hundreds of different compounds are emitted in realistic scenarios, and tens of model species are used to represent them, depending on the lumping approach employed, and their inputs in models generally specify the total amounts of VOCs emitted separately from input giving the moles of model species used per unit of total VOC emissions, i.e., the VOC speciation. In addition, oxides of nitrogen (NO_x) are generally emitted as a mixture of NO and NO_2, and in some scenario HONO is also assumed to be present at some level in NO_x emissions. Generally emissions inventories give total amounts of NO_x emitted but not the speciation, so the speciation in terms of relative amounts of NO, NO_2, and (if applicable) HONO, needs to be specified separately.

The .GROUP command (Table 5) can be used to define groups of emitted species in order to make it more convenient to specify total emissions and speciation equally. The PREP input lines

.GROUP *group*
= *species1* + *species2* + …
...

define a group named *group* and list the model species contained in it (named *species1, species2,* etc). For example, the input lines could be .GROUP NOX and then = NO + NO2 + HONO, or they could be .GROUP HC and then lines listing all the model species used to represent VOCs that might be emitted, which in general may take several lines. The following special coefficients are then added to the model for each such input group: TOT.*group*, INI.*group*, EMR.*group*, and the a coefficient named R*n#species* in added to each species in the group, where *n* is the group number, which is 1 for the first group, 2 for the 2nd, etc. Note that for the airshed scenarios used for reactivity scales (discussed in Section 3.6.2) the convention is to use NOX for group 1, HC for group 2, and THC (test compound or mixture in reactivity simulations) group 3, but this is not required by the software.

The parameters defined by the .GROUP command can also be used to define initial reactant concentrations in addition to (or instead of) emissions. This is useful for scenarios where a fraction of the "emissions" are assumed to be present initially as carry-over from previous day, or are emitted at night before the reactions start. The parameter INI.*group* specifies the fraction of species in the group that are present initially. In this case, the conceptual model is that this fraction of the group is instantaneously "emitted" at the start of the simulation into a volume defined by the initial value of the mixing height (HEIGHT) in the simulation.

The software then computes the code for calculating changes in concentrations due to emissions as follows:

$$\mathrm{dMoles/dt}^{\mathrm{Emit}} = \mathrm{TOT.}\mathit{group} * \mathrm{EMR.}\mathit{group}(\mathrm{t}) * \mathrm{R}\mathit{n\#species}$$

$$\mathrm{d(ppm)/dt}^{\mathrm{Emit}} = \mathrm{dMoles/dt}^{\mathrm{Emit}} * \mathrm{EMIT.FAC} / \mathrm{HEIGHT}$$

$$\mathrm{d(ppm)/dt}^{\mathrm{Emit}} = \mathrm{EMIT.FAC} * \mathrm{TOT.}\mathit{group} * \mathrm{EMR.}\mathit{group}(\mathrm{t}) * \mathrm{R}\mathit{n\#species} / \mathrm{HEIGHT}$$

and likewise computes the initial concentrations as:

$$\mathrm{Initial}\ \mathit{species} = \mathrm{TOT.}\mathit{group} * \mathrm{INI.}\mathit{group} * \mathrm{R}\mathit{n\#species} * \mathrm{EMIT.FAC} / \mathrm{HEIGHT(0)}$$

where HEIGHT(0) is the mixing height at the start of the simulation. Note that the INI.*group* parameter, the fraction of the group that is present initially, implies that fraction of the group that is emitted during the simulations should be 1 - INI.*group*. However, the software implementation is such that the fraction of the group that is emitted is given by

$$\mathrm{Fraction\ emitted} = \mathrm{INI.}\mathit{group} * \Delta\ \mathrm{time}$$

where Δ time is the number of minutes in the simulation if INI.*group* is constant, or by the integral of INI.*group* over time during the simulation if INI.*group* varies with time. Therefore, the value(s) of INI.*group* should be normalized such that

$$\text{Fraction Emitted} = \text{Integral (EMR.}\textit{group}\text{) over time} = 1 - \text{INI.}\textit{group}$$

Note that the software does do this normalization nor does it check it, so the user should do this normalization when preparing the model inputs.

It is recommended that TOT.*group*, INI.*group*, and the R*n*#*species* be treated as constants throughout the simulation, with EMR.*group* being used to represent how emissions vary with time. The recommended units for of TOT.*group* is millimoles meter^{-2} per simulation, INI.*group* should be a unitless number between 0 and 1, and the integral of EMR.*group* over simulation time should be 1 - INI.*group*.

Deposition. Deposition refers to the loss of a reactant on a surface, with a rate that is calculated by multiplying a surface area associated with the scenario times a deposition velocity that has units of distance per unit time. If the AIRSHED option is specified, then coefficients named D#*species* is created for all active species for specification of their deposition rates. They are all zero by default, but nonzero defaults can be defined using .COE input in the model preparation input. The rate of change of the calculated concentration of the species due to deposition is given by:

$$\text{D(Moles)/dt}^{\text{Dep}} = -\text{D\#}\textit{species}\cdot \text{A}\cdot\text{C} \qquad \text{Conc} = \text{Moles / A} * \text{HEIGHT}$$

$$\text{dConc/dt}^{\text{Dep}} = \text{d(Moles)/dt}^{\text{Dep}} \;/\; \text{A} * \text{HEIGHT} = -\text{Conc} * \text{D\#}\textit{species}\text{ / HEIGHT}$$

This means that the pseudo-unimolecular loss rate at any given time in the simulation is D#*species* /HEIGHT. Note that the surface area of the parcel, A, cancels out. The units of D#*species* should be distance per minute, where the distance unit should be the same as used for HEIGHT. This parameter is independent of the units used for concentration or moles.

Entrainment. As mentioned above, the conceptual model if AIRSHED is specified includes an aloft layer containing pure air or reactants that can be entrained into the reacting parcel if the mixing height of the airshed (the HEIGHT parameter) increases. A parameter named A#*species* is created for each active species to specify their concentration aloft, which is zero by default. Note that no changes are made to concentration of species in the reacting layer due to entrainment if HEIGHT is constant or decreasing with time, and that changes due to entrainment are only affected by the current value and rate of change of HEIGHT, not its history of past values.

In order for entrainment to be calculated properly, the parameter HEIGHT must not have any discontinuities, and must be specified using the "L#" (linear interpolation) option of .VARYCO input at simulation time (see Table 10 and Section 3.6.2). This input involves giving values of the parameter at specified times, and computing the their values at intermediate time due to linear interpolation. This input is used to derive the rate of change of HEIGHT with time [d(HEIGHT)/dt], since processing "L#" input requires computing the rate of change for each time segment where height is input. Therefore, the change of concentration of any species due to time is given by

$$\text{dConc/dt}^{\text{Entrain}} = \text{Max (0, d(HEIGHT)/dt)} * (\text{A\#}\textit{species} - [\textit{species}])\text{ / HEIGHT}$$

where [*species*] is the concentration of the species in the reacting parcel. Note that reactants that are not present in the aloft layer are diluted at a rate given by Max (0, d(HEIGHT)/dt) / HEIGHT.

3.2.11. Inserted Instructions

One of the major functions of the PREP program is to create the Fortran code to calculate rates of changes of concentrations of active species during the course of the simulation, and perform other model-specific computations. As discussed above, the PRP program calls PREP to create this Fortran code and then calls the compiler and linker to crate the model-dependent simulation program using this code. In most cases the user need not be concerned with this Fortran code, but there are cases where explicitly inserting model-specific Fortran code is useful in implementing features in the model that otherwise would not be available. This might include calculations of initial concentrations of some reactants that may depend on initial concentrations of other reactants, calculations of rate coefficients or product yields that may depend on concentrations of species, calculations of "dummy" species concentrations from concentrations of other species or coefficient values, etc. Examples of inputs and applications where inserted instructions may be useful are given in Section 3.2.11.

As indicated on Table 5, there are three types of inserted instructions that can be created, depending on which part of the model simulation process the instructions are to be executed. These are as follows.

- .INS INIT is used for instructions that are to be executed during the initialization process, i.e., after all the initial concentrations and parameter values are input before any concentration changes due to chemical reactions are calculated. This code is executed every time an .INT command in the model simulation input file (see Section 3.3.1) is processed, which is usually only once during a simulation unless it is a multi-part simulation. A typical case would be to insert code that calculates initial concentrations of species that depend on initial concentrations of other species.
- .INS DIFF is used for instructions that are to be executed every time rates of change of active species due to chemical reactions are to be calculated, which occurs at each time step in the simulation, and also when the Jacobian matrix (derivatives of rates of changes with respect to concentrations of active species) is calculated. This code is executed multiple times during the simulations. A typical case would be to compute parameters values that depend on other species or parameters that change during the course of the simulations and that affect rates of reactions and the species concentrations that are being simulated.
- .INS SAVE is used for instructions that are to be executed whenever concentration-time data are to be output. This is more appropriate than .INS DIFF for code that only affects data at the particular time and does not affect calculation results. A typical case would be to output dummy species created to give functions of variable parameters or concentrations (such as species ratios, etc.) that may be of interest to include in simulation output files for diagnostic or other purposes.

The input for the Fortran code in the inserted instructions can be any type of legal Fortran code except that line numbers cannot be used, the code can start in the first column, and continuation lines are indicated using a "&" character at the end of the line(s) to be continued, not a character in column 6. Input lines cannot be longer than 80 characters, so input lines need to be continued if more than 80 characters is required in a line of Fortran code. The list of codes that can be used to refer to model-specific or simulation data is given in Table 8. Not all of these can be used in all cases, as indicated in the table. PREP converts those to the Fortran code for the appropriate array element or parameter reference for the type of data being referenced.

3.2.12. Example Inputs for Adjustable Parameter Mechanisms

SAPRC-07 Detailed Mechanism Example: Chamber Simulation Model. Example 6a shows a model preparation input file that could be used for chamber evaluations of a single VOC species using the SAPRC-07 detailed mechanism. Comments in the file indicate how the input is used. It refers to several .RXN files that have components of the mechanism. One of these is BASEMECH.RXN, which has the reactions of the inorganic species, explicitly represented organic compounds, organic product model species and chemical operators representing peroxy radical reactions. Example 6b shows portions of BASEMECH.RXN showing examples of reactions with various types of kinetic parameters, and also

Table 8. Codes used to indicate model-specific data in inserted instruction input.

Codes	Refers To	Restrictions
'*CoefName*' or "# *CoefName*"	Variable coefficient with name *CoefName*.	No restrictions
[*Species*] or "C *Species*"	Concentration of species named *Species*.	No restrictions except steady-state species should only be used with .INS SAVE.
"K *RxnLbl*"	Rate constant for reaction with label *RxnLbl*	Use only with .INS DIFF
"R *RxnLbl*"	Rate of reaction with label *RxnLbl*	Use only with .INS SAVE
"D *Species*"	Rate of change of species named *Species*	Use only with .INS SAVE for active or buildup-only species
TEMP	Simulation temperature	No restrictions
TIME	Simulation time	No restrictions.

portions of the mechanisms used to compute rates of reactions of the various chemical operators used by SAPRC-07 to represent effects of peroxy radical reactions. The SAPRC-07 mechanism and reactions, model species, and chemical operators are described elsewhere (Carter, 2010a); the purpose of showing these examples is to illustrate how the software is used to implement some of its components.

One of the features of how this software implements the SAPRC-07 detailed mechanism is the use of generalized model species and reactions whose rate constants and product yield parameters are assigned based on the compound (or mixture of compounds) being represented. The file LUMPGEN1.RXN contains the reactions of GEN1, one of these generalized model species, and Example 6c shows selected portions of this file. The model simulation input file is expected to provide the rate constants or kinetic parameters for its reactions with OH, O_3, O^3P, NO_3 and photolysis and the variable coefficients giving the yields of all the possible products of these reactions. For this purpose, reactions are labeled "G*nRx*" and product yield coefficients are labeled "G*nRxPd*", where *n* refers to the generalized reactant number (1 in this case), *Rx* is the 2-character reactant code (OH, O3, etc.) and *Pd* is the 2-character product code. These are produced by the lumping processing programs as discussed in Section 3.4. Note that for detailed model species that undergo photolysis, the parameter such as "G1HV*pfcode*" are used to indicate the action spectrum to use, where *pfcode* is a two-character code that indicates the type of action spectrum, and the quantum yield is indicated by "G*n*HVQY". For example, for a ketone where PF=MEK-06 should be used, the parameter "G*n*HVpk" is set to 1.0, and all the others are kept at the default value of zero. The .INS INIT code shown in Example 6c implements this approach. Note that this terminology for the reaction labels and coefficient names is based on the requirements of the lumping processing programs; they do not have any special meanings as far as the PREP program is concerned.

3.3. Model Simulations

As discussed in Section 2.4, the model simulations are carried out using the model-specific simulation programs that are prepared as discussed in the previous section. These programs can only be used for the particular model being prepared, but allow for the user to vary many of its rate constant and other parameter values, as well as reaction conditions such as temperature, dilution, and light intensity and spectrum and initial reactant concentrations and (for models allowing emissions) emissions rates.

Example 6. Example of portions of model preparation input for the SAPRC-07 baseline mechanism with generalized reactions for one added detailed model species. This example is suitable for evaluation of the mechanism against results of environmental chamber experiments.

Example 6a: Main preparation input (.PRP) file

```
NOx - AIR REACTIONS OF ONE ADJUSTABLE MECHANISM COMPOUND
! This is needed because LMPGEN1.RXN has a large number of variable coefficients.
MAXVCO 550
! Default temperatures
TEMP 300.0
TREF 300.0
! Force order of first 3 active species
.ACT
= O3 + NO + NO2
! Allow "INERT" model species to be input without error
.COE
INERT0.0
!
! Input portions of mechanism. Convert units from molec, cm, sec to ppm, min
.UNITS=PPM
! Input base mechanism
@BASEMECH.RXN
! Input generalized mechanism for one test compound
@LMPGEN1.RXN
! Input chamber wall model
@WALLS.RXN
!
! Added for computing D(O3-NO) = {O3(T)-NO(T)} - {O3(0)-NO(O)}
.DUM
= D(O3-NO) + INIT_NO + INIT_O3 + FIRST
.INS INIT
IF ([FIRST].EQ.0.0) THEN
   [INIT_NO]=[NO]
   [INIT_O3]=[O3]
   [FIRST]=1.0
ENDIF
.INS SAVE
[D(O3-NO)] = ([O3] - [INIT_O3]) - ([NO] - [INIT_NO])
!
! Added for computing integrated OH (ppt-min units)
.NODIL
INTOH
.UNITS=OK
.RXN
IOH)CONST 1.0E+6;OH = OH + INTOH
.END
```

Example 6b: Selected portions of BASEMECH.RXN

Portions showing various types of reactions and kinetic parameter input (SAPRC-07 mechanism)

```
.RXN
1)     PF=NO2-06              ;NO2 + HV = NO + O3P
2)     5.68e-34 0.000 -2.60   ;O3P + O2 + M = O3 + M
22)    FALLOFF                ;OH + NO = HONO
       7.00e-31 0.000 -2.60
       3.60e-11 0.000 -0.10
       0.60 1.00
27)    K0+K3M/1+K3M/K2        ;OH + HNO3 = H2O + NO3
       2.40e-14 -0.914 0.00
       2.70e-17 -4.370 0.00
       6.50e-34 -2.653 0.00
```

Example 6 (continued)

```
29)   K1+K2[M]               ;OH + CO = HO2 + CO2
      1.44e-13 0.000
      3.43e-33 0.000
BR22) 5.20e-13 -1.947 0.00   ;MECO3 + HO2 = CCOOH + #.7 O2 + #.3 O3
BR32) SAMEK BR22             ;RCO3 + HO2 = RCOOH + #.75 O2 + #.25 O3
BE06) 2.54e-11 -0.815 0.00   ;ISOPRENE + OH = #.986 RO2C + #.093 {RO2XC +
                              zRNO3} + #.907 xHO2 + #.624 xHCHO + #.23 xMACR +
                              #.32 xMVK + #.357 xIPRD + yR6OOH + #-0.167 XC
```

Portions showing computation reaction rates of of SAPRC-07 peroxy radical operators

```
.RXN
BR07) 2.60e-12 -0.755 0.00   ;RO2C + NO = NO2
BR08) 3.80e-13 -1.788 0.00   ;RO2C + HO2 =
BR09) 2.30e-12               ;RO2C + NO3 = NO2
BR10) 2.00e-13               ;RO2C + MEO2 = #.5 HO2 + #.75 HCHO + #.25 MEOH
BR11) 3.50e-14               ;RO2C + RO2C =
BR25) 4.40e-13 -2.126 0.00   ;MECO3 + RO2C = MEO2 + CO2
...
RO01) COEF RO2RO             ;xHO2 = HO2
RO02) COEF RO2XRO            ;xHO2 =
...
.INS DIFF
'RO2NO' = "K BR07"*"C NO"
'RO2HO2' = "K BR08"*"C HO2"
'RO2NO3' = "K BR09"*"C NO3"
'RO2RO2' = ("K BR10" * "C MEO2") + ("K BR11" * ("C RO2C" + "C RO2XC"))
'RO2RO3' = "K BR25" * ("C MECO3" + "C RCO3" + "C BZCO3" + "C MACO3")
'RO2RO' = 'RO2NO' + 'RO2NO3' + 'RO2RO3' + 0.5*'RO2RO2'
'RO2XRO' = 'RO2HO2' + (0.5*'RO2RO2')
```

Example 6c: Portions of LMPGEN1.RXN
Reactions of generalized VOC species

```
.RXN
G1OH) ;GEN1 + OH = #G1OHN2 NO2 + #G1OHHO OH + #G1OHRH HO2 + #G1OHME MEO2 +
       #G1OHQ1 MECO3 + #G1OHQ2 RCO3 + #G1OHQM MACO3 + #G1OHQB BZCO3 +
...
       #G1OHp7 xPRD7 + #G1OHRR yROOH + #G1OHR6 yR6OOH + #G1OHRA yRAOOH +
       #G1OHXC XC + #G1OHXN XN
G1O3) ;GEN1 + O3 = #G1O3N2 NO2 + #G1O3HO OH + #G1O3RH HO2 + #G1O3ME MEO2 +
...
G1N3) ;GEN1 + NO3 = #G1N3N2 NO2 + #G1N3HO OH + #G1N3RH HO2 + #G1N3ME MEO2 +
...
G1OP) ;GEN1 + O3P = #G1OPN2 NO2 + #G1OPHO OH + #G1OPRH HO2 + #G1OPME MEO2 +
...
G1HV) COEF=G1HV QY=G1HVQY ;GEN1 + HV = #G1HVN2 NO2 + #G1HVHO OH + #G1HVRH HO2 +
       #G1HVME MEO2 + #G1HVQ1 MECO3 + #G1HVQ2 RCO3 + #G1HVQM MACO3 +
...
.COE
G1HVPa  0.0  !  PF=C2CHO           Note: text following the "!" is ignored.
G1HVPk  0.0  !  PF=MEK-06
G1HVPg  0.0  !  PF=GLY-07R
G1HVPm  0.0  !  PF=MGLY-06
G1HVPb  0.0  !  PF=BACL-07
G1HVPc  0.0  !  PF=MACR-06
G1HVPv  0.0  !  PF=MVK-06
G1HVPn  0.0  !  PF=IC3ONO2
G1HVPh  0.0  !  PF=COOH
.INS DIFF
'G1HV' = ('G1HVPa'*'C2CHO') + ('G1HVPk'*'MEK-06') + ('G1HVPg'*'GLY-07R') + &
   ('G1HVPm'*'MGLY-06') + ('G1HVPb'*'BACL-07') + ('G1HVPc'*'MACR-06') + &
   ('G1HVPv'*'MVK-06') + ('G1HVPn'*'IC3ONO2') + ('G1HVPh'*'COOH')
```

These model-specific simulation programs are usually not invoked directly but are generally run by using the INT, INTRUN, (Section 2.4) or REACTCAL (Section 3.6) programs as discussed in more detail below. As discussed in Section 2.4.2, basic model simulations are conducted using the INT program, which simply calls the model specific program that is indicated in the simulation input or parameter files and does no additional pre- or post-processing. When this program is used, the simulation input files must be created or edited directly by the user. For that reason, most of the discussion and examples in this section are based on the assumption that the INT program is used.

The simulation programs can also be invoked using the INTRUN program if lumping pre-processing of the input files is desired (Section 3.4), or by REACTCAL (Section 3.6) if the purpose is to conduct incremental reactivity simulations. These programs create the simulation program input files as part of their input pre-processing, so the user would edit other types of files when creating simulation inputs when INTRUN or REACTCAL are used. However, the input formats for the simulation program input files are the same regardless of how the simulation program is invoked, and knowledge of the simulation program inputs described in this section is also necessary to prepare at least some of the inputs for INTRUN or REACTCAL.

As discussed in Section 2.4.2, the INT program is invoked by giving the command "INT *calcname* [*modelname*]", where *calcname* is the name of the INT input file and *modname* is the name of the model's .PRM file, if used (i.e., input from *calcname*.INT and *modelname*.PRM). The main model simulation input is in the *calcname*.INT file, whose format and contents are discussed in detail below in Sections 3.3.1. In addition, the INT program and the model simulation program also read the parameter files MODELING.PRM and (given on the command line) *modelname*.PRM for parameters and options that control the programs, as discussed in Section 3.1 and Table 3. Note that if *modelname*.PRM is given on the command line then the options and parameters override those in MODELING.PRM if they conflict. The output files that are created depend on the options and input given in the .INT file, and are discussed below in Section 3.3.6. Options for processing or examining these output files are discussed in Section 3.5.

Options in the parameter files that control the operations of the model simulation programs are given in Table 3 with the notation "INTs" in the "Programs" column. Among the expected input in the parameter file is the "MOD=" input that gives location of the modeling program data (.MOD) file that was created when the model was prepared. This .MOD file is required for the model simulation program to run so it will fail if it cannot be found. Other input in the parameter files are needed depending on the input in the .INT file and are discussed in the context of the discussion of the inputs (or outputs) in the relevant sections below. An additional option in the parameter files that controls the INT program itself (and also INTRUN) is the DFLMOD= parameter, which indicates which model simulation program to use if it is not specified in the .INT input file. The actual model simulation programs ignore this parameter.

3.3.1. Model Simulation Input (.INT) Files

Model simulation input (.INT) files are plain text files consisting of lines with no more than 80 characters. As with preparation program input (.PRP) files these files have three sections, (1) a single comment section that describes the simulation and is copied to the listing output (.OUT) file but is otherwise ignored, (2) parameter input, and (3) simulation control input. As with other types of input files, any lines that start with a "!" are treated as comments and are ignored.

<u>Parameter Input</u>. The parameter input can consist of any number of lines and is terminated by a line consisting of a "." in column 1, which is treated as the first line in the simulation control input, discussed below. As with preparation program input files, any line that starts with a "!" is treated like a

comment. But unlike preparation input files, any other characters in the termination line that starts with a "." are ignored, so generally a line with just a "." is used. The options or parameters that can be input here, and their effects on the program, are summarized in Table 9. The program gives a warning message if the parameter input contains input that is not recognized, but the input is otherwise ignored and execution is continued.

Simulation Control Input. Most of the model simulation input files consist of simulation control input, which gives initial concentration and parameter values and other input that determines the conditions of the simulation, and gives the commands to carry out the simulations using the input provided previously. This input starts with the "." that terminates the parameter input portion discussed above. As with the parameter input, any line that starts with a "!" is ignored. Input and commands that can appear in the simulation control input are summarized in Table 10. Note that in most cases unrecognized input in the simulation control portion will result in the program being terminated, though some errors only cause error messages if the "ERROK" option is specified in the parameter input (see Table 10).

Note that as indicated on Table 10 the simulation control input data fall into three general categories: simulation condition input, commands, and miscellaneous. The only input in the latter category is the "NAME" command, which changes model species and parameter names, affects subsequent but not previous inputs, and is very rarely needed. Simulation condition input consists of specifications of time ranges and initial concentration and parameter values that determine what is being simulated and the parameter values used. Some of these parameters can vary with time during the simulation or temperature, as indicated on the table. Commands cause the simulations to be carried out and in some cases control the names of the files where the results, allow for the possibility of more than one simulation to be carried out during this execution of the program, control initial or default values for subsequent calculations if more than one simulation is conducted, and control whether the results of the calculation, if saved in .CSV or .CDA files (see Section 3.3.6) are appended to the previous file or saved in a new one (in which case the name of the new file must be given). Command lines are executed in the order they are input, so their order and location in the file relative to the other input is important. However, the order generally does not matter for simulation condition input commands between the start of the simulation control input and the first command input line, or between different command input lines for multiple calculation simulations, except that if a parameter or initial species concentration is input more than once, the previous input values are over-written. For simple calculations, the only command input usually needed is the ".INT" command, which causes the calculation to be carried out and is the last input line in the file.

At the start of the input the default values for the parameters and the constant species concentrations are those specified in the model preparation input files, and are zero for the non-constant model species in the model. The simulation condition input then changes these values as desired, and these updated values are used in the simulation resulting from executing the first ".INT" or .INT *file* command. If the file has multiple simulations (i.e., more than one .INT or .INT *file* command), then the initial concentrations or parameter values for the next simulation are reset to the defaults after the execution of an ".INT" command, and any non-default values would have to be input by subsequent simulation condition input lines. The only exception to this are time-varying parameter values specified in the .VARYCO inputs or the temperature dependent parameter values that may be input, which are the same for subsequent simulations unless additional input is given to change them.

The defaults for multiple calculation simulations are those specified in the model preparation inputs unless the .SETDEF command is executed, in which case the defaults for parameters and initial concentrations for subsequent simulations are those at the time the .SETDEF command is executed. The .SETDEF command does not cause the start of a simulation, and additional simulation input can appear after a .SETDEF command to control the conditions for the next .INT command. But this additional input

Table 9. List of options and parameters that can be specified in the parameter input portion of the model simulation input (.INT) files.

Code [a]	Description
MODEL=*model*	Name of the model simulation program. This is used by INT to determine which simulation program to run. The simulation program uses this to compare the name with its own name and gives a warning message if they are different, but otherwise it ignores this and does not treat it as an error. This is optional if the program is invoked directly or run using INT, INTRUN, or REACTCAL and the program is named in the parameter file.
	<u>Concentration-time output control</u>. [By default, concentration time data are only output in the listing output (.OUT) file.]
SAVE	Saves the concentration-time results of the calculations in file *calcname*.CDA, where *calcname* is the name of the .INT input file. The location of the output file depends on the other inputs and parameter options as discussed in Section 3.3.6.
SAVE CSV	Saves the concentration-time results in a .CSV file in the local folder. By default the file is named *calcname*.CSV, where *calcname* is the name of the .INT input file. However, if a "SAVE *file*" option input appears <u>before</u> this line, the file is named *file*.CSV.
SAVE *file*	Saves the concentration-time results of the calculation in file *file*.CDA or *file*.CSV in the local folder. If "SAVE *file*" is given by itself, the output is *file*.CDA. If the option input "SAVE CSV" is input <u>after</u> the "SAVE *file*" line, the output file is *file*.CSV.
SAVE RATE SAVE INTRATE	Include the rates (for "SAVE RATE") or integrated rates ("SAVE INTRATE") for all the reactions in the output files. Both of these can be used if both rates and integrated rates are to be saved. Rates are given the names "R#*label*" and "IR#*label*", respectively, where *label* is the reaction label.
NSAVSS *n*	By default, the concentrations of only the first 5 steady-state species are included in the .CDA or .CSV output files. This can be used to cause the first *n* steady-state species to be output, where *n* is an integer greater than zero. If it is desired to save the concentrations of all steady-state species, then set *n* at a number larger than the total number of model species in the model.
INTRATE	Causes integrated reaction rates to be output in the .CDA or .CSV files. Similar to "SAVE INTRATE" but doesn't cause the files to be created, so has no effect unless another "SAVE" option is already included.
MODELSET=*name*	Causes the .CDA files to be stored in the folder indicated by *name*, rather than the name of the parameter (.PRM) file specified on the command line or the default name "CALCS". Has no effect on .CSV output. See Section 3.3.6 for a discussion of where the .CDA output files are located.
	<u>Listing output (.OUT) file control</u>. [By default, a listing output (.OUT) file is created listing the species and reactions in the model, echoing the simulation inputs, and listing the concentration time results in a format suitable for a 132-character line printer. Error messages, if applicable, are also given in this file.]

Table 9 (continued)

Code [a]	Description
NOOUT	Suppresses creation of the simulation output (.OUT) file unless errors occur. This is included in chamber model and reactivity scenario input files included in this distribution because they have "SAVE" options to save the results of the simulations of interest, and OUT files are not normally used. All the other options related to .OUT files are ignored if this is specified.
NPMODL	Suppress listing of the model's species and reactions in the output file.
WRITR	Includes rates of the reactions in the .OUT file.
WRITDY	Includes rates of change of active species in the .OUT file.
NOPRINTT	Do not include the concentrations as a function of time in the .OUT file. WRITR and WRITDY is ignored if this option is used.
DEBUG	Show each time step in the .OUT file. Generally it will show many time steps are needed at the start but the size of the steps increase as the simulation proceeds.
	<u>Solver Control</u>. These parameters control the operations of the solver used to calculate the rates of change of the model species.
NOCHEM	Run the simulation with the chemistry "turned off", i.e., no concentration changes due to chemical reactions. This is useful for verifying correct input that controls emissions and dilution rates and entrainment of species from aloft, as discussed in Section 3.2.10.
EPS *value*	Sets the error control parameter used by the LSODE implementation (Hindmarch, 1872) of the Gear solver (Gear, 1971) to a value of *value*, which should be a real number less than 1. A lower value gives a more accurate simulation but requires smaller time steps and therefore more computer time to run the simulation. The default value is 10^{-4}, which was found to be the largest value that did not affect the results to a significant extent in the test simulations. However, the distributed input file controlling this parameter in the incremental reactivity simulations uses a value of 10^{-6} to minimize numerical errors when determining effects of additions of small amounts of test compounds. This is a unitless quantity.
CMAX *value* CMIN *value*	Sets the approximate magnitude for the maximum ("CMAX") and minimum ("CMIN") active species concentrations used for error control purposes by the LSODE implementation (Hindmarsh, 1972) of the gear solver (Gear, 1971) to a value of *value*, which should be a real, positive number. The default values of CMAX and CMIN are 1 and 0.01, respectively, which should be appropriate for atmospheric simulations using ppm units for the model species. If other units are used this should be set to the approximate maximum magnitude of the concentrations of active species, and the simulations tested using problems with known analytical solutions using various values of EPS and CMAX.
MAXSTP *n*	Sets the maximum number of time steps to *n*, where *n* is a positive integer. The default value is 500. The simulation is aborted when this maximum is reached. Normally 500 steps is sufficient and if there is a numerical problem (such as would occur if a temperature of zero were input) then increasing the maximum number of time steps beyond this only increases the amount of computer time that is wasted before the calculation aborts.

Table 9 (continued)

Code [a]	Description
	Debugging output control. These are not normally used but are sometimes useful for debugging or diagnostic purposes.
WRITCO	Outputs all the variable coefficients values at the end of the simulation in the file VARYCO.TMP. Performs the simulation as usual otherwise.
WRITK	Outputs a file WRITK.CSV with the rate constants for the default temperature of the model or the initial temperature in the input file. Does not output photolysis rates, and does not perform the model simulation.
ERROK	Allows the simulation to proceed even if some errors occur. The results are generally not valid but the output may still be useful for debugging.
	Environmental Conditions Input.
SUN	Calculate solar photolysis rates using the simulation program's built-in actinic flux data, which is based on the actinic fluxes of Peterson (1976) using the "best estimate" albedos. These are calculated given the absorption cross sections in the model and the solar zenith angle, where the zenith angle is calculated from the simulation time, solar time offset, and the solar latitude and declination input as discussed in Section 3.3.5.
P(Z)=*file*	Calculate solar photolysis rates using parameters calculated to fit the model's photolysis rates as a function of zenith angle in the file *file*.FZS. See Section Section 3.3.5 for a discussion of how to derive these .FZS files for a particular model and set of solar actinic fluxes. Zenith angles are calculated in the same way as for the "SUN" option, as discussed in Section 3.3.5.
LAT DEC	Default latitude and solar declination for solar photolysis rate calculation. Same as .LAT and .DEC input in the simulation control portion of the .INT file. See Section 3.3.5 for a discussion of how solar photolysis rates are calculated.
CZMIN *value*	The minimum cosine of the solar zenith angle for calculation of photolysis rates is set to *value*, which is a real number between zero and 1. This is used to calculate photolysis rates at night. The default value is 10^{-6}. Using a value of zero sometimes cause the simulations to fail so the default is a small but nonzero value.
TEMP *temperature*	Default simulation temperature. This can also be input in the simulation control portion of the input file using .TEMP or the parameter TEMPR.

[a] Case sensitive unless indicated. Text in *italics* indicate variable input as described in the comments.

Table 10. Summary of input that can be given in the simulation control input section of model simulation input (.INT) files.

Code [a]	Description
	Simulation Condition Input: Simulation Time.
.T0 *time*	Set the starting time of the simulation to *time*. This is zero by default. This is used in subsequent calculations in multi-calculation simulations unless the .CONT command is executed, in which case the simulation start time in the subsequent calculation is the simulation end time of the previous calculation. It is also not applicable if a .GET command is input for this calculation because in that case the default starting time is the time given on that command. However, the .T0 command can be used to change the start time if it comes after the .GET command. The default time units is minutes, or minutes after midnight if solar photolysis rates, which depend on the time of day, are calculated.
.TEND *time*	Set the ending time of the simulation to *time*. This is 360 minutes by default. This used in subsequent calculations in a multi-calculation simulation unless the .CONT command is executed, in which case a new .TEND must be specified for the subsequent calculation.
.TEND+ *time*	Increment the ending time of the simulation by *time*. Mainly useful in multi-part simulations using the .CONT command.
.DPRN *time*	Output time interval of the simulation. This is 10 minutes by default. This is applicable for subsequent calculations unless different values are input for that calculation.
	Simulation Condition Input: Species Concentrations and Parameters
name value	Gives the model species or variable parameter with name *name* the initial concentration or value of *value*, where *value* is a real number. The list of model species names is searched first, followed by the list of variable coefficient names, so if a variable coefficient has the same name as a model species then its value cannot be changed. These remain in effect until an .INT or .INTAPND command is processed, and become the default values for multiple simulations if a .SETDEF command is processed. (Note: If a parameter and a model species has the same name, then the input is assumed to refer to the model species.)
+*name value*	Increments the current model species concentration by the amount *value*. This is normally useful to simulate reactant injections in multi-part simulations using the .CONT command, but can be used in other contexts if desired. Not applicable for parameters.
#*name value*	Gives the variable parameter named *name* the value of *value*. Useful if a parameter has the same name as a model species. Note that in those cases the model species is referenced if there is no "#" prefix.
.FAC *factor* .NOFAC	Multiplies all initial concentration input after this point by the factor *factor*, which is a real number. ".NOFAC" input cancels this, i.e., sets *factor* to 1 for all subsequent concentration input. This does not affect parameter input.

Table 10 (continued)

Code [a]	Description
.VARYCO (followed by time-varying coefficient input)	Read time-varying coefficient input. The subsequent input lines consist of sets of inputs for particular coefficients giving the coefficient name, values for given times, and how the values are to be interpolated. The input for each time varying coefficient starts with either "L#*name*" or "C#*name*", where *name* is the name of a variable coefficient in the model, and "L#" and "C# are codes indicating how coefficient values are to be used at times that are not specified. Note that if *name* is "TEMPR" then this can be used to specify how temperature varies with time. Following this are lines with a time, one or more spaces, and the coefficient value at that time. Both of these must be decodable as real numbers. "L#*name*" or "C#*name*" Input for another coefficient ends the input for this one, and a #END input terminates all .VARYCO input. Input for all time-varying coefficients must be given in a single set of .VARYCO input, and if more than one .VARYCO input set is in an .INT file then none of the input from the previous one is used when the next .INT or .INTCONT command is processed. If "L#" is used the values for times between those given are determined by linear interpolation of values at times given, while if "C#" is used the values used are for those of the latest time for which a value was given, i.e., they change in a stepwise manner. In both cases values for times before the first or after the last input time are the same as the first or last, respectively. .VARYCO input given for times outside the time range of the simulation are ignored.
.NOVCO	Cancels the effect of the previous .VARYCO command. The affected coefficient values revert to their default values and can be changed by "*name value*" input as indicated above. This is only useful for multi-simulation input files.
.GET *file time* .GET * *time*	Get the model species concentrations from the previously-created .CDA file *file*.CDA for the time *time*. The .CDA file must exist, and could have been created in this or in a previous calculation, but if it is a previous calculation it must have used the same parameter file, or both should use none. If * is used in place of *file* then it gets the data from the previous calculation. Data for time *time* must also be in the file; no interpolation or extrapolation is done. The initial simulation time is changed to *time*, but can be changed to zero or something else by using a subsequent ".T0 *time*" input line.
	<u>Simulation Condition Input: Special Parameters</u>
TEMPR *temperature* .TEMP *temperature*	Simulation temperature is set as *temperature*, in degrees K. Note that "TEMPR" is defined as a variable parameter for all models, so this is just an instance of variable value parameter value input.
.DIL *dilution*	Sets the dilution rate to *dilution*, which is zero by default. The dilution rates would normally be in units of $\min^{-1}$.
	<u>Simulation Condition Input: Thermal or Constant Kinetic Parameters</u>
K(*rxnlbl*) *value*	Changes the rate constant for the reaction labeled *rxnlbl* to the value *value*, which must be a real number. The kinetic parameters for the reaction must be constant, extended Arrhenius, falloff, or with special T and P dependence as indicated on Table 6, and not a photolysis reaction or dependent on another reaction or coefficient.

Table 10 (continued)

Code [a]	Description
KP*n*(*rxnlbl*) *value*	Changes kinetic parameter *n* for the reaction labeled *rxnlbl* to the value *value*, which must be a real number. The kinetic parameters for the reaction must be extended Arrhenius, falloff, or with special T and P dependence as indicated on Table 6, and not constant or a photolysis reaction or dependent on another reaction or coefficient. This cannot be used in conjunction with a "K(*rxnlbl*)" input for the same reaction. "*n*" is an integer with a value between 1 and 9 and indicates the type of kinetic parameter, as indicated by the order the parameters are listed in the second column of Table 6. For example, KP1, KP2 and KP3 for an extended Arrhenius reaction is A, Ea, and B, KP7 and KP8 for a falloff reaction is F and N, etc.
	Simulation Condition Input: Photolysis Parameters - Constant Light Spectrum
.SD-*sdrfile*	Read the spectral distribution from the file *sdrfile*.SDR for use in calculation of photolysis rates. The .SDR file must be located where indicated by the SDR= input in the parameter file if it is not in the local folder. The format of these files is discussed in Section 3.3.4. If the .SDR file has only a relative spectral distribution, then a P(*rxnlbl*) input line, discussed below, must be given to give a photolysis rate to normalize the total light intensity to the appropriate level.
.SD- (followed by spectral distribution input)	Reads the spectral distribution data from the subsequent lines. The format for this input is the same as discussed for .SDR files is discussed in Section 3.3.4. It is terminated by a blank line or with "#END" as the input line.
P(*rxnlbl*) *value*	Scales the spectral distribution input such that the photolysis rate as calculated for the photolysis reaction with label *rxnlbl* is equal to *value*. A spectral distribution must be input using the .SD- input as indicated above. Note that the reaction with label *rxnlbl* in the model preparation input must have its kinetic parameters specified as PF=*phfname*, where *phfname* is the name of some photolysis data (.PHF) file. If the reaction has non-unit quantum yields, they must be included in the photolysis data file, and not be given as QY= input in the kinetic parameter input for the reaction. For example, if the model has the reaction "R1) PF=NO2 ;NO2 + HV = NO + O", and the experiment to be simulated has a relative spectrum distribution given in the file LIGHTS.SDR and has a measured NO$_2$ photolysis rate of 0.3 min^{-1}, then the simulation input file should contain the inputs ".SD-LIGHTS" and "P(R1) 0.3" to calculate the appropriate photolysis rates. The order of these inputs does not matter, as long as they are before an .INT, .INTAPND, or .SETDEF command.
	Simulation Condition Input: Photolysis Parameters - Time-Varying Solar Light
.LAT *latitude* .DEC *declination*	Sets the latitude and declination for calculating the solar zenith angle as a function of time, where *latitude* and *declination* are in degrees. Only applicable if the SUN option is used or P(Z) input is specified in the options input. Same as the LAT and DEC options in the options input.

Table 10 (continued)

Code [a]	Description
.TSTART	Difference between the time of the simulation and the solar time used when calculating the solar zenith angle for the given latitude and declination, i.e., the solar time when time=0. Expected to be in minutes, with solar time being minutes after midnight. Depends on the time zone, longitude, and whether daylight time is being used.
	Simulation Commands.
.INT	Conduct the simulation using the conditions previously input. This should be the last input line unless multiple simulations are desired. The default parameters and initial conditions are used if they are not specified in the previous input. At least one .INT command must be given in the .INT file or there will be no simulation. This is usually the last line in the file unless multiple simulations are to be conducted.
.INT *file*	Same as .INT except that the results go into the .CSV or .CDA file named *file* rather than with the same name as the simulation input (.INT) file, if the SAVE or SAVE CSV option is in effect. This is required in multi-calculation simulations if the results are to go into separate files, but is allowed in single simulation inputs as well. The files go in the subfolder indicated by the parameter file name or in CALCS folder if no parameter file is used.
.END	Terminate the input with no additional processing. Any subsequent input is ignored. This is not necessary and has the same effect as an end-of-file.
	The following commands are only applicable for simulations with multiple calculations.
.CONT	This simulation is a continuation of the previous one. Starting species concentrations and the starting simulation time are the same as those at the end of the previous simulations, and the default initial concentrations are ignored. But parameter values revert to their default values (or values at the time the last .SETDEF command was input). This command is only allowed immediately following an .INT or .INTAPND command.
.INTAPND	Same as the .INT command except that the results are appended to the output file created in the previous simulation if the SAVE or SAVE CSV option is in effect.
.INTAPND *file*	Same as .INTAPND except that the results are appended to *file*.CDA rather than the file named by the name in the simulation input file. The file named *file*.CDA must exist, and could have been one created in a previous simulation, or one created earlier in this one.
.SETDEF	Sets the current parameter values and initial concentrations for subsequent calculations to the values that have been input up to this point.

[a] Case sensitive unless indicated. Text in *italics* indicate variable input as described in the comments.

only affects the calculation started by the next .INT command, and is not the default for subsequent calculations.

If the .CONT command is executed immediately after an .INT command, the final model species concentrations become initial model species concentrations for the subsequent calculations, the initial simulation time is set at the final time, and the final simulation time for the subsequent calculation must be given using a ".TEND" or ".TEND+" command. However, the parameter values refer to the model or .SETDEF defaults. No input is allowed between the .CONT command and the .INT command before it. This is useful for simulations of situations where some parameter, such as light intensity, is changed in the middle of a simulation, or if the effect of adding a reactant in the middle of an experiment is being simulated. Note that if you are simulating the effect of adding more of a reactant that is already present, you would want to use the "+*name value*" input, where *name* is the name of the species added, and *value* is the amount to that its concentration is to be increased.

3.3.2. Specification of Reactant Inputs

Initial Concentrations. Initial concentrations of reactants are given as part of the simulation input using one of the methods listed below. Default initial concentrations are zero except for constant species, where defaults can be specified during model preparation.

- The initial concentration of a particular species can be input using an input line giving the name of the species and its concentration in the units of the simulation, which is generally ppm. The input concentrations can be modified using previous .FAC input as indicated on Table 10.
- Initial concentrations for all species can also be specified using a .GET or .CONT command, where results of previous simulations are used. The .GET command is used to reference results of a previous model simulation run, while the .CONT command is used for multi-part simulations as discussed in Table 10.
- Initial concentrations of species assigned a group using the .GROUP command with the AIRSHED option can be input if the input or default values of the parameters TOT.*group* and INI.*group* are nonzero. The R*n*#*species* parameter, which gives the moles of *species* per mole of the group, must also be specified or nonzero by default. The units for amounts of compounds in this group are input depend on the value of the EMIT.FAC parameter. This is discussed in Section 3.2.10.

Emissions without the AIRSHED option. It is necessary to make appropriate provisions for emissions in the model preparation input in order to carry out simulations where species are injected into the reacting parcel during the simulations. If the AIRSHED option is not specified during model preparation, then the only way that emissions can be input is to add a pseudo zero-order reaction in the .RXN input at model preparation time as discussed in Section 3.2.9. This pseudo reaction should normally include a constant species forming the emitted species with a unit rate constant (or a rate constant that converts input units), with the concentration of the constant "reactant" species being the emissions rate, which can be specified in the simulation input. The default emissions rates can also be specified in the model preparation input.

Emissions with the AIRSHED option. The inclusion of the AIRSHED option in the prepared model allows emissions to be specified in area flux units, which is considered more appropriate for ambient simulations than concentration/time units. The flux units are generally moles or millimoles meter^{-2} per unit time, depending on the value of the EMIT.FAC parameter, as discussed in Section 3.2.10. A mixing height, given by a parameter called HEIGHT, must also be specified in order to convert flux to concentration units, with the default being meters. Two options are available for emissions input for such models.

- Emissions rates of species that are named in .EMI input in the model preparation input can be specified in the model simulation input by giving nonzero values to "E#*name*" parameters, where *name* is the name of the species. The units of these parameters are flux per unit time, depending on the value of the EMIT.FAC parameter, with the default units being millimoles meter^{-2} min^{-1}.
- Emissions rates of species assigned a group using the .GROUP command with the AIRSHED option can be input if the input or default values of the parameters TOT.*group* is nonzero, and INI.*group* is less than 1. EMR#*group* parameter, which is used to allow for time-varying emissions and must be normalized to be consistent with INI.*group*, must also be defined. For example, if the default value of EMIT.FAC is used and the units of TOT.*group* are millimoles meter^{-2} min^{-1}, then the appropriate

See Section 3.2.10 for further discussion of the inputs used to specify group emissions initial concentrations.

3.3.3. Dilution, Deposition, and Entrainment

Dilution of all reactants. As indicated in Table 10, the .DIL input at simulation time is used to define a dilution rate for all active and buildup-only species in the simulation. It also can be specified by giving a value to the "DILUTION" parameter (Table 7) at the simulation input, or a default value in the preparation input. It can be used both with or without the AIRSHED option. If the AIRSHED option is specified, this is in addition to any dilution that may occur if the mixing height (the HEIGHT parameter) increases.

Deposition without the AIRSHED option. The loss of reactants to surfaces can be represented in models not using the AIRSHED option by adding "reactions" in the model preparation input that represent first order loss processes, e.g.,

label) COEF=*Dep1* ;*species1 = [counter species]*
or *label*) 1.0 ;*species1*+ #*Dep1* = *[counter species]*

where *Dep1* is the name of the coefficient that contains the pseudo-unimolecular loss rate coefficient, and *species1* is the model species. Its default value is zero unless it is given by .COE input during model preparation. It is not necessary to include products in these loss "reactions", but a counter species could be used if it is desired to track the total amount lost by this process. Note that loss rates are given as a unimolecular rate constant, so in general it will depend on the amount of surface present. The AIRSHED option must be used if it is desired to specify deposition rates in terms of area flux.

Deposition with the AIRSHED option. Use of the AIRSHED option allows the loss of reactions to surfaces to be represented as a surface flux that can be specified for each compound, based on the conceptual model of the reacting parcel being defined by fixed area surfaces on the ground and at the mixing height, with the height being specified as an input parameter (Section 3.2.10). If this option is specified in the parameter section of the PRP file, then .DEP input (Table 5) in the model input section of the file can list the species where deposition can occur. This cannot be used for steady state species. The .DEP input creates parameters called "D#*name*" for species listed in this input, where *name* is the species name (or the first 5 characters thereof) that gives the deposition velocity (units of distance^{-1} time^{-1}) for the compound onto the ground surface. A parameter called "DEP.FAC" is also created, which is available for units conversions or to turn deposition on or off for all such species during simulation input. The default value is 1, meaning that a factor is not needed. The rate of change of the compound due to deposition is then given by

$$d[\textit{species}]/dt^{\text{Depostion}} = \text{-DEP.FAC} * \text{D\#}\textit{species} \text{ / HEIGHT} * [\textit{species}]$$

where *species* is the name of a model species and HEIGHT is the parameter giving the mixing height. Note that the units of DEP.FAC * D#*species* are distance per unit time (deposition velocities), and must be consistent with the units used for HEIGHT and time in the simulation, which are usually meters and minutes. An appropriate value of DEP.FAC can be used if different units are desired for the deposition velocities.

Entrainment and Dilution with the AIRSHED option.

As discussed in Section 3.2.10, the conceptual model implemented by the AIRSHED option allows for entrainment of air from an unreacting layer aloft as the inversion height increases. The aloft layer does not contain any reactants by default, but concentrations of reactants in the aloft layer can be specified at simulation time (or as defaults specified at model preparation time) by assigning concentrations to "A#*species* parameters, where *species* is the name of an active model species. The rate of change of the species due to increases in the mixing height is given by

$$d[\textit{species}]/dt^{\text{Entrainment}} = (\text{A\#}\textit{species} - [\textit{species}]) * \max(0, \Delta\text{H}) \text{ / HEIGHT}$$

where [*species*] and A#*species* are the concentration of the species in the reacting and aloft layers, respectively, HEIGHT is the mixing height, and ΔH is the instantaneous change in the inversion height. Note that if A#*species* is zero, then the increase in the inversion height causes the species to be diluted with a first order rate coefficient of ΔH/HEIGHT. No entrainment or dilution occurs if HEIGHT is not increasing, i.e., if ΔH is not positive.

If entrainment is to occur in the model simulation, then a time-varying inversion height needs to be specified using the .VARYCO simulation input using the "L#" option, where the values at intermediate times are determined by interpolation (see Table 10 and Section 3.3.1). When processing time-varying parameter input with this option, the simulation program computes and saves both the slope and the intercept at each time segment for the purpose of deriving interpolated values. In the case of the HEIGHT input, the slopes of these segments are the instantaneous values of ΔH. Note that the ΔH used at the times when values are input is the value for times between that input time and the next.

3.3.4. Photolysis Rates with Constant Light Sources and Spectral Distribution Data

General Considerations. The recommended procedure for modeling photolysis reactions is to calculate the photolysis rates from the absorption cross sections and quantum yields as a function of wavelength for the various photolysis reactions and for the and the intensity as a function of wavelength for the light source, which can be sunlight or an artificial light source such as blacklights. The specification of photolysis reactions during model preparation was discussed in Section 3.2.7 and shown on Table 6. The calculation of photolysis rates, and the specification of light intensity and spectra and calculation of photolysis rates for light sources with constant spectra is discussed in this section. The calculation of photolysis rates for time-varying solar light, and the specification of solar actinic flux data, is discussed in the next section.

In the context of this discussion, the term "photolysis rate" refers to the rate constant for the reaction of a model species with light, which is treated as a first order reaction of the model species by the software. In theory, photolysis rates are calculated from

$$k^{\text{phot, light}} = F^{\text{light}} \cdot \int_{\lambda} J^{\text{light}}(\lambda) \cdot \text{Abs}^{\text{phot}}(\lambda) \cdot \text{QY}^{\text{phot}}(\lambda)\, d\lambda \qquad (1)$$

where $k^{phot, light}$ is the first order rate for this photolysis reaction and light source, $J^{light}(\lambda)$ is the intensity of the light source at wavelength λ, $Abs^{phot}(\lambda)$ and $QY^{phot}(\lambda)$ are the absorption cross section and the quantum yield for that reaction at that wavelength, and F is a scaling factor that depends on the units of k^{phot} and J^{light}, or depends on the overall light intensity if the values of J^{light} refers to a relative spectral distribution, as is generally the case for artificial light sources. The software assumes the units of λ are microns (micrometers), the units of absorption cross sections are cm^2 $molec^{-1}$, base e, quantum yields are unitless, and the units of $J^{light}(\lambda)$ are photons cm^{-2} min^{-1} $micron^{-1}$, or something proportional to that if F^{light} is chosen to give $k^{phot, light}$ in min^{-1} or to or give a desired rate constant for a selected photolysis reaction.

For applications such as modeling environmental chamber experiments, the set of $J^{light}(\lambda)$ values for the light source is usually specified in relative terms depending on the type of light, and the F^{light} value is derived to agree with a photolysis rate derived from actinometry experiments, $k^{std, light}$

$$F^{light} = k^{std, light} \Big/ \int_{\lambda} J^{light}(\lambda) \cdot Abs^{std}(\lambda) \cdot QY^{std}(\lambda)\, d\lambda$$

where "std" refers to the photolysis reaction whose rate coefficient is measured in the actinometry experiments. Because artificial lights can have any intensity, generally only relative $J^{light}(\lambda)$ values are available except for solar photolysis scenarios.

In practice, J^{light}, Abs^{phot} and QY^{phot} are only given at discrete wavelengths, and the wavelengths for which J^{light} are given may be different than those for Abs^{phot} and QY^{phot}, so Equation (1), or the version where "$\int$" is replaced by "Σ" cannot be used. The software allows for the absorption cross section and quantum yield data to be given for any set of wavelengths, which can differ for each photolysis reaction and for the light spectra used, with no requirements for the wavelength intervals other than they be input in ascending order and the total number of wavelengths not exceed the program dimensions, which are relatively large and can be increased if needed. The exact method used depends on whether the solar light intensities are used for ambient or sunlight simulations, or whether a constant light spectrum is given for artificial light irradiations. The former is discussed in the following section and the latter is discussed below.

Calculation of Photolysis Rates with a Constant Spectrum. Calculations with a constant spectrum consist of simulations where the model simulation input file does not use the "SUN" or "P(Z)" options (see Table 9) and the constant light source spectrum is taken from a spectral distribution file *sdrfile*.SDR given by the "SD-" input (see Table 10) in the simulation input file. This is appropriate for simulations using artificial light sources. In these cases, the software assumes that the actual continuous spectrum of the light is approximated by line segments connecting the points defined by the wavelengths given in the .PRP or .SDR file, and calculates the photolysis rates accordingly. This is sometimes referred to as the "point" method for specifying light spectra, to distinguish it from methods where spectra are given in terms of total intensities for various wavelength intervals.

The photolysis rates are calculated by splitting the wavelength region covered by both the light intensity and the absorption cross section into wavelength segments where values are given for at least one of the two sets, as shown for example on Figure 3. Linear interpolation is used to obtain the values for wavelengths where the action spectrum or light spectrum are not given. For each segment, the photolysis rate is calculated using Equation (1), as if the action or light spectra were given by trapezoids defined by the points at the end of these segments. For example, for the second segment on Figure 3, the contribution to the total photolysis rate for the segment is given by

$$k^{phot}\ \text{Contribution} = IJ \cdot IAQ \cdot (\lambda2-\lambda1) + \{(IJ \cdot SAQ)+(IAQ \cdot SJ)\} \cdot (\lambda2^2-\lambda1^2)/2 + SJ \cdot SAQ \cdot (\lambda2^3-\lambda1^3)/3$$

where

(SAQ, IAQ) = slope and intercept of line defined by (λ1, AQ1), (λ2, AQ2)
(SJ, IJ) = slope and intercept of line defined by (λ1, J1), (λ2, J2)

and λ1 and λ2 are the wavelengths defining the segment, AQ1 and J1 are the values of $Abs^{phot} \cdot QY^{phot}$ and J^{light} at λ1, and similarly for AQ2 and J2, as shown on the figure. Note that in this case AQ1 and AQ1 are as input, while J1 and J2 are interpolated, while for other segments the ones that are input and interpolated may differ. The total photolysis rate is then the sum of the contributions calculated for each segment, multiplied by the scaling factor F, which is 1 by default but can also be derived as discussed below.

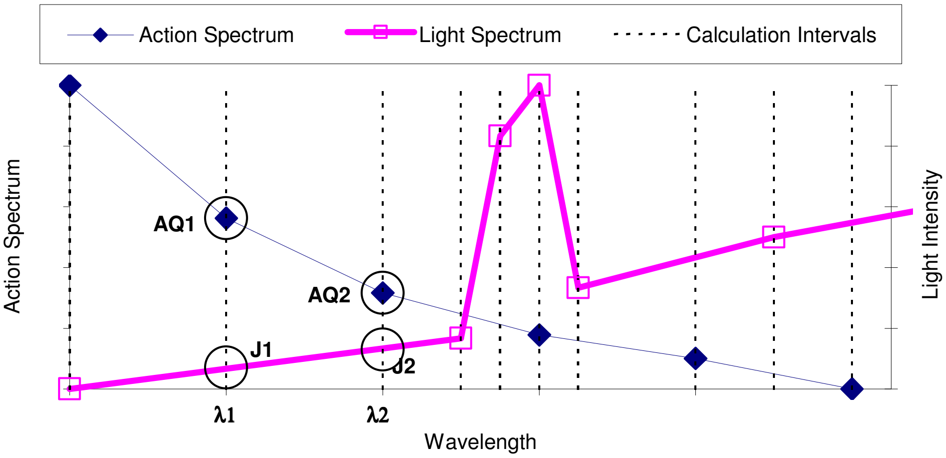


Figure 3. Calculation of photolysis rates in various wavelength inputs from input absorption cross sections x quantum yields (AQ) and light intensity (J) as a function of wavelength (λ).

In most cases the spectral distributions in the .SDR files are relative spectral distributions and are not sufficient by themselves to calculate absolute photolysis rates. Use of the default scaling factor, F, of 1 will usually give incorrect results. This is the case for all the .SDR files in the distributed for simulating the environmental chamber experiments. In these cases, in addition to including an "SD-*sdrfile*" input in the simulation input file, it is also necessary to input the rate constant for a photolysis reaction using the "P(*photlabel*) *value*" input in the simulation input file, where *photlabel* is some photolysis reaction in the model whose quantum yields are included in the .PHF file (not the kinetic parameter portion of the .RXN input). In the case of the indoor environmental chamber experiments in the distributed files, all experiments have an assigned NO_2 photolysis rate derived from the results of NO_2 actinometry experiments, so the light intensities are scaled to give the assigned NO_2 photolysis rates. The simulation input files implement this using a "P(1) *value*", where "1" is assumed to be the label for the reaction representing the NO_2 photolysis to NO + O^3P. (Therefore, models using the distributed input files for these experiments must have an NO_2 photolysis reaction, and must use "1" as its reaction label, or the input files would have to be edited to give the label used for this reaction.) The simulation program then uses the input photolysis rate to calculate the scaling factor, F, to use when calculating that and all the other photolysis reaction rates. Note that this scaling factor is not preserved in multi-calculation simulations, unless the .SETDEF command is executed.

Format of Spectral Distribution (.SDR) Files. Spectral distribution (.SDR) files are plain text files used by the model simulation program to input spectral distributions (usually relative) light sources with a constant spectrum. The first line is a label that is used to describe the spectrum but otherwise is ignored by the programs. The subsequent lines are either comments, indicated by a "!" in the first column, or two floating point numbers, separated by a tab or one or more spaces. The first number is the wavelength in microns, and the second is light intensity in units of photons cm^{-2} min^{-1} $micron^{-1}$, or something proportional to that if it is a relative spectral distribution, which it usually is. The wavelengths must be given in ascending order. Note that the spectral distribution data are not photon fluxes for the wavelength intervals given, as is generally the case for solar spectral intensity input as discussed in the next section. Instead, the program assumes that the actual spectrum is approximated by line segments between the points given. The spectral input is terminated by a blank line or the end of the file.

Spectral distribution data can also be given directly in the .INT file (or a file referenced by it for indirect input) following the "SD-" line, without the name of a .SDR file (see Table 6). The format of the data following the "SD-" line is the same as discussed above for .SDR files, except that a blank line must be used to terminate the spectral distribution input

3.3.5. Photolysis Rates with Solar Light and Actinic Flux Data

The general principals for calculation of photolysis rates with time-varying solar light is the same as discussed in the "General Considerations" portion of the previous section. The difference is that the light spectra and intensity, which depend on the simulation time and other parameters, are input differently, and a somewhat different algorithm is used to calculate the time-varying photolysis rates. The specific procedures and data used for this purpose are discussed in this section.

Calculations with solar or solar-like spectra consist of simulations where the model simulation input either has the "SUN" or "P(Z)" options specified in the options section, and do not reference .SDR files. These calculations require that the absolute solar light spectra be input as a function of zenith angle, if the "P(Z)" option is used, or, if the "SUN" option is used, to use absolute solar light spectra as a function of zenith that are built into the model simulation programs. For the "P(Z)" option, the programs JZTOKZ and KZSFIT must be run prior to the model simulation in order to get the parameters needed to calculate the photolysis rates for the models as function of solar zenith angle, while the simulation program does all the processing necessary to calculate photolysis rates if the "SUN" option is used. Use of the JZTOKZ and KZSFIT programs are discussed above in Sections 2.4.1 and 2.4.3, and the formats of the files they use are discussed later in this section. The methods these programs use to calculate photolysis rates are discussed below.

Calculation of the Solar Zenith Angle. The solar zenith angle (Z) is the angle of the sun in the sky, where a zenith angle of zero is direct overhead sun, when the solar light intensity is the greatest. The programs do not actually use the zenith angle itself, but use CosZ, the cosine of the zenith angle when the sun is above the horizon, which is 1 for direct overhead sun, and 0 for the sun at or below the horizon. It is calculated as follows

$$\begin{aligned} \mathrm{CosZ} = \mathrm{Max}\{\mathrm{CZMIN},\ &\mathrm{Sin}(\mathrm{LAT}\cdot\pi/180)\cdot\mathrm{Sin}(\mathrm{DEC}\cdot\pi/180) \\ &- \mathrm{Cos}(2\cdot\pi\cdot T^{solar}/1440)\cdot\mathrm{Cos}(\mathrm{LAT}\cdot\pi/180)\cdot\mathrm{Cos}(\mathrm{DEC}\cdot\pi/180)\ \} \end{aligned} \tag{2}$$

where

$$T^{solar} = \mathrm{Min}\ (\mathrm{TIME+TSTART},\ 1440\text{-}\{\mathrm{TIME+TSTART}\})$$

and TIME is the simulation time in minutes, TIME+TSART and T^{solar} are solar times in minutes after midnight, and LAT, DEC, and CZMIN are simulation parameters referring to the latitude, solar declination, and "night" photolysis rates as discussed in Table 9. Note that the Sin and Cos functions in Equation (2) operate on angles in radians.

Calculation of Actinic Fluxes with the "SUN" Option. The model simulation programs include built-in actinic fluxes based on those provided by Petersion (1976) for his "best estimate" surface albedos. These are given in the file PETERSON.JZS included with the distributed files the REACT/INPFILES folder (see Table 1) for zenith angles of 0, 10, …, 70, 78, and 86, and the wavelengths intervals given in Table 11. The data give the photons/cm^2 for the interval. For zenith angles between 0 and 78 degrees, the actinic fluxes in units of photons/cm^2 in the interval λ_i and λ_{i+1} are fit to the functions

$J_\lambda(Z) = A_\lambda \cdot \exp\{-B_\lambda \cdot \mathrm{Sqrt}([1+E_\lambda]/[\mathrm{Cos}(Z)^2+E_\lambda])\}$(3) where Z is the solar zenith angle and A_i, B_i, and E_i are the parameters corresponding to λ_i on Table 11. For zenith angles between 78 and 90 degrees, the actinic fluxes are calculated by

$$J_\lambda(Z) = J_\lambda(78) \cdot \{\max[\mathrm{CZMIN}, \mathrm{Cos}(Z)]/\mathrm{Cos}(78)\}^{F_\lambda} \tag{4}$$

where $J_\lambda(78)$ is the actinic flux calculated for Z=78 degrees using Equation (3) and F_λ is the "F" parameter corresponding to to λ on Table 11. Note that if Cos(Z) is less than the parameter CZMIN it is set to CZMIN for the purposes of calculating the actinic fluxes using Equation 4. This results in rates at night being slightly above zero, which is found to give more stable calculation results in some simulations.

If it is desired to update the actinic fluxes used with the "SUN" option, the parameters in Table 11 are given in data statements in the simulation program source file PGMS\SOURCE\INT\NEWPHK.FOR. These can be modified as desired, and will be incorporated in models once NEWPHK.FOR has been re-compiled and added to \PGMS\SOURCE\SUBS.LIB, and the model preparation is re-run.

In order to reduce the amount of memory required for each photolysis reaction, a somewhat smaller number of wavelength intervals is used for calculation of photolysis than those shown on Table 11. The wavelengths that are used to define the intervals for photolysis rate calculations are indicated by the "*"'s on Table 11. The photolysis rate calculation intervals are the same as those on Table 11 for wavelengths less than 350 microns and between 420 and 500 microns, but two intervals are lumped together between 350 and 420 microns, and all the intervals above 500 microns are lumped together. The actinic fluxes for the larger intervals are derived by summing those in the intervals on Table 11 that they include. For example, the actinic flux used to calculate the photolysis rate corresponding to 0.350 microns refers to the total flux between 0.350 and 0.360 microns, and is the sum those calculated using the parameters on Table 11 for 0.350 and 0.355 microns. Likewise, the flux corresponding to 0.5 microns is the sum of those calculated using the parameters on Table 11 for all the intervals at or above 0.5 microns.

Calculation of photolysis rates using the "SUN" option. The photolysis rates are calculated for a particular zenith angle by summing over each wavelength interval the product of the actinic flux for that interval and the product of the absorption coefficient and quantum yields for the photolysis reaction in the corresponding interval. As discussed above, absorption cross section and quantum yield data are read in at different wavelengths for each photolysis reaction, and are given as values for the particular wavelength, not averages for various intervals. The following procedure is used to calculate the photolysis rates using these data and the actinic fluxes for the various intervals calculated as described above.

- The product of the absorption cross section and quantum yield is calculated for each input wavelength given in the .PHF file for the reaction. This is referred to as the "action spectrum" in the subsequent discussion.

Table 11. Parameters used for the calculation of actinic fluxes as a function of zenith angle using the model simulation program's "SUN" option.

λ [a]	Parameters [b]				λ	Parameters [b]			
(microns)	A	B	E	F	(microns)	A	B	E	F
0.295 *	4.71e+16	5.253	0.046	2.0	0.415	1.81e+17	0.372	0.004	1.5
0.300 *	4.98e+16	3.037	0.044	2.0	0.420 *	3.50e+17	0.357	0.003	1.5
0.305 *	6.13e+16	1.981	0.043	2.0	0.430 *	3.54e+17	0.339	0.002	1.5
0.310 *	7.33e+16	1.354	0.035	2.0	0.440 *	4.06e+17	0.326	0.002	1.5
0.315 *	7.54e+16	1.057	0.034	2.0	0.450 *	4.60e+17	0.324	0.002	1.5
0.320 *	8.06e+16	0.896	0.034	1.8	0.460 *	4.66e+17	0.312	0.001	1.5
0.325 *	1.04e+17	0.791	0.032	1.6	0.470 *	4.72e+17	0.301	0.001	1.5
0.330 *	1.03e+17	0.722	0.030	1.5	0.480 *	4.64e+17	0.292	0	1.5
0.335 *	9.68e+16	0.671	0.027	1.5	0.490 *	4.62e+17	0.285	0	1.5
0.340 *	1.00e+17	0.634	0.025	1.5	0.500 *	4.80e+17	0.290	0	1.5
0.345 *	9.75e+16	0.605	0.023	1.5	0.510	4.69e+17	0.285	0	1.5
0.350 *	1.08e+17	0.578	0.020	1.5	0.520	4.75e+17	0.282	0	1.5
0.355	9.69e+16	0.556	0.018	1.5	0.530	4.75e+17	0.281	0	1.5
0.360 *	1.05e+17	0.533	0.016	1.5	0.540	4.66e+17	0.278	0	1.5
0.365	1.26e+17	0.513	0.015	1.5	0.550	4.71e+17	0.280	0	1.5
0.370 *	1.12e+17	0.495	0.013	1.5	0.560	4.75e+17	0.279	0	1.5
0.375	1.22e+17	0.476	0.011	1.5	0.570	4.85e+17	0.280	0	1.5
0.380 *	9.96e+16	0.458	0.010	1.5	0.580	9.80e+17	0.273	0	1.5
0.385	1.08e+17	0.443	0.009	1.5	0.600	9.94e+17	0.273	0	1.5
0.390 *	1.10e+17	0.429	0.008	1.5	0.620	9.83e+17	0.258	0	1.5
0.395	1.31e+17	0.414	0.007	1.5	0.640	1.00e+18	0.251	0	1.5
0.400 *	1.56e+17	0.409	0.007	1.5	0.660	1.02e+18	0.245	0	1.5
0.405	1.73e+17	0.394	0.005	1.5	0.680	1.00e+18	0.236	0	1.5
0.410 *	1.81e+17	0.384	0.005	1.5	0.700	1.51e+19	0.236	0	1.5

[a] * = Wavelength where total absorption cross section x quantum yield product is calculated for the interval between this and the next wavelength (or 1 micron for the last wavelength)

[b] Parameters for calculating actinic flux in the interval between the wavelength in the same row and the wavelength on the row below as a function of zenith angle using the fitting function given in the text. The parameters given for the last wavelength are for the interval 0.7-1 micron.

- The averages of the action spectrum are calculated for each of the wavelength intervals for which the actinic fluxes are calculated, defined by the wavelengths with the "*"'s on Table 11). For the purpose of this calculation, the action spectrum is assumed to consist of straight lines between each of the wavelengths where the absorption cross sections and quantum yields are input.
- The photolysis rate is calculated as the sum of the product of the average action spectrum x actinic flux over each of the intervals.

The actinic fluxes and photolysis rates are calculated for each time step in the simulation when the zenith angle changes.

Calculation of photolysis rates using the "P(Z)" option. The P(Z) option is used if it is desired to calculate solar photolysis rates using any set of actinic fluxes is provided by the user. If this option is specified, parameters for calculating the photolysis rates are read from a *Lightmodel*.FZS file whose name is specified in the "P(Z)=*Lightmodel*" input to the simulation program, and which is prepared as part of the model preparation process as described in Sections 2.4.3 and 3.2.8. In this case, the actinic fluxes as a function of solar zenith angle for the conditions of the simulation have already been input and used to calculate the photolysis rates as a function of zenith angle for the mechanism, which in turn were used to derive the parameters in the *Lightmodel*.FZS file.

The parameters used to calculate the photolysis rates are designated A, B, E, F, and Z^{low}, with the first 4 being analogous to the A-F parameters used for calculating actinic fluxes given in Equation (3) above and on Table 11, and Z^{max} being analogous to the 78 degrees used in Equation (4) above. The photolysis rates, k^{phot}, at zenith angle Z are then calculated from these parameters as follows:

$$k^{phot}(Z) = A \cdot \exp\{-B \cdot \mathrm{Sqrt}([1+E]/[\mathrm{Cos}(Z)^2+E])\} \qquad \text{if } Z \leq Z^{low} \quad (5)$$

$$k^{phot}(Z) = k^{phot}(Z^{max}) \cdot (\max[\mathrm{CZMIN}, \mathrm{Cos}(Z)]/\mathrm{Cos}(Z^{max}))^F \qquad \text{if } Z > Z^{low} \quad (6)$$

- The photolysis rates used to derive the values of the A-F and Z^{max} are calculated using the JZTOKZ program, using actinic flux values input for various wavelengths and zenith angles and the absorption cross section and quantum yield data for the various photolysis reactions. The method used by JZTOKZ to calculate the photolysis rates from the actinic fluxes and cross section and quantum yield data is similar to that used to by the model simulation program to calculate the rates using the Peterson (1976) actinic fluxes, except in this case the actual actinic fluxes are used rather than fluxes calculated by fitting parameters. The specific procedure is as follows. The actinic fluxes are read in for various wavelength intervals from the .JZS file. The format of these files, and the wavelength intervals used, are given in Section 3.2.8.
- The product of the absorption cross section and quantum yield, or the action spectrum, is calculated for each input wavelength given in the .PHF file for the reaction.
- The averages of the action spectrum are calculated for each of the wavelength intervals for which the actinic fluxes are input. For the purpose of this calculation, the action spectrum is assumed to consist of straight lines between each of the wavelengths where the absorption cross sections and quantum yields are input.
- The photolysis rate is calculated as the sum of the product of the average action spectrum x actinic flux over each of the intervals.

The JZTOKZ program outputs these photolysis rates in a .KZS file. The KZSFIT program reads this file and finds the A-F and Z^{max} parameters that best fit these photolysis rates as a function of zenith angle, and outputs the parameters in the .FZS file that is used as the input to the model simulation program.

3.3.6. Model Simulation Output Files

The model simulation programs can produce several types of output files, depending on the options specified in the simulation input (.INT) file as indicated on Table 9. By default the only output file produced is the listing (.OUT) file, but in practice the listing output is usually not used and is not produced unless errors occur, and calculated concentration-time data are output in .CSV or calculation data (.CDA) files as specified using SAVE commands. These types of output files and their uses are summarized below.

Simulation listing (.OUT) files. By default the simulation programs produce an output listing file named *intfile*.OUT, where *intfile* is the name of the .INT file that has the primary input for the simulation. (For example, the command "INT TEST" will produce a file called TEST.OUT, depending on the parameters in TEST.INT.) These are ASCII files formatted for making printouts using 132-column printers back in the days when such printers were widely used, and are generally not used currently except for debugging purposes. The contents of this file, and whether it is produced at all, depend on the parameters in the "Listing output (.OUT) file control" section of Table 9. By default, the file will contain first a complete listing of the model reactions, model species and parameters, will list the input data and options, and then list the concentrations of the model species at the initial and various output times in the simulations. Additional output can be produced depending on the output file control options indicated on Table 9. These files are not formatted for convent use by other programs for display or analysis of simulation results, and generally SAVE options are used to produce more conveniently formatted simulation data (.CSV or .CDA) files as discussed below.

Simulation data files (CSV format). Comma-separated (CSV) files with simulation results are output if the "SAVE CSV" option is specified in the simulation input (.INT) file. These files are created in the same folder where the simulation is run, i.e., the folder with the .INT file. These files are formatted so they can be readily loaded into spreadsheet programs for analysis and plotting of simulation results. The first record consists of the following, separated by commas: (1) number of species output; (2) name of .INT input file; (3) default output folder name for .CDA files (not applicable for CSV files); (4) name of the model (model simulation program); and (5) date and time of the simulation. The second record is the header of the data rows, and consists of the word "Minutes" for the time column, and the name of all the species whose data are output. Note that the species names are given in a fix column format with 10 columns, so are truncated with blanks. The subsequent columns contain the simulation times in minutes and then the concentrations or values for each time that is output.

By default, the output data consist of concentrations of active, buildup-only, dummy, and the first 5 steady-state species, in that order, if applicable. (The ordering of the species within the various types is determined by the order of their input in the model preparation program input.) If the "SAVE RATE" or "SAVE INTRATE" option are specified, the rates or integrated rates of each of the reactions at the output times are output, using the label "R#*rxnlbl*" or "IR#*rxnlbl*", respectively, where *rxnlbl* is the label for the reaction specified in the reaction input to the simulation program (see Section 3.2.3).

Note that some spreadsheet programs will not load all the data in the CSV files produced directly by the simulation programs if there are more than 255 species that are output. In these cases, use of .CDA output (discussed below) followed by use of the CDA2CSV or CDA2CSV2 program, would give better results. These programs are discussed in Section 3.5, below.

Simulation data files (CDA or standard format). Standard format simulation data (.CDA) files are output if the "SAVE" option is specified without giving the "CSV" option. This is the recommended format if it is desired to use other programs or spreadsheets supplied with this software to analyze or plot the data. These files contain the same data as in the CSV files discussed above, but the format and location of these files are different. Note that "SAVE RATE" or "SAVE INTRATE" will also cause the rates of the reactions to be output as indicated above in the case of the CSV files.

CDA files are ASCII files with data in a fixed width format. The first record contains the code "A8" that indicates the file format, the number of species output, the name of the input .INT file, the name of the parameter file or output file folder, the model or simulation program name, and the date and time, with Fortran format code ("A8",I5, 3(1X,A8),1X,A16,1X,A). The next set of records contain the header data with the name of the species only, 8 species per line, with the Fortran format (8(A10,1X)). Following this are the records for the calculation data for each simulation output time. For each time the first record is the time in minutes (format F10.2), followed by records with the concentration or value data, 8 species per line, in the format (1PE10.3,7E11.3). The user generally does not generally need to be concerned with the format of these files because they are intended as input for data reading programs.

By default, CDA files are output in folders named either by the named parameter file used when the simulation is run or in a folder named "CALCS" if no parameter file is specified. The root for the simulation output file folders is the folder given in the "CAL=" input in the parameter input (see Section 3.1) or the folder where the simulation program was run if no "CAL=" input record is present. For example, if MODELING.PRM has the option "CAL=C:\MODEL\CALCDATA\", the simulation is run using the command "INT RUN1 SAPRC07" (input from RUN1.INT and using parameters in SAPRC07.PRM), and RUN1.INT contains the option "SAVE", then the calculation output is C:\MODEL\CALCDATA\SAPRC07\TEST.CDA. Note that this allows the calculation results from different models to be saved in different folders, and results of any calculation whose results are saved in CDA files can be retrieved given the input file name and the model parameter file used as discussed in the following section.

Options are available for the simulation program to output the simulation data .CDA files in different locations with different names other than those indicated above, though generally this is not done. Available options are given in Table 9.

3.3.7. Simulation Program Command Line Options

Although it is normally recommended that the model simulation programs be called by INT, INTRUN, or REACTCAL, they can also be invoked directly or through use of a user-written interface program or macro. Several command line options are available when called directly or by another program, and these are discussed in this section.

The command to call the simulation programs directly is "*programname inputfile* [*option(s)*]", where *programname* is the full path name of the simulation program's .EXE file, *inputfile* is the name of the .INT input file (i.e., input from *inputfile*.INT), and *options(s)* can include one or more of the following:

- "SAVE=*file*": The output file is named *file* and is in the local folder, rather than named by the input file name and in the normal output location depending on the name of the parameter file (if applicable) and output options given in the input file. If the .INT file has the "SAVE" option the output file is *file*.CDA and is in the .CDA format, and if it has "SAVE CSV" or no "SAVE" option the output file is *file*.CSV.
- "/RATE": The rates of the reactions are saved in the output file, just as they would occur if "SAVE RATE" were in the input file, as discussed in Section 3.3.1.
- "/INTRATE": The integrated rates of the reactions are saved in the output file, just as they would occur if "SAVE RATE" were in the input file, as discussed in Section 3.3.1.
- "MAX *species*": The only output of the program is the file STATUS.TMP, which contains the maximum and final concentrations of the model species named *species*. Options related to outputs in the .INT file are ignored. This is used, for example, by MROPT2 to determine the maximum ozone concentrations calculated using various inputs.
- "REACTCAL": Suppresses reading the parameter file named on the command line. Used when invoked by REACTCAL and is needed because REACTCAL only uses the option that would otherwise refer to the parameter file to determine the location of output files, but not generally used otherwise. (See Section 3.6 for a discussion of REACTCAL.)

- "*prmfile*": Any option other than those listed above is taken to be the name of a parameter (.PRM) file. Options are read from *prmfile*.PRM after reading the default parameters from MODELING.PRM, and over-write those in MODELING.PRM if different (unless the REACTCAL option is also given). The other options can be given after this, but only one *prmfile* can be given. This also controls where the .CDA output file is located (if applicable), as discussed in Section 3.3.6.

For most modeling application the user will not invoke the model simulation programs directly, but instead use INT, INTRUN, and REACTCAL, which in turn invoke these programs and in some cases use these command line options. However, programmers who want to write additional programs to aid in modeling applications may the command line options useful.

3.4. Detailed Model Species and Lumping Processing Programs

In the context of this software, detailed model species (DMS) refer to model-independent names given to the many hundreds of individual organic compounds or simple isomeric mixtures of such compounds whose reactions may be modeled in atmospheric simulations. Use of model-independent compound designations allow for preparing model-independent inputs that are useful for simulations using any model. These model-independent compound names serve as input to the lumping processing programs that are used to assign the model species used to represent these compounds in the model simulations. The model-independent species names, and the files and programs useful for providing information about them or defining complex mixtures are discussed first, followed by a discussion of the lumping programs used to process this input for use with specific models.

3.4.1. Detailed Model Species and Composition Files

There are many ways that an organic compounds can be named and designated, though unique designations, such as standard IUPAC nomenclature or INCHI strings can yield long character strings that are awkward to use as input to modeling programs. Instead, for SAPRC model development and intercomparisons with other mechanisms, we have established a set of 2-8 character SAPRC detailed model species (DMS) names for all the compounds and simple isomeric mixtures that were inputs to ambient or chamber simulations, or for which reactivity values were of interest. Currently this includes names for >900 compounds and ~300 isomeric mixtures. These are listed in the ASCII file MECH\SAPRC07\DMS.PRM, along with their carbon numbers, molecular weights, and other information. DMS.PRM is used as input the RENORCMP program, discussed below, or by the DMSRCT reactivity summarizing program discussed in Section 3.6, to obtain carbon number or molecular weight information. These files should be consulted when preparing input files for simulations with new compounds so an existing name would be used if available. If a new compound is added, a new unique 2-8 character name should be assigned to it (using the same naming requirements as used for model species -- see Section 3.2.5) and it should be added to the DMS listing files for future use.

DMS.PRM file. DMS.PRM is a plain text file that starts with comments that are ignored and end with a line containing only a "." in the first column, followed by data lines, with each containing a information about the compound in a fixed column format. This includes (1) the compound or mixture name, (2) carbon number, (3) information about the type of species (4), molecular weight, and a compound/mixture code indicating if this is a mixture optionally the type of representation of the compound in the model, input with FORTRAN format code " (A8,2X,I4,1X,F8.2,12X,A4)". The compound name and other information can be included in the input lines after these data. In addition to names of individual compounds, the file can also contain names for simple and complex mixtures. The composition of the complex mixtures can be obtained in .CMP files in the location indicated by the

"CMP=" input in MODELING.PRM. These are indicated using the compound/mixture code "Mix", and this is used by the DMSRCT program to derive incremental reactivities for mixtures (see Section 3.6.4).

Composition Files. Most ambient modeling scenarios and some experiments to be modeled utilize complex mixtures of many organics. For preparing inputs for such simulations, it is often useful to separately specify the total amount of the mixture present (in units such as ppmC, or mass) and the relative composition (speciation) of the mixture in terms of individual compounds to which model species have been assigned. For this purpose, it is useful to utilize composition (.CMP) files that give the relative compositions in appropriate units, such as moles of each compound in one mole carbon of the mixture. These composition inputs can then be multiplied by the total amount to obtain the input concentrations. This is particularly useful for ambient simulations when calculating effects of varying total VOC inputs, as is done when deriving the MIR and EBIR scenarios for reactivity calculation applications (see Section 3.6) or when modeling multiple experiments using varying amounts of the same complex mixture.

The distributed folder REACT\CMPFILES contains a number of composition files used either for reactivity scale calculations or experiments with complex mixtures. These include mixtures representing anthropogenic VOC emissions, ambient mixtures aloft, and various complex mixtures for which reactivity values have been calculated or used in chamber experiments. These are ASCII files that contain the SAPRC detailed model species (DMS) name of the compound or isomeric mixture and then its relative amount in molar units. Most files are normalized to 1 ppmC, so the data contain moles of compounds per mole carbon of mixture. These files can also contain input with "!" in the first column, which are treated as comments and ignored by programs reading these files.

The .CMP files can be used directly as input to the lumping pre-processing programs to obtain the model species used to represent the mixture. They generally give relative compositions in terms of moles compound per mole carbon mixture, and thus are multiplied by factors giving the total amounts of mixture present to obtain absolute concentration inputs for the model species (e.g., see the "#FAC" command on Table 12, below). For example, to model an experiment containing 2 ppmC of the "Aromatics-100" solvent whose speciation is given in REACT\CMPFILES\AROM100.CMP, the input to the lumping program (described below) could contain:

```
.LUMP
#FAC 2.
@root\REACT\CMPFILES\AROM100.CMP
.
```

where *root* is the absolute location of where the distributed files was installed. Alternatively, the third line could be replaced by @CMP\AROM100.CMP, and the mechanism-independent parameter file MODELING.PRM could contain the line "@=CMP\ *root*\REACT\CMPFILES\", which has the effect of replacing "@CMP\" with "@*root*\REACT\CMPFILES\", (see Table 3) giving the same results. Note that this must be included within .LUMP input to cause the lumping processing programs to convert these model-independent inputs into model species.

RENORCMP program. As indicated above, .CMP files are usually intended to give relative compositions of complex mixtures, with mixtures of organics typically giving moles compound per mole carbon mixture. The RENORCMP provides a method to conveniently normalize compositions in .CMP files, either to give moles per moles carbon or moles per unit mass. The program is run in a DOS box using the command:

RENORCMP *inputCMPfile* [*outputCMPfile*] [*factor*] [*option*]

Where *inputCMPfile* and *outputCMPfile* are names of .CMP files (without the ".CMP"), *factor* is the normalization factor if not 1.0, and *option* is the option. The output file name, factor, and option are optional. If the output file name is not given it is the same as that for the input file, and the input file is over-written by the output file. The default value for *factor* is 1.0, and if *option* is not specified than the normalization is by carbons, i.e., the default is to normalize to moles compound per mole carbon mixture. The available values for *option* are "M" or "PPB". The "M" option causes the results to be normalized by mass, with the output data being moles of compounds per *factor* grams of mixture. The "PPB" option causes the results to be normalized by carbon as with the default, but the results are multiplied by 1000. Note that the output file name and factor must be specified on the command line if the "M" or "PPB" options are used.

The RENORCMP program reads the DMS.PRM to get the carbon numbers or molecular weights for the compounds that are needed for the normalizations. This file must be either in the local folder where the program is run or be specified by "DMS=*fullpathname* input in MODELING.PRM, where the file could have any name or location, but *root*\MECH\DMS.PRM in the distributed files. The program will fail if this file is not found or if a compound name is not found in this file. Note that if a compound name is not in DMS.PRM it almost certainly will not be in the .LCC or .LPC files read by the lumping program to determine the model species used for the compound. Therefore, this program also provides a means to verify that all the model-independent compFound names are recognized as such.

3.4.2. Lumping Processing Programs

As discussed in Section 2.4 and Table 2, and shown on (for example) Figure 2, lumping processing programs LUMPGEN or LUMPINT are used to pre-process model simulation input files to convert input concentrations (or emissions rates) of compounds identified using model-independent detailed model species names into inputs of model species used to represent them in the model being used. In addition to lumping processing, the lumping processing programs can also pre-process input concerning spectral distribution data for simulations where the spectral distribution varies with time or where spectral distribution input is contained in the .INP file rather than separate .SDR files. The former can be useful in simulation of outdoor chamber experiments or in simulations of experiments where the spectrum of the light source varies from experiment to experiment.

Their main inputs are simulation input files that do not depend on the model being used (i.e., refer to names of real compounds and not model species) and lumping control files that determine which model species and parameters go with each compound being simulated. The main output is model-dependent simulation input (.INT) files that can be directly used by the model simulation programs. This is referred to as "lumping processing" in the subsequent discussion. This is useful when evaluating the models against measurement data or simulating real-world or ambient scenarios.

Normally the lumping programs are not run directly, but are called as needed by the INTRUN program, which is used in place of the INT program (discussed in Section 3.3.1) to carry out model simulations where lumping processing is needed. The INTRUN program determines which of these lumping programs needs to be run based on the "LMPPGMS=*program*" input in the parameter file used when the INTRUN program is run (See Table 3 in Section 3.1), where *program* can be either "LUMPGEN" or "LUMPINT".

The LUMPGEN program is used for simpler condensed or fixed-parameter models where the parameters for the lumped model species do not depend on the compounds they represent, as is the case for most current existing chemical mechanisms. The LUMPINT program can also be used for these simpler mechanisms, but is primarily designed for use with mechanisms where at least some compounds are represented using adjustable parameter model species, where kinetic and mechanistic parameters for some of their model species depend on the compounds being represented. Although these programs perform the same functions and use the same model-independent inputs, they use different types of lumping control files as discussed in the following sections.

The inputs to the lumping processing programs giving the simulation inputs are the same as simulation inputs in model-dependent .INT files, as discussed in Section 3.3.1 and shown on Table 9, except that they can also contain lumping processing input as shown on Table 12. The .LUMP input is used for to give model-independent species concentrations that need to be converted to model species using the lumping programs, and #VARYSD or #SDADJ are optionally used for processing spectral distribution input as described on Table 12. These are discussed in the following section, where the two lumping programs are discussed in more detail.

3.4.3. INTRUN program for Simulations with Lumping Processing

The INTRUN program is used to convert a model-independent input file to an input file that can be used as input model simulations, and then run the appropriate simulation program as indicated in the parameter file. It is run using the command

INTRUN *INPfile PRMfile* [*inserted data file* | *options*] [*options*]

where *INPfile* is the model-independent simulation input (.INP) file, *PRMfile* is the model-dependent parameter (.PRM) file, and the optional inserted data file and other options are discussed below. Note that the .INP and .PRM extensions are not given with their names. The model-independent input (.INP) files have the same format as .INT files as discussed above in Section 3.3.1, except that model-independent species names are used, and additional pre-processing input can be included as discussed below. Model parameter files are discussed in Section 3.1, and in this case it contains information on the lumping program and files to be used, as well as the model simulation program to be used and the parameters it uses.

The specific parameters in model-dependent .PRM files used by INTRUN are listed in Table 3 in Section 3.1. INTRUN also reads parameters from MODELING.PRM, which in general are parameters that are model independent, which includes the "INP=*folder*" input that gives the location of the model-independent input (.INP) files. The model-dependent input parameters used by INTRUN include "DFLMOD=*model*" to indicate the simulation program to be used, "MOD=*folder*" to indicate the location of the simulation program executable and files, "LMPPGM=*program*" to indicate which lumping program (LUMPGEN or LUMPINT) to use, and several others (see Table 3). The default is LMPPGM=LUMPGEN, for use with fixed-parameter mechanisms[4]. In addition, the model-dependent parameter file must have parameters needed by the lumping program used, as discussed below for the specific programs.

The model-independent simulation input (.INP) files have the same format as .INT files as discussed in Section 3.3.1 and can include the same types of input as listed on Table 9 and Table 10 in that section. The lumping programs just copy this input from the input .INP files to the output .INT file without modification. In addition, the .INP files can contain additional input in the simulation condition input section, as listed in Table 12. The lumping programs process this input as discussed below and outputs the model-dependent results in the corresponding location of the .INT file that they create. The resulting .INT file is then used by INTRUN as the input to the model simulation program that it runs.

---

[4] Files downloaded prior to January 26, 2015 will have LUMPINT as the default lumping program. If LUMPGEN is to be used as discussed in this document, the input "LMPPGM=LUMPGEN" must be specified in the parameter file or INTRUN will fail.

Table 12. Summary of preprocessing input to INTRUN used by lumping that can be given in the simulation condition input section of model-independent simulation input (.INP) files. These are compatible with both LUMPGEN and LUMPINT.

Code [a]	Description	
	Lumping Processing	
.LUMP [NOLUMP] [#FAC *value*] *species1 conc1* *species2 conc2* … .	The following input records are to be processed by the lumping program based on the input in the lumping control file before being output to the .INT file. The following input can be included. Input is terminated by a "." in the first column.	
	NOLUMP	This optional input, which must immediately follow the .LUMP line before any other lines, forces use of a separate model species for all compounds represented by adjustable parameter model species, resulting in each such compound being represented explicitly in the simulation. See the discussion of LUMPINT in Section 3.4.5. This is ignored (with no error or warning messages output) if the input is being processed for a non-adjusted-parameter mechanism using LUMPGEN.
	#FAC *value*	Optionally multiplies all the input concentrations by this value. For example, can be used to specify the total amounts of a mixture input if the subsequent input gives the relative compositions of the mixture in terms of compounds in the mixture.
	Species conc	The name and concentration of a compound. The program converts this to the concentration of a model species and (for LUMPINT) mechanistic parameters if the parameters of the model species depends on the compound(s) being represented.
.LUMPPRE *prefix*	The model species names output are given the 1-4 character prefix *prefix*. This is useful if relative concentrations are to be converted to parameters used for group or aloft input for models prepared with the AIRSHED option, as discussed in Section 3.2.10. For example, the prefix "A#" means that the output gives aloft concentrations, and the prefix "R2#" means the output gives relative compositions of species in a mixture defined by the 2nd group in the model, such as total VOCs. The default is no prefix.	
	Spectral Distribution Processing	
#VARYSD=*vsdfile*	Read spectral distribution data as a function of time from *vsdfile*.VSA, located in the folder indicated by the "VARYSD=*folder*" input of the parameter file (generally MODELING.PRM since these .VSA files are model-independent). The lumping program uses this, together with the absorption cross sections and quantum yields read from the model simulation program's .MOD file, to calculate photolysis rates as a function of time, which are output as time-varying coefficients in the .INT file. VSA file format: These are ASCII files with fixed column input, with the Fortran format codes indicated in parentheses. The first line contains a label that is ignored, the number of wavelengths for which light flux values are given, and a factor to multiply all the factors by (8X,I4,E12.4). This is followed by the wavelengths in microns, 10 per line (10x,10F12.4), with one more wavelength	

Table 12 (continued)

Code [a]	Description
	being input that the number of fluxes because the fluxes are given for intervals. This is then followed by a set of records giving the time in minutes and the light fluxes at the various wavelengths at that time, 10 per line (F8.1,2X,10F12.6)/(10X,10F12.6)). The light fluxes are fluxes in the corresponding wavelength intervals in units (after multiplying by the factor in line 1) of photons cm^{-2} min^{-1} per interval. The interval for the first flux is the first and 2nd wavelength input, so the number of wavelengths input is one greater than the number of fluxes.
#SDADJ= *input* #END	Inputs factors to adjust the spectral distributions in the low wavelength range. The inputs consist of wavelengths (in microns) and factors, with wavelengths ascending. The relative values of the factors are significant but the absolute values are not. This should only be used when spectral distributions are input as relative spectral distributions, not absolute actinic fluxes. This is used in simulations of certain smog chamber experiments where the low wavelength intensities of the light source is changing over time, but the higher wavelengths intensities are not. See Section 3.3.4 for a discussion of spectral distribution input for simulations with constant spectral distributions. These factors are used on the spectral distributions input using the ".SD-*sdrfile*" input (see Table 10 in Section 3.3.1).

3.4.4. LUMPGEN Program for Simple Lumping Applications

The LUMPGEN program is used for lumping and (if applicable) special spectral distribution processing for models using simple or fixed parameter lumping approach. Such models represent compounds using one or more model species but do not have model species parameters that depend on the compounds being represented. This is the case for almost all current atmospheric chemical mechanisms except for adjustable parameter versions of the SAPRC mechanism, for which LUMPINT must be used. Normally LUMPGEN is not run directly by the user, but is called by INTRUN as the first step of the simulation process, as discussed above.

The LUMPGEN program is called by INTRUN to pre-process lumping input if the "LMPPGM=LUMPGEN" option is specified in the parameter file (Table 3), and usually does not have to be run directly. To run LUMPGEN directly, use the command

LUMPGEN *inputfile outputfile parmfile* [PREFIX=*char*]

where *inputfile* and *outputfile* are is the full names (including extension and also the path if different from the folder where the program is run) of the morel-independent simulation input and the model-dependent output files, respectively, *parmfile* is the name of the model parameter (.PRM) or lumping control (.LCC) control file used, and the PREFIX option allows for prefix characters before the names of the model species that are output. The PREFIX option on the command line has the same effect as the .LUMPPRE input when run using INTRUN, as indicated on Table 12. The input and output files always have the extensions .INP and .INT, respectively, when run using INTRUN but can have any extension when LUMPGEN is run directly. The program tries to open *parmfile*.PRM to get the options that it needs to run, which includes the name of the LUMPGEN lumping control (.LCC) file that contains the lumping assignments for the model. This is specified in the "LCC=*LCCfile*" parameter option. If *parmfile*.PRM does not exist in the local folder, then LUMPGEN assumes the lumping control file is *parmfile*.LCC. The program fails if there is no LCC option in the parameter files (either *parmfile*.PRM or MODELING.PRM), or if there is no *parmfile*.PRM and *parmfile*.LCC does not exist. Note that unlike most other programs, LUMPGEN does not produce any output to the terminal if it runs successfully, though it will give error messages to the terminal if it fails.

The formats of the model simulation inputs and output files have already been discussed in conjunction with the INTRUN program in the previous section. Briefly, the input and output files are simulation input files as discussed in Section 3.3.1, except that the input files can also contain additional input as indicated on Table 12, above. Any input except for that in Table 12 is copied to the output file without modification, while input listed on Table 12 is processed to produce model-dependent input for the output file. (Note that the PREFIX option only affects output of model species names when processing .LUMP input as indicated on Table 12.)

The LUMPGEN lumping control files must have the extension .LCC and are in the location given by the "MECH=*folder*" option of the parameter file or in the location where the command is run if no such option is given. It is specified either in the parameter file using the "LCC=*LCCfile*" option or directly on the command line as discussed above. It is required input to LUMPGEN.

Lumping control files (such as LUMPGEN .LCC files or LUMPINT .LPC files discussed in the following section) are ASCII files that use records with various keywords for different types of input, in most cases with subsequent input lines, or input from files that are named on the keyword lines. The types of input that can be included in these types of files are listed and described in Table 13.

For use with LUMPGEN, the main inputs needed are the SPECIES command listing the lumped model species names to be output, and the ASSIGN command giving the model species assigned to each model-independent species name. The model species assignments can either be to a ingle model species name if the compound is represented by a single model species on a molar basis, or to a list of names and coefficients, each separated by one or more spaces, if otherwise. Coefficients give moles of model species per mole of compound, and are indicated by "#*number*" before the species name, where the "#" character is required to indicate it is a coefficient and not a model species, and *number* must be decodable as a valid floating point number. Coefficients need not be given if the assignment is one mole model species per compound.

An example of a lumping control (.LCC) file for a fixed parameter version of SAPRC-07 is shown in Example 7 in Section 3.4.7, below. The format for other mechanisms is the same. Note that some mechanisms, such as Carbon Bond use more than one model species to represent a compound, and others, such as RADM-2, use reactivity weighting for some compounds, where the moles of model species used for the compound depend on the reactivity of the compound being represented (See Carter, 2020 for current model species assignments for various mechanisms for emitted compounds). In the case of SAPRC-07, a single model species is used for most individual compounds, but some detailed model species refer to simple mixtures that may require more than one model species, depending on the components.

3.4.5. LUMPINT Program for Adjustable Parameter Lumping

The LUMPINT program is used for lumping and (if applicable) special spectral distribution processing for models using the adjustable parameter lumping approach for at least some of the model species. In this approach, the model uses "adjustable parameter" model species to represent compounds given their mechanistic parameters, such as shown in Example 6 in Section 3.2.12. The LUMPINT program reads a lumping control .LPC file that includes GROUP input listing names used for lumped parameter model species and their parameters, and PARM input giving names of files containing species names and their mechanistic parameters, as summarized in Table 13. This input is then used to derive the concentrations of the lumped species from those of the species they represent, and

Table 13. Summary of inputs in lumping control files used by LUMPGEN or LUMPINT.

Keyword	Description
	<u>LUMPGEN or LUMPINT input</u> (.LCC or .LPC files)
SPECIES	Subsequent input lines give a list of model species that may be output, one per line, terminated a line with "END". This should include all model species referenced in ASSIGN input in LPC files, or the model species output will be given as a comment. The input line can optionally include the carbon number and molecular weight of the species, but use of this information is no longer supported.
PREFIX *chars*	Model species names are output with the 1-4 character prefix *chars*. This may be useful if aloft concentrations or group speciation input is to be output for airshed model simulations as discussed in Section 3.2.10.
! *comment*	Any input line with a "!" in the first column is ignored, so can be used for comments.
	<u>LUMPGEN input</u> (.LCC files only)
ASSIGN [*file*]	Inputs lumped model species assignments for fixed parameter mechanisms. Subsequent records contain the model-independent name of a compound, followed by one or more spaces and then the model species assignments in terms of moles model species per mole compound, and terminated by the end of file if a file name is given for input or an "END" line if not. See text in Section 3.4.4 for more information in the format of this input and an example.
	<u>LUMPINT input</u> (.LPC files only)
NOLUMP	Forces use of a separate model species for all compounds represented by adjustable parameter model species, resulting in each such compound being represented explicitly in the simulation. See the discussion of LUMPINT in Section3.4.5. This option can also be specified when calling LUMPINT using INTRUN (Table 12). Note that rate constant input in PARM input is ignored if this option is specified.
REPLACE [*file*]	Inputs lumped model species assignments for compounds represented by non-adjusted parameter model species, or gives "lumped molecule" representations for compounds that are represented as having the same mechanism as other compounds. In the latter case the lumped model species used is determined by the GROUP and PARM input discussed below. Each input line consists of the model-independent species name, one or more spaces, and either the model species assigned to the compound or a list of one or more compounds used to represent the compound using the "lumped molecule" method. Compounds represented by mixtures can also be input as discussed in Section 3.4.1. The REPLACE input is terminated by end-of-file if a file is named, or by an "END" line if a separate input file is not named. Although this can also be used by LUMPGEN, the ASSIGN command is recommended for that application. More than one set of REPLACE input can be included in the file. REPLACE input is processed iteratively, allowing replacement or model species assignments of species names that have been given as a replacement in other input lines, regardless of the order.
GROUP *parms*	Used to specify information and parameters for lumped parameter model species. Each input line refers to a single model species with adjustable parameters. These input lines must include the following inputs following information following the "GROUP" keyword, separated by spaces:

Keyword	Description	
	Group no.	An integer between 0 and 9 indicating the group of compounds that may be represented in this group. The compounds and parameters are listed in separate "PARM" input lines that reference this group number.
	Lumped Species name	The lumped parameter model species name to be output
	Parameter Prefix	The prefix used for names of all mechanistic parameters derived for this model species based on those of the compounds they represent.
	Processing option 0	This is used to indicate the option for processing parameters. Currently only option type "0" is supported and documented.
	Minimum kOH	The minimum OH radical rate constant (kOH) for compounds to be represented by this model species. If not given, all compounds listed in PARM commands with this group number are represented by this species. The maximum OH radical rate constant represented by this species is determined by the next highest minimum kOH for model species in this group. The rate constants should be given in the same units as used in the simulations (usually ppm^{-1} min^{-1}).
	See Section 3.4.5 for more information and an example.	
PARM *parms*	Used to associate each group of compounds that are represented by lumped parameter model species with a group of model species specified.	
	Group No.	The group of compounds that may be represented in this group. This must match the group number in at least one GROUP input line, which give the model species for the group. Must be between 1 and 9.
	File format type 0	A code indicating the format of the mechanistic parameter file. Currently only type "0" is used and documented.
	Parameter file name	The name of a file containing model-independent names of compounds and the mechanistic parameters to be used for them. These are given .GNA extension names by convention, but extension must be given. The location of the file can be specified using the MECH= parameter of the parameter file or MODELING.PRM (Table 3).
	There must be only a single space separating these inputs on the line. See Section 3.4.5 for more information, a discussion of parameter files, and examples.	
TEMP *degk*	Then default temperature used when outputting rate constants for mixtures of compounds whose rate constants depend on temperature.	
INTFMT INTOH	These inputs are no longer supported.	

the values of their parameters based on the parameters of the input species. If more than one species in the input file is represented by the same lumped parameter model species, the output concentration of the model species is the sum of concentrations of the input species, and its parameters are weighted averages of those for the input compounds they represent.

As with LUMPGEN, LUMPINT is called internally by the INTRUN program if the parameter file contains the "LMPPGM=LUMPINT" option, and is not generally run directly. To run it directly, use the command

LUMPINT *inputfile outputfile parmfile*

where *inputfile* and *outputfile* are is the full names (including extension and also the path if different from the folder where the program is run) of the model-independent simulation input and the model-dependent output files, respectively, *parmfile* is the name of the model parameter (.PRM) or lumping control (.LPC) control file used. The PREFIX option is not available for LUMPINT, since that can be specified in the lumping control file. The other considerations involved with running LUMPINT are the same as discussed for LUMPGEN in the previous section.

Table 13 includes input in lumping control (.LPC) files used by LUMPINT. As indicated there, the REPLACE command is used to assign compounds to model species that do not have adjustable parameters, similar to the ASSIGN command in lumping control files for LUMPGEN. However, the REPLACE command can also be used in cases where compounds are represented on a mole-for-mole basis by mechanisms for other compounds (the "lumped molecule" method), or where names are input that refer to mixtures of compounds. In those cases input using the specified model-independent name is replaced by input of the replacement name or mixture, and the model species used to represent them are determined by GROUP and PARM input for the replacement species. Mixtures are indicated using lists of compounds followed by their relative amount (if not mole-for-mole), separated by spaces. If a mixture involving many compounds is included, and the list won't fit in an 80-character line, then the list can be continued in subsequent lines as long as the first character of the continuation line is blank. In that case, the last character of the first input line end with a "&" following a space, and the first character of the continuation line must be blank Examples of lumping control input are given in Section 3.4.7.

Each set of adjustable parameter model species that react similarly, and have the same format for specification of mechanistic parameters, are organized into groups. The PARM commands give the names used for these model species and their adjustable parameter prefixes for the model species in each group, and optionally the minimum OH radical rate constant for compounds to be represented each model species. Each model species named in PARM inputs must also be model species in the mechanism, and whose parameters start with the parameter prefix given in the PARM input. For example, if the PARM input is "PARM GEN1 G1 …", the .RXN input for this model species would look like

```
.RXN
Label) 0.0  ;GEN1 + OH = #G1OHp1 prod1 + #G1OHp2 prod2 + …
…
Label) COEF=G1HV QY=G1HVQY  ;GEN1 + HV = #G1O3p1 prod1 + #G1O3p2 prod2 +
…
.COE
G1HVPa 1.0
G1HVPk 1.0
…
.INS DIFF
'G1HV' = ('G1HVPa' * 'C2CHO') + ('G1HVPk' * 'MEK-06') + …
etc.
```

Note that for SAPRC mechanisms, product yield and other mechanistic parameters associated with adjustable-parameter model species have names constructed by (parameter prefix for model species) + (2-character code for type of reaction) + (2-4 character code for type of product or parameter). For photolysis reactions, the coefficient G1HVQY is used to define the overall quantum yield and G1HV is a coefficient giving the index (as a real number) for the type of photolysis reaction, defined using .COE and .INS DIFF input such as shown above. In this example, the coefficients 'C2CHO' and 'MEK-06' refers to the photolysis rate for higher aldehydes and ketones used in the SAPRC-07 mechanisms, and the parameter "G1HVPa" or "G1HVPk" are set to 1.0 if the respective photolysis rate is to be used..INS DIFF is used to set G1HV because the photolysis rates may change with time in some simulations. In practice, this reaction input is put in general species .RXN files where the "?" character is used in place of the species number (e.g., "OLE1" and "O1" are changed to "OLE?" and "O?"), and global search-and-replace options are used to change the "?" to the index number to prepare the separate .RXN input for each model species in this group.

The PARM commands give names of parameter files (called .GNA files by convention, though the extension must be specified and any could be used) that gives the names and parameter values of compounds that can be represented by model species in this group. If the lower limit OH radical rate constants are given, and the NOLUMP option is not specified in the INTRUN input (see Table 12), they indicate how the model species for this group are to be allocated to compounds in this group based on their OH rate constants, which are given in the parameter (.GNA) files. Compounds with a given kOH are lumped with the model species in this group that has the lowest lower limit rate constant specified in the GROUP input. If the rate constant is lower than the lowest minimum kOH specified for model species in the group, the compound is assumed to be unreactive and is not input. Mechanistic parameter files are discussed in the next section.

If the NOLUMP option is given, the minimum OH radical rate constants given for the model species are ignored, and no each compound represented by adjusted parameter species is assigned its own model species, which means that these compounds are represented explicitly in the model. The first compound input assigned to a group is assigned the first model species in the group, the second is assigned the second species, until the number of model species given in GROUP input for that group has been exceeded, in which case the program aborts. Therefore, if it is desired to explicitly represent several compounds within a group with their own model species, there need to be enough model species defined for that group to represent these compounds. In some cases this requires having more model species in a group that would be necessary if NOLUMP were not specified. In those cases, additional model species are added with no minimum kOH added. These are not used if NOLUMP is not specified if the previous GROUP input for this group is sufficient to cover the full range of OH radical rate constants.

3.4.6. Mechanism Parameter Files for Adjustable Parameter Species

Mechanism parameter files are files containing the names and mechanism parameters for compounds that are represented using adjustable parameter model species in the mechanism. These include most compounds represented using the distributed SAPRC07 mechanism. Such files with parameters with multiple compounds are read by LUMPINT during processing of PARM commands as discussed above, and by convention are given the extension .GNA. In addition, as discussed below in Section 3.6.3, versions of these files containing parameters for only a single compound can optionally be used as input to REACTCAL for the purpose of calculating incremental reactivities of such compounds. These single-compound mechanistic parameter files are given the extension .GN1, but have the same format as GNA as discussed in this section, though they can contain additional information as discussed below and in Section 3.6.3.

Mechanism parameter files are plain text files containing header input listing the parameter names used, input giving names, rate parameters, and coefficient values for individual compounds, and comments. Comments are any input lines starting with a "!" in the first column, and can appear anywhere in the files. The first non-comment inputs must be the list of parameters, where the first line is "PARM *n*", where *n* is the number of parameters that can be input, and the subsequent lines are the names of these *n* parameters, 8 per line, input using the FORTRAN format code 8(A6,2X). For the SAPRC07 mechanism, the first two characters of the parameter name refer to the type of reaction and the rest refer to the product or type of parameter, as shown in the example above. The parameter names output by LUMPINT consist of the prefix specified on the PARM line in the .LPC file, followed by the parameter names used in the .GNA file.

The first line of the inputs defining parameters for a compound contains the model-independent name used for the compound in the model inputs and outputs. These "detailed model species" (DMS) names, which can be no more than 8 characters long, as discussed in the following section. The species name is the only input on this line that is used by LUMPINT, though optionally the line can contain additional information (following at least one space) that can be included with the output if this is a single-compound parameter file read by REACTCAL (See Section 3.6.3).

The subsequent inputs for each compound give the parameter names and values. The parameter names do not include the prefix that designates the model species used (e.g., "G1" in the examples given in the previous section), but indicate the type of reaction (OH, O3, N3, OP, or HV for OH, O_3, NO_3, O^3P, or photolysis) and the type of parameter. For rate constants, the parameter code "K(*type*)", where *type* is the type of reaction [e.g., "K(OH)"], and the parameter values are the rate constant in cm^3 $molec^{-1}$ s^{-1} units, and optionally the Arrhenius A factor in the same units and the activation energy in kcal/mole. Each of these values must be separated by at least one space. The rate constant is treated as temperature-independent if the A factor and activation energy are not included. For other types of parameters, the input is only the parameter name and parameter value, separated by at least a space. The parameter names can be no longer than 6 characters, because the 2-character model species code is prepended to it.

3.4.7. Lumping Control Examples

Fixed Parameter Lumping. Example 7 gives examples of portions of a lumping control file used to assign lumping for fixed parameter mechanism. These files assign one or more model species in the model to mechanism-independent compound names. The "SPECIES" input lists the model species that can be output, and the "ASSIGN" input gives the model species for each compound or simple mixture name. In this example, the model species for "Explicit" model-independent names where each name refers to a singe compound are input first, followed by the "Lumped Molecule" names that refer to simple isomeric mixtures. These are derived from assignments of compounds to these mixture names, using the assignments for the components.

Adjustable Parameter Lumping. Example 8 gives examples of portions of a lumping control file used for processing input to the adjustable parameter version of the SAPRC-07 mechanism, and also portions of files that are referenced for additional input. The REPLACE input shows examples of assignments of compound or mixture names to model species or other compounds or mixtures, as discussed in the second column. Example 8a shows the format used in various cases. Note that if a mixture name is assigned to a mixture of more than 4 compounds then it is necessary to have a second input line, as shown on Example 8a.

Example 8 shows the GROUP and PARM commands used to define model species and give assignments for alkanes and compounds that can be lumped with them (group 1) and for aromatic hydrocarbons (group 2). The CHAMLUMP.LPC file has similar inputs for alkenes (group 3) and for

Example 7. Example of a lumping control (.LCC) file for fixed-parameter mechanisms that are read by LUMPGEN. This example shows selected input in the lumping control file for fixed-parameter SAPRC-07.

```
! Lumped mechanism lumping file for SAPRC07.  Created 27-Dec-2010 12:09
! Created using F:\WORK\SAPRC07\DMSlump.xls
!
SPECIES
CO
HCHO
…
TERP
END
!
ASSIGN
! Explicit species
!
METHANE          CH4
ETHANE           ALK1
…
THIOCARB         ARO1
MDI              INERT
!
! Lumped Molecule species
!
BR-C5            ALK4
BR-C7     #0.500 ALK4     #0.500 ALK5
C4-OLE    #0.500 OLE1     #0.500 OLE2
C9-BEN    #0.666 ARO2     #0.334 ARO1
(etc)
```

photoreactive compounds (group 4). The group 1 model species are named GEN*n* and their parameter prefixes are "G*n*", where *n* is the model species number in that group, and the names and parameters of compounds represented by group 1 model species are given in the file ALKOTH.GNA. The lump type code "0" is required and the 5th parameter is the minimum OH radical rate constant, in ppm^{-1} min^{-1} units. For example, if compounds in this group have an OH rate constants of $2x10^4$ and $3x10^2$ ppm^{-1} min^{-1} then they represented by GEN5 or GEN2, respectively, and Group 1 with rate constants of less than $3x10^2$ ppm^{-1} min^{-1} are treated as unreactive. On the other hand, any group 2 compound listed in AROMATIC.GNA or NAPT.GNA with rate constant of less than $2x10^4$ is represented by GAR1.

The .LPC file shown in Example 8 can also be used when the NOLUMP option is specified in the lumping input in the INTRUN input. In this case, 5 model species are available to represent alkanes and compounds lumped with them (group 1) and 3 are available for aromatics (group 2). Note that if NOLUMP is not specified, all group 2 compounds (listed in AROMATIC.GNA or NAPT.GNA) with OH radical rate constants less than 2 x 10^4 ppm^{-1} min^{-1} are represented by GAR1, those with higher rate constants are represented by GAR2, and GAR3 is not used. This third aromatic model species is included to allow modeling of chamber experiments containing up to three different aromatic compounds without lumping these compounds. The preferred protocol for evaluating mechanisms using chamber data is not to lump the reactants, in order for the evaluation to measure the performance of the chemical model rather than the lumping approach. For that reason chamber experiments containing only single organic reactants or very simple mixtures always have the NOLUMP option specified in their input (.INP) file where lumping input given. However, experiments containing complex mixtures such as full atmospheric surrogates or petroleum distillate mixtures require too many model species for the NOLUMP option and for that reason do not have NOLUMP given in their input files. For that reason, lumping control files used for chamber simulations, such as CHAMLUMP.LPC, must provide for both options.

Example 8. Portions of CHAMLUMP.LPC, used when processing input for modeling environmental chamber experiments using the adjustable parameter version of the SAPRC-07 mechanism.

Input (keywords underlined)	Discussion
TEMP 300.0	The default temperature for deriving temperature-dependent output parameters is 300^{o}K.
SPECIES CO 1 28.01 HCHO 1 30.03 … M-XYLENE N-C6F14 END	List of model species that may be output. The first includes species in the base mechanism (with carbon numbers and molecular weights also shown, though they are not used by LUMPINT), followed by the list of species that are represented explicitly in the model used for chamber evaluations. The "END" record indicates the end of the SPECIES input.
REPLACE FORMALD HCHO ACETALD CCHO … METHANE CH4 END REPLACE OLDNAMES.LPM REPLACE LUMPMOLE.LPM	Input assigning model species for compounds represented using fixed-parameter model species. Three sets of REPLACE inputs are used in this file, the first where the inputs are following the keyword and terminated by the "END" record, and the other two where the input is taken from files. The OLDNAMES.LPM file is used to replace obsolete species names used in some old chamber input files with the names currently used, and LUMPMOLE.LPM file contains "lumped molecule" and assignments for simple mixtures. Example 8a gives examples of input in these files.
GROUP 1 GEN1 G1 0 2.0E+2 GROUP 1 GEN2 G2 0 5.0E+2 GROUP 1 GEN3 G3 0 2.5e+3 GROUP 1 GEN4 G4 0 5.0E+3 GROUP 1 GEN5 G5 0 1.0E+4 PARM 1 ALKOTH.GNA GROUP 2 GAR1 B1 0 GROUP 2 GAR2 B2 0 2.0E+4 GROUP 2 GAR3 B3 0 PARM 2 0 AROMATIC.GNA PARM 2 0 NAPT.GNA	These GROUP and PARM commands define the model species, compound list, and parameters for model species used for alkanes and other non-aromatic compounds with similar reactivity characteristics (group 1), and for aromatic hydrocarbons (group 2). Other groups, used for alkenes (group 3) and photoreactive species (group 4), are not shown in this example. The PARM command lists the input file(s) that give names of compounds in the groups and their mechanistic parameters. The file ALKOTH.GNA gives the names and parameters for all alkanes and other compounds represented by model species listed in group 1, and the files AROMATIC.GNA and NAPH.GHA contain these data for the group 2 compounds. Inputs for the other groups are similar. Examples of the contents of .GNA files are given in Example 8b. These examples are discussed further in the text (Section 3.4.5).

Example 8a. Portions of the files OLDNAMES.LPM and LUMPMOLE.LPM that are used for REPLACE inputs in CHAMLUMP.LPC

Representative Input Lines	Discussion
From OLDNAMES.LPM	
C2 ETHANE CARBITOL DGEE	Translation of species names used in older chamber input files to the names currently used for SAPRC-07 detailed model species. Each replacement is for a single name. The model species assigned to ETHANE and DGEE are derived from the GROUP and PARM inputs.
From LUMPMOLE.LPM	
CYC-C7 ME-CYCC6 N-C17 N-C16	The first example gives the assignment of a single compound (methyl cyclohexane in this case) to a name given to a category of compounds. The second gives a "lumped molecule" assignment, where the mechanism and parameters derived for n-hexadecane are also to be used for n-hepadecane.
BR-C9 24-DM-C7 0.500 4-ME-C8 0.250 2-ME-C8 0.250 C9-BEN 135-TMB 0.111 123-TMB 0.111 124-TMB 0.111 N-C3-BEN 0.334 & M-ET-TOL 0.111 O-ET-TOL 0.111 P-ET-TOL 0.111	
	These examples show assignments of mixtures of compounds to names used to define categories of compounds. The first is for branched C_9 alkanes (BR-C9) and the second is for C_9 alkylbenzenes (C9-ARO). The latter requires two input lines because the mixture has more than 4 compounds. Each of these compounds must either have parameters defined in PARM input or be assigned model species in REPLACE input in the lumping control file.

Example 8b. The first portion of the file ALKOTH.GNA that is used to give mechanism parameters for alkanes and compounds lumped with them.

```
! Created from DMSmech7.xls 04/10/13
! File = ALKOTH.GNA
!
PARMS   48
OHrh      OHR2      OHa2      OHRR      OHNR      OHZR      OHa3      OHk3
OHXC      OHk4      OHk6      OHR6      OHto      OHa1      OHme      OHq2
OHco      OHq1      OHmg      OHg1      OHip      OHmv      OHQM      OHA1
OHHO      OHQ1      OHZ1      OHMG      OHZ2      OHBA      OHQ2      OHRH
OHA2      OHK3      OHA3      OHK4      OHTO      OHME      OHK6      OHz1
OHC2      OHz2      OHba      OHz3      OHXN      OHcl      OHca      OHct
!
ETHANE
K(OH)   1.340E-12     0.992      2.00
OHrh    1.0000
OHR2    1.0000
OHa2    1.0000
OHRR    1.0000

PROPANE
K(OH)   1.490E-12     0.173      2.00
OHrh    0.9650
(etc)
```

An example of contents of a mechanistic parameter (.GNA) file that can be specified as input in the PARM input is shown in Example 8b. These are ASCII files where lines with a "!" in the first column are treated as comments and ignored. The first non-comment inputs must be the list of parameters, where the first line is "PARM *n*", where *n* is the number of parameters that can be input, and the subsequent lines are the names of these *n* parameters, 8 per line, input using the FORTRAN format code 8(A6,2X). For the SAPRC07 mechanism, the first two characters of the parameter name refer to the type of reaction and the rest refer to the product or type of parameter. The parameter names output by LUMPINT consist of the prefix specified on the PARM line in the .LPC file, followed by the parameter names used in the .GNA file.

The next set of inputs in the .GNA file give the names and parameters for the compounds in this group. Example 8b shows all the input needed for ethane and the start of the input for propane. The first line is the name of the compound, followed by the rate parameters for the reaction with OH, giving either the rate constant or the Arrhenius A factor in cm^3 $molec^{-1}$ s^{-1} units, the activation energy in kcal/mole, and the "B" parameter in the extended Arrhenius temperature dependence, if used (see Table 6). The A factor and "B" parameters are optional, and if missing only the temperature-independent rate constant is used. The next inputs contain the parameters associated with the OH reaction, each referring to a product or chemical operator species formed in that reaction. If the compound also reacts with other species such as O_3, then input lines giving rate parameters and parameter values for these reactions are also given. Two-character codes are used for each type of reaction supported by the model, with the codes "OH", "O3", "N3" "OP", "Cl", and "HV" being used for OH, O_3, NO_3 and O^3P, Cl, and photolysis reactions, respectively. Example 8b shows only "OH" parameters, but the input for the other types of reaction is similar except for the characters used for the reaction type.

3.5. Processing Simulation Output

Several utility programs and Excel spreadsheet templates are available with the distributed files to summarize, analyze, plot or process results of model simulations using these programs. In all cases, the simulation output data are expected to be in .CDA files contained in folders determined by the model or parameter file employed, as discussed above in Section 3.3.6. Therefore, the simulation should be run using the "SAVE" (not "SAVE CSV") option in the simulation input file. In most cases these programs are also compatible with use of the "SAVE RATE" or "SAVE INTRATE" options that output rates or integrated rates of reactions.

The CDA2CSV and CDA2CSV2 programs can be used to output CSV files containing data in the CDA files in two different formats. The CSV files produced by CDA2CSV are similar to those produced directly by the simulation program using the "SAVE CSV" format, except there is no padding after the names of the species so the names load into spreadsheets without trailing blanks, though the CSV files cannot be read by programs expecting fixed column width output. In addition, species whose values are zero at all times are not output, which can reduce the numbers of columns output if simulations use fewer species than contained in the model. This makes them more generally useful for importing the data into spreadsheets. As with CSV files produced directly by the simulation program, separate columns are used for each species output, and separate lines are used for each time for which data are output. These files may not load completely into spreadsheets if there are more than 254 species whose data are output. The CSV files produced by CDA2CSV2 differ in that data for each time is given in separate columns, with data for each species given in a separate line. This is more suitable for processing output data for models with large numbers of species, or if the "SAVE RATE" or "SAVE INTRATE" options are given in models with large numbers of reactions. However, it is less suitable for processing data for calculations with large numbers of output times.

In both cases, the first record of the output .CSV files gives the calculation and parameter file names and the date and time of the simulation, the second gives the calculation label from the .INT file, the third gives the name of the model, and the fourth gives the number of species whose data are output. Files output by CDA2CSV2 also has a record giving the number of time points output, but that is not given in those output by CDA2CSV. Following these there is a blank record followed by a record giving the column headers. As indicated above the column headers for the files output by CDA2CSV are the species names, while those output by CDA2CSV2 are the simulation times in minutes. For files output by CDA2CSV the subsequent records are the data for each simulation time, with first the time in minutes and the concentrations or values for the species as indicated by the column headers. For files output by CDA2CSV2 the subsequent records are data for each species, with the species name followed by the concentration or values for each of the times indicated in the column headers.

To run CDA2CSV or CDA2CSV2 give the command

CDA2CSV[2] *intfile* [*prmfile*] [*outfile*]

where *intfile* is the name of the .INT input file for the simulation, *prmfile* is the parameter file used when running the simulation (or "CALCS" if no parameter file used), and *outfile* is the name of the CSV file to be produced (without the .CSV extension). For example, the command "CDA2CSV2 RUN1 SAPRC07 RUN1" reads SAPRC07\RUN1.CDA and outputs RUN1.CSV in the format with time data in columns. If *prmfile* is not given it is assumed to be "CALCS". If *outfile* is not given it is assumed to be "TMP", i.e., the output file is TMP.CSV. For CDA2CSV the additional option SAVE=*listfile* can be included as a fourth option on the command line, where *listfile* is a complete file name giving a list of names of species to be output, one species name pre line.

All these programs assume that the root folder for the CDA files are as indicated in the CAL= input line in MODELING.PRM. For example, if MODELING.PRM contains the option "CAL= C:\MODEL" and the command is "CDA2CSV2 RUN1 SAPRC07", the input is taken from the file C:\MODEL\SAPRC07\RUN1.CDA. For this reason, it is not recommended that CAL= options appear in parameter files for individual models (e.g., "SAPRC07.PRM") because model-specific parameter files are not read by these or other data conversion programs. If this option is in MODELING.PRM, it is used by both the model simulation program that produces the CDA files and the CDA2CSV and other programs that read these files, so the latter should find these files where they were output.

Several other programs and some Excel spreadsheet templates are also available for processing simulation results. Documentation for these and other programs and spreadsheet templates useful with this software is in preparation or planned (see Section 4).

3.6. Calculation of Reactivity Scales

The incremental reactivity of an organic compound with respect to ozone formation in the atmosphere is defined as the additional amount of ozone formed when a small amount of the compound is added to the emissions in a "base case" scenario representing atmospheric conditions, divided about the amount added, to the limit of small amounts added. Calculating incremental reactivities requires a minimum of two model simulations: (1) a "base case" simulation representing the conditions of the ambient scenario for which incremental reactivity values are desired, and (2) a "test" simulation representing the same conditions except that a small amount of a selected test compound added to the base case reactive organic (ROG) inputs. This can be done using a single prepared model that has all the model species needed to represent base case conditions plus a model species representing the test explicitly, though in practice separate prepared models are usually used for base case and for test calculations. This is because explicit mechanisms are needed to calculate reactivities of any test compound, and it is generally more efficient to have separate models for explicitly representing different types of VOCs so it is not necessary to have explicit reactions for all compounds of interest in a single model.

This distribution includes all the programs and files needed to calculate the MIR and other ozone reactivity scales that were calculated for the SAPRC-07 mechanism (Carter, 2010a,b,e), using the procedures and scenarios first described by Carter (1994a). These are listed on Table 14. The REACTCAL program, discussed in more detail in Section 3.6.3 below, is the major program used for conducting both base case and test calculations needed for reactivity scales. Use of this program imposes certain requirements for both the prepared model and the simulation inputs, which are applicable regardless of which mechanisms and scenarios are used, and these are discussed first in Sections 3.6.1 and 3.6.2, respectively. Following the discussion of the REACTCAL program and its input and output files in Section 3.6.3, programs available for processing and summarizing reactivity output are described in Section 3.6.4.

3.6.1. Model Preparation for Reactivity Calculations

Calculating incremental reactivity scales requires a minimum of two model simulations: (1) a "base case" simulation representing the conditions of the ambient scenario for which incremental reactivity values are desired, and (2) a "test" simulation representing the same conditions except that a small amount of a selected test compound added to the base case reactive organic (ROG) inputs. This can be done using a single prepared model that has all the model species needed to represent base case conditions plus a model species representing the test explicitly, though in practice separate prepared models are usually used for base case and for test calculations. This is because explicit mechanisms are needed to calculate reactivities of any test compound, and it is generally more efficient to have separate models for explicitly representing different types of VOCs so it is not necessary to have explicit reactions for all compounds of interest in a single model. In addition, use of the REACTCAL program requires additional information from results of base calculations that is not needed in test calculations. Furthermore, if reactivity calculations are needed for many types of VOCs, it is generally more efficient to have separate models for test calculations of different types of VOCs, so it is not necessary to have explicit reactions for all compounds of interest in a single model. The specific requirements for the two types of prepared models for use with REACTCAL are discussed below.

<u>Requirements for Both Base and Test Models.</u> The use of REACTCAL to calculate reactivity scales requires use of the AIRSHED option and use of .GROUP input (see Section 3.2.10), which must define groups named "NOX", "HC" and "THC", in that order. The group called "NOX" should include model species for NO, NO_2, and optionally other N-containing species that may be co-emitted with NO_x, such as HONO. The group called "HC" should include model species representing anthropogenic reactive organic gas (ROG) emissions in the base case scenario. These must be groups 1 and 2, respectively, to be consistent with the distributed model-independent input files used for reactivity simulations. The input files are set up based on separate specifications of total HC and NO_x inputs, with separate input (or .CMP files in the case of HC) being used to define the relative speciation. A third group, called "THC" is used for the test compound in the test models and certain counter species in the base model.

The models also need to define other species and parameters that may be used as inputs for the reactivity scenarios. For example, distributed input files defining the "EKMA"-type include inputs for biogenic species called "ISOP, "APIN", and "UNKN", and also for CO, and uses parameters called E#ISOP, …, E#CO to specify their emissions, regardless of the model used. Therefore, the base and test

Table 14. Files used for Reactivity Calculations using the SAPRC-07 mechanism.

File [a]	Description
	Programs
PGMS folder (must be on DOS path)	
REACTCAL.EXE	Program to conduct reactivity calculations. Prepares scenario input file for a base or added compound simulation, then calls the model simulation program to run the simulation, then processes the incremental reactivity results.
MROPT2.EXE	Program to find NOx levels that yield MIR, MOIR, and EBIR conditions for a given model and base case scenario. Results go into the .BAS file for the scenario.
DMSRCT.EXE	Program to produce an output file summarizing reactivity results in mass units, to serve as input to REACTTAB.
REACTTAB.EXE	Program to collect and tabulate reactivity results for display or importing into spreadsheets.
	Input files
REACT\ folder	
MODELING.PRM	Parameters to use when running reactivity calculations and programs. See Table 19.
SCENARIO.PRM	Input parameters defining the base case scenarios for incremental reactivity calculations. See Section 3.6.3.
LUMPALL.BAT	Batch file to process VOC composition input for the mechanism. Uses the .CMP files input, and produces the processed composition (.RXP or .LMO) files in the mechanism subfolder. Needs only be run if the compositions change. Also used to process mechanism inputs for test VOCs in TESTHC.INS.
REACT\CMPFILES folder (location specified by the CMP= input in MODELING.PRM) [b]	
ARBMIX1.CMP *ALOFT*.CMP	File giving the composition of the base ROG mixture used to represent anthropogenic emissions from all sources (ARBMIX1.CMP) or the composition of VOC pollutants aloft, if applicable (ALOFT.CMP). File contains SAPRC compound ID and ppm compound in 1 ppmC of mixture. Change references to this in LUMPALL.BAT and SCENARIO.PRM if a different mixture is to be used.
REACT*SAPRC07* folder (location specified by RXP= and BASEXT= in MODELING.PRM) [c]	
Note on *SAPRC07* name:	The folder name should identify the mechanism, and currently is included for SAPRC07. Change the name, and the references to it in MODELING.PRM, if a different mechanism is used.
TESTHC.INS	Processed mechanism input for test VOCs for the mechanism.
ARBMIX1.RXP *ALOFT*.LMO	ARBMIX1.CMP or ALOFT.CMP processed for reactivity input. Different extension used because of different format for base ROG and aloft VOC input. Use LUMPALL.BAT to produce a new version of a different aloft composition. Give it the same name as the .CMP file used to produce it, and change references to it in LUMPALL.BAT and SCENARIO.PRM if needed.

Table 14 (continued)

File [a]	Description
REACT\INPFILES folder (location specified by the INP= input in MODELING.PRM)	
ScenType.INP	Input conditions defining the inputs to the individual scenarios that are not included in SCENARIO.PRM. The scenario input file name (*ScenID*) is referenced as the SCENTYPE input in SCENARIO.PRM. Examples of contents of .INP files for reactivity calculations are given in Table 17.
MECH*SAPRC07* folder (mechanism folder -- files may vary with mechanism. Location specified by parameters in MODELING.PRM) [b]	
TESTHC.INP	Test VOC input file listing compounds supported by this mechanism for reactivity calculations and indicating which mechanism file to use when conducting simulations where it is added. This is used as input to LUMPALL.BAT to produce TESTHC.INS in the REACT*SAPRC07* folder.
AIRLMPBS.* AIRLMPG.*	Mechanism model simulation files for base case and added test VOC calculations. The mechanism suffix "BS" or "G" is specified in TESTHC.INS (.INP), to indicate the mechanism used for test simulations.
STDZ640.FZS	Parameters for calculating photolysis rate constants for the photolysis reactions in the mechanism, given the solar zenith angle. See Section 3.3.4. The file name is referenced in the P(Z) input in SCENARIO.PRM, and that input should be changed if a different file is used.
EXPLKOH.PRM	File giving the OH radical rate constant for all test compounds for which reactivities are to be calculated, except for compounds whose rate constants are given in the TESTHC.INS input file if this is a lumped parameter mechanism (as is SAPRC-07). This information is used to compute how much test compound to add in the incremental reactivity simulations.
DMS.PRM	List of SAPRC detailed model species names for all supported compounds, giving their molecular weights and carbon numbers. Used to normalize .CMP files to give ppm compound / ppmC mixture and to derive mass-based incremental reactivities from the initially calculated mole-based values.
BASELUMP.LPC	Lumping control file used for pre-pre-processing ground-level and aloft VOC pollutant composition files for input to reactivity simulations. This assigns compounds to the model species appropriate for them in base case simulations.
TESTLUMP.LPC	Lumping control file for processing test VOC input for reactivity calculations using adjustable parameter mechanisms such as SAPRC07. This is used to prepare the TESTHC.INS giving the list of compound names and other information in the TESTHC.INP file discussed above. This assigns all compounds represented using adjustable parameter model species to a the adjustable model species used in the test model.

Table 14 (continued)

File [a]	Description
Output files	
REACT\CDTFILES*SAPRC07* folder (Location of CDTFILES folder is specified in the CAL= input in MODELING.PRM. The subfolder with the results for this mechanism is has the same name as given for model-specific outputs.) [b,c]	
ScenID.CDA	Concentration-time calculation results of the base case calculation. Same format as used in basic model simulations. This has the data used by REATCAL to derive the data stored in *ScenID*.BAS (see below).
REACTCAL.CDA	Concentration-time calculation results of the last added test VOC simulation that was carried out with this mechanism. This has the data used by REATCAL to derive the reactivity results stored in *ScenID*.CLC and .CLD (see below). This is over-written when the next added VOC calculation is done with this mechanism.
REACT*SAPRC07*/CLCFILES folder	
ScenID.BAS	Results of the base calculation for a scenario. If this is a base case scenario, it is named in SCENARIO.PRM. (Note that *ScenID* is different from *ScenType*. *ScenID* refers to the complete scenario, while *ScenType* refers to the inputs that are given in *ScenType*.INP, which could be used by more than one scenario with different values of parameters given in SCENARIO.PRM. The Scenario ID is reverenced by the user or batch files doing reactivity calculations or retrieving the results. This summarizes inputs and results and also gives NOx levels giving MIR, MOIR, and EBIR conditions after the MROPT2 program is run for the scenario.
ScenID.CLC ScenID.CLD	Unprocessed incremental reactivity results for this scenario. The .CLC file has the ozone reactivity data, and the .CLD file has data that can be used to derive other reactivity metrics or effects on formation of different compounds. The reactivity data are output in molar units.
REACT*SAPRC07*/RCTFILES folder	
ScenID.RCT	Processed reactivity results in mass units. Produced by DMSRCT using *ScenID*.CLC as input and DMS.PRM (referenced by the DMS= input in MODELING.PRM) to get molecular weights.

[a] File names that are specific to the SAPRC-07 scales using the Carter (1994a) scenarios are shown in *italics*. These names can be changed if a different mechanism is used, but corresponding changes would need to be made in modeling.prm and SCENARIO.PRM.

[b] See Table 3 for a list and description of inputs in MODELING.PRM for general use.

[c] Reactivity results for a mechanism are output in a subfolder to REACT which is given the name of the mechanism in the RXP= and BASEXT= input in MODELING.PRM (see Table 19). Some of these data are given in subfolders to this folder that are always named CLCFILES and RCTFILES. Raw calculation results go in a subfolder to the CDTFILES folder with the same name.

models should include these either as model species that can be emitted (included .EMI input for preparation), or define them as parameters and include .INS input (Section 3.2.11) to derive initial concentration and emissions inputs of the corresponding model species actually used for them in the mechanism.

Example of Reactivity Model Preparation Input. Example 9 shows selected portions of distributed model preparation input files for calculation of Carter (1994a, 2010e) incremental reactivity scales. These inputs are discussed further below.

Composition Input for Both Base and Test Models. Model simulations using model-independent inputs defining scenario conditions are normally carried out using a two-step process, where a lumping pre-processing program is used to convert the model-independent inputs to inputs in terms of species and parameters in the model, which are then used to carry out the model simulations. For calculations using INTRUN (Section 3.4.3), both of these steps are carried out at the same time, with the output of the lumping programs that is used as input in the simulation step being deleted if the simulation completes successfully. However, REACTCAL is different in that it expects that the all the lumping processing steps have already been completed prior to any actual reactivity simulations, with the lumping output being saved in files located in the locations indicated using the "RXP=" or "TESTHC=" inputs in MODELING.PRM.

Both the base and test models need input giving compositions of VOC mixtures that are initially present, emitted or present aloft (if applicable) in the base case scenarios. The model species used for these mixtures listed in the .GROUP HC mechanism preparation inputs, and any active species in the model can be represented as being present aloft by specifying an aloft concentration using a A#*species* parameter. The scenario inputs specifying the concentrations of these species in the simulation include the total amounts of VOCs input or present aloft in mole carbon units, and gives the names of composition (.CMP) files giving relative compositions of these inputs. (See Section 3.4.1 for a discussion of .CMP files.) Note that the name of the composition file used for base VOC inputs (represented using GROUP HC) is specified in the BASEHC input, and that used for aloft VOCs is specified in the ALOFTHC input, as indicated on Table 16.

The composition files specified in the BASEHC or ALOFTHC input give the relative concentrations of the compounds in terms of mechanism-independent detailed model species, and these need to be converted into relative concentrations of model species to use in the simulations. REACTCAL expects that the lumping processing for these files have already been carried out prior to any simulations of scenarios using these mixtures, and the results are obtained in files with the same name in the folder indicated by the "RXP=" input in MODELING.PRM. These files must have the extension .RXP and .LMO for ALOFT files. The *BaseHCname*.RXP files is used to specify inputs for base case ground level and emitted VOCs, and contain input lines of the form "R2#*ModSpe Amount*", where *ModSpe* is the name of a model species and *Amount* is the moles of this model species per ppmC of the mixture. The "R2#" prefix indicates that this input is being used to specify relative concentrations of compounds in group HC, which is required to be group 2. The *AloftHC*.LMO is used to specify the moles of model species per mole carbon of VOCs aloft, and contains lines of the form "*ModSpe Amount*" to give the moles model species per mole carbon aloft VOCs. The prefix is not used in this case; REACTCAL prepends the model species name with "A#", and multiplies the moles/moleC values in the .LMO file times the specified total ppmC of VOCs aloft to get the absolute concentrations of the species aloft.

The distributed files for calculating SAPRC reactivity scales include the file REACT\LUMPALL.BAT that has the DOS commands to run LUMPINT to convert the .CMP files that may be needed for the reactivity scenarios for the SAPRC07 mechanism. This includes commands to create REACT\SAPRC07\ARBMIX1.RXP from REACT\CMPFILES\ARBMIX1.CMP and REACT\

Example 9. Selected portions of model preparation input files used for calculation of incremental reactivity scales using the box model scenarios used for the Carter (1994a) reactivity scales.

Input Data [a]	Comments
(Model title) (different for base and test)	One-line comment describing model
AIRSHED TEMP 300.	Indicates "AIRSHED" options used (see Section 3.2.10) and gives default temperature
MAXVCO 660 (base) or MAXVCO 5000 (test)	Maximum number of variable coefficients. Must be larger in the test VOC model because of the use of adjustable parameter mechanisms.
.ACT = O3 + NO + NO2 + HONO = (list of compounds in .GROUP HC) GEN1 (test only)	List of active species. This is not really needed but is useful to control the order of species that are output. "GEN1" is the adjusted parameter model species used in the test (AIRLMPG) model.
.COE INERT A#INERT R2#INERT R3#INERT (test only)	Allows input for an "INERT" model species to be included in the input. They are given only so the program will not abort if these names are used. The first coefficient is used for the initial concentration, the 2nd for aloft concentration, and their others for possible inclusion in .GROUP inputs..
.EMI = NO + NO2 + HONO .GROUP NOX = NO + NO2 + HONO .COE R1#NO 0.8 R1#NO2 0.2 R1#HONO 0.0 INI.NOX 1.0	Specifies NO_x species that can be emitted and defines the "NOX" emissions group and sets the default values for this group's parameters (see Section 3.2.10). Note that the default is for all NOx species to be present initially.
.EMI = (list of model species in group HC) GEN1 (test only) .GROUP HC = (list of model species in group HC) .COE TOT.HC 0.0 INI.HC 1.0	Specifies VOC species that can be emitted and defines the that can be emitted and defines the "HC" emissions group. Note that the default is no HC emissions and all present initially.
.GROUP THC = (list of all explicitly represented VOCs) = NO + NO2 + HONO GEN1 (test only) .COE TOT.THC 1.0 (base) or 0.0 (test) INI.THC 1.0	Specifies test VOC input. Explicit VOCs are listed so their reactivities can be calculated, and GEN1 is the adjustable mechanism species that can be used in test calculations. Note that this is also needed for the base mechanism and nonzero input is the default because it includes reactions of counter species that are input with the same schedule as the test VOCs in the reactivity calculations.

Example 9 (continued)

Input Data [a]	Comments
.UNITS=PPM @BASEMECH.RXN @ATMMECH.RXN @WALLS.RXN	Input giving the reactions and rate parameters in the base mechanism is taken from several files. These give the rate parameters in molecule, cm, sec units so their input follows the .UNITS=PPM command to convert the units to the ppm-minute units used in the simulations.
.COE ISOP 0.0 E#ISOP 0.0 APIN 0.0 E#APIN 0.0 UNKN 0.0 E#UNKN 0.0 .INS INIT "C ISOPRENE" = "C ISOPRENE" + 'ISOP' "C TERP" = "C TERP" + 'UNKN' + 'APIN' .INS DIFF 'E#ISOPRE' = 'E#ISOP' 'E#TERP' = 'E#APIN' + 'E#UNKN'	Code for computing biogenic species based on biogenic inputs in EKMA models. These biogenics are represented by species called "ISOP", "APIN", and "UNKN". The .COE input defines coefficients that need to be defined to accept inputs of their initial and emitted amounts. The .INS INIT code, which is executed at the start of the simulation, computes the initial concentrations of SAPRC model species based on values of these input parameters. The .INS DIFF, which is executed at each time step in the simulation, computes the emissions rates from these parameters. It is executed at each time step to allow emissions to vary with time during the simulation.
(additional input)	Defines input to compute integrated OH, total PANs, and output the instantaneous NO_2 photolysis rate. See comments in the .PRP files. Not needed for the reactivity simulations.
This input is for Base models only	
(This input is for the base model only) @BASETEST.RXN (base only)	Input file to compute concentrations of an inert counter species "TRACE" and counter species with various OH rate constants ("T.3" … "T3C") with the same schedule as the test VOC input. Required for reactivity calculations. One mole of each counter species is input for each mole of TOT.THC.
This input is for Test models only	
(This input is for the test model only) @S07GEN1.RXN @(files for GEN1 products not in base) @(files for adjusted parameter product species of GEN1, if used)	Input files for reactions of GEN1, an adjustable parameter model species that can represent different compounds depending on the parameters. In some cases products can be formed that are not in the base mechanism, so additional files are included to give reactions of such species. In addition, the SAPRC-07 reactivity model allows for optional use of adjustable parameter model species as products of GEN1, and files defining reactions and parameters for these species are included as well.

[a] Taken from input files AILMPBS.PRP for the base model and AIRLMPG.PRP for the added VOC model. Input lines listing multiple species are not fully copied but described in parentheses. Comments in the file (which are similar to those given here) are not shown. Except as noted otherwise, the input is for both AIRLMPBS (base) and AIRLMPG (test).

SAPRC07\ALOFT.LMO from REACT\CMPFILES\ALOFT.CMP using LUMPGEN and the lumping control file MECH\SAPRC07\SAPRC07L.LCC. LUMPGEN rather than LUMPINT is used because the base case model uses only fixed-parameter model species to represent compounds present in both the base case and test simulations, so parameter input is not accepted. The .LCC file has assignments of SAPRC07 model species to all the compounds that may be present in the base case mixtures (see Section 3.4.4). The batch files first creates a temporary input file containing the composition and lumping commands, then runs LUMPGEN to create the output files. The temporary files are created as follows:

Temporary file for TESTHC processing

```
.LUMPPRE R2#
.LUMP
@CMPFILES\BaseHCname.CMP
.
```

Temporary file for ALOFTHC processing

```
.LUMP
@CMPFILES\AloftHCname.CMP
.
```

These temporary files are then processed using the command LUMPGEN *tmpfile outputfile*, where *tmpfile* is the temporary file created as indicated above, and *outputfile* is the .RXP or .LMO output file that are needed by REACTCAL to process simulations with these mixtures.

Note that these .RXP and .LMP files are analogous to the .FZS files used for specification of photolysis rates (Section 2.4.3) in that they are properties of both the mechanism and the scenario. They are part of the scenario in the sense that they specify compositions that are input into the scenario, and are dependent on the mechanism because they use model species assignments specific to the mechanism. The distributed LUMPALL.BAT file shows the commands needed to prepare inputs for scenarios using the ARBMIX1 composition for base VOC mixtures and the ALOFT composition for ambient mixtures, but other scenarios may require other mixtures to be processed. Reactivity simulations using such scenarios cannot be done unless these mixtures have been processed. The commands used in LUMPALL.BAT in the REACT folder can be used as a template to create the additional commands needed to pre-process for reactivity input using other base case mixtures.

Additional Requirements for Base Models. Models for base case reactivity calculations using the REACTCAL need to include calculations of the counter species listed and described in Table 15. The table also indicates how the species concentrations are used by REACTCAL. An example of model preparation input that can define these species is as follows.

```
.ACT
= TRACE + T.3 + T01 + T03 + T10 + T30 + T1C + T3C
.GROUP THC
= TRACE + T.3 + T01 + T03 + T10 + T30 + T1C + T3C
.COE
TOT.THC      1.0
R3#TRACE     1.0
R3#T.3       1.0
…
R3#T3C       1.0
.RXN
label)  0.0           ;TRACE =
label)  CONST 3.0E+2  ;OH + T.3 = OH + T.3PROD
…
label)  CONST 3.0E+5  ;OH + T3C = OH + T3CPROD
```

Note that it is not necessary to give labels for the reactions, and the rate constant units are assumed to be in ppm^{-1} min^{-1} units.

Table 15. List and description of active counter species that need to be included in base case models if REACTCAL is used to calculate reactivity scales.

Species	kOH (ppm^{-1} min^{-1})	Description
TRACE	0	Unreacting species. Used to compute corrections for dilution and varying emissions schedules.
T.3	3 x 10^2	Model species that is consumed by reaction with OH, but which does not cause OH consumption or formation of any reactive products. The OH rate constants for each are given in the 2nd column. Used to estimate how fractions of compounds with given kOH values react in the simulations.
T01	1 x 10^3	
T03	3 x 10^3	
T10	1 x 10^4	
T30	3 x 10^4	
T1C	1 x 10^5	
T3C	3 x 10^5	

Note that including these counter species in the "THC" group means that these counter species are represented in the base case calculations as being initially present or emitted with the same fraction and schedule as the used for the model species used to represent the test compound in the test calculations (see below). The final concentration of TRACE gives the final concentration of the test compound if it did not react, so it provides a measure of the amount of test compound added (if TOT.THC=1) that takes into effect dilution and time-varying emissions if applicable. This is used to calculate the molar incremental reactivities of the test compound as follows:

IR (test VOC) = {[corrected test O_3] - [corrected]} / (TOT.THC in test calc.)
[corrected base or test O_3] = [final base or test O_3) / [final base TRACE]

Note that the incremental reactivity units calculated this way are moles O_3 per mole VOC reacted. Note that the final TRACE level would 1.0 in a static scenario with all reactants present initially, resulting in the incremental reactivity just being the difference in final O_3 calculations divided by the amount added, with no correction needed.

As indicated on Table 15, the "T.3" … "T3C" counter species are used to estimate the amount a VOC reacts in the scenario given its OH rate constant, if it only reacts with OH. This is used to derive the amount of test VOC to add in the test calculation if its OH rate constant, kOH, is known, as follows:

(TOT.THC in test calc.) = MOLRCT * [final base TRACE] / (estimated factor reacted)

MOLRCT is an input parameter to REACTCAL (see Table 16 in Section 3.6.2) that gives a target moles of test compound to add in the reactivity simulations, based on considerations of balancing the desire to calculate incremental reactivities for small amounts added, while having sufficient amount added to minimize numerical error. The estimated factor reacted is derived from its OH rate constant and the final [T*]/[TRACE] ratios in the base calculation, which is input to the REACTCAL program as discussed in Section 3.6.1. For example, if kOH = $3x10^2$ ppm^{-1} min^{-1}, then the estimated reacting would be the final ratio ([TRACE]-[T.3])/[TRACE]. Interpolation is used when kOH is different than the values defined for the reacting tracers. Rapidly reacting compounds are given a code that tells REACTCAL to assume that it is completely reacted in the test calculation, i.e., use 1.0 as the estimated factor reacted.

Additional Requirements for Test Models. The test model does not need the tracer calculations discussed above, but instead uses the "THC" group to specify the model species that may be used for the test calculation. The model preparation input must include a .GROUP THC input naming all the model species that may be used to represent test species, which could include any of the species in the base mechanism. Generally it is a good idea to include all or most of the compounds in group "HC" in group "THC" so incremental reactivity calculations can be done for compounds that are explicitly represented in the base mechanism. However, if the test compound is not part of the base mechanism, then a new model species must be added to represent that compound explicitly, and it should be added to the list of species in the .GROUP THC command. Note that while the default value for TOT.THC should be 1 in the base model, it should be zero in models for test calculations.

If reactivity scales are to be calculated using an adjustable parameter mechanism, than use of a model with a single adjustable parameter model species should be sufficient to explicitly represent compounds that can be represented by this method, assuming that their mechanistic parameters have been derived. The parameter values for the various potential test compounds are not part of the prepared mechanism, but are contained in parameter files used by LUMPINT as described in Section 3.4.5. Although not used in the prepared model, these parameters for the individual compounds must be considered to be a part of the detailed mechanism. These parameter files are used as input in pre-processing the mechanism for input into LUMPINT as discussed below in Section. The mechanistic parameters that are input include the OH rate constant for the compound, which is used to calculate how much to add in the test compound as discussed above.

Specifications of Mechanisms for Test Compounds. When used for test simulations, REACTCAL is given the model-independent detailed model species name of the test compound on the command line. REACTCAL then determines which model species and (if applicable) mechanistic parameters to use for the model species from the file named in the TESTHC= input in MODELING.PRM, which for the distributed files is REACT\SAPRC07\TESTHC.INS. This file gives the model-independent species name, a code indicating which model to use for the test calculation (allowing different test models to have explicit mechanisms for different groups of test compounds), the model species to use for the compound (there may be more than one depending on the model), and mechanistic parameters if the model species is an adjustable parameter species. In practice, this file is not created manually, it is created using a file such as MECH\SAPRC07\TESTSHC.INS as input to a lumping program (LUMPINT for adjustable parameter mechanisms such as SAPRC07, or LUMPGEN for fixed-parameter mechanisms), using an appropriate .LCC or .LPC file to determine the lumping, as discussed below. However, TESTHC.INS files can be created manually, and are relatively straightforward for explicitly represented species.

The TESTHC.INS file is an ASCII file containing a series of a minimum of three input lines for each detailed model species whose reactivity may be calculated. Comments lines, which are also allowed and are ignored by the program, are indicated by a "!" in the first column. The first input lines for each compound contains

##*DMSname*
ModelCode
R3#*ModSpe* 1.0

Where *DMSname* is the model-independent detailed model species name that might be specified on a REACTCAL command line and used in files summarizing the results, *ModelCode* is a code indicating the model to be used in the test calculations, and *ModSpe* is the name of the model species used to represent it in the test model. The "##" in the first line is used to terminate input for the previous compound as well as the start of the input for the next one. The test model that REACTCAL calls to do the test simulation is obtained from the "MOD-*ModelCode*=" input in MODELING.PRM, and allows different model programs to be used depending on the compound. The rest of the input lines is the input to be inserted into the model simulation input file to represent the test compound. As discussed above, model species to be used for test compounds must be specified in the model preparation input as part of GROUP THC, which is expected to be the third group defined after the groups for NOX and HC, with a default relative contribution, or R3#*species* value, of zero. The "R3#*ModSpe* 1.0" input says that the model species used for this compound, *ModSpe*, makes a 100% contribution to this group. This is all the information needed if the model species *ModSpe* represents the compound *DMSname* explicitly.

Additional input is needed if the compound is being represented in the model by an adjusted parameter species, whose rate constants and product yield and other mechanistic parameters can be varied depend on the compound it represents. In the SAPRC07, the adjusted parameter model species used for test compounds is named "GEN1", the model code used is "G", and MODELING.PRM includes the input "MOD-G=AIRLMPG", indicating that the test model to be used for such compounds is AIRLMPG. The TESTHC.INS is therefore as follows:.

DMSname
G
R3#GEN1 1.0
K(G1OH) *kOH, AOH, EOH*
G1OH*p1 parm1 (for OH reaction)*
…
K(G1O3) *kO3, AO3, EO3*
G1O3*p1 parm1 (for* O_3 *reaction)*
…

Here, *kOH, AOH,* and *EOH* are the rate constants, Arrhenius A factors, and activation energies for the OH reaction in ppm^{-1} min^{-1} or kcal/mole units (and likewise for *kO3*, etc.), *p1*, etc are parameter codes and *parm1*, etc. are parameter values for the various reactions. These match the codes given for the adjustable parameter model species in the preparation input files, e.g., MECH\SAPRC07\LMPGEN1.RXN.

The most convenient way to prepare TESTHC.INS files for adjusted parameter mechanisms is to use LUMPGEN to output this file using using as input a file such as MECH\SAPRC07\TESTHC.INP that gives the input for the compounds involved, processed using an appropriate lumping control file for this purpose. The batch file LUMPALL.BAT in the REACT folder has the lumping commands to prepare TESTHC.INS for SAPRC07, as well as input files for the base case VOC mixtures as discussed in the next section. The relevant command is

```
LUMPINT ..\MECH\SAPRC07\TESTHC.INP SAPRC07\TESTHC.INS ..\MECH\SAPRC07\TESTLUMP
```

where the input is TESTHC.INP, the lumping control file is TESTLUMP.LPC, both in the mechanism folder, and the output is TESTHC.INS in the REACT\SAPRC07 folder. The input data in TESTHC.INP is similar to that in TESTHC.INS in that it has groups of inputs for each compound beginning with inputs starting with "##*DMSname*", where *DMSname* is the name of the compound, and can also have comments starting with "!". The inputs can optionally be exactly the same as used in TESTHC.INS, in which case they are just copied as is into TESTHC.INP. This can be used for special "pure mechanism" test species that may be useful for some applications, with the most useful being the test species "NULL-3", whose reactions should have no effect on model predictions and are useful to verify that the test model and base model would give the same results if such a species is added. But for most compounds, the input in TESTHC.INS is as follows:

DMSname
G *(or* STD *if this is known to be represented explicitly)*
.LUMPPRE R3#
.LUMP
DMSname 1.0
.

The lines starting at .LUMPPRE are treated as input to LUMPINT (or LUMPGEN if applicable -- depending on input in MODELING.PRM) and are converted to the output needed in TESTHC.INS, which is in the format indicated above.

The lumping control file used to create the TESTHC .INS file from the .INP file would have the same assignment input for explicitly represented compound as lumping control file used for other simulation input, as discussed in Section 3.4. However, for adjusted parameter mechanisms processed using .LPC files, the lumping control file should have only one GROUP input line, which for SAPRC07 is "GROUP 1 GEN1 G1". Then the .GNA files containing the compound names and SAPRC07 adjustable mechanism parameters specified using various "PARM 1 0 *filename*.GNA" input lines. This type of .LPC file would not usually be suitable for processing general input, since it lumps all compounds with adjustable parameter mechanisms into a single model species. The SAPRC07 files include TESTLUMP.LPC for processing test compound input, which is different from the BASELUMP.LPC file used for processing VOC input for base case and general modeling applications, which assigns compounds to their recommended model species for ambient simulations and is used for pre-processing base VOC input (see above).

If REACTCAL is to be used to calculate reactivities for compounds that are not represented using adjustable parameter model species, as would be the case for all species using adjustable parameter mechanisms and explicitly represented species in any mechanism, then it also needs to obtain information about how rapidly the model species reacts in order to determine how much to add in the test calculations. This is not needed for compounds represented using adjustable parameter mechanisms, since the rate constant information can be obtained from the TESTHC input as discussed above. The rate constant information for explicitly represented species obtained from a file named in the "EXPLICIT-KOH=" input in MODELING.PRM. In the cases of the distributed file for the SAPRC07 mechanism, this is MECH\SAPRC07\EXPLKOH.PRM. This is a plain text file with one record for each model species, or comments indicated by a "!" or a blank in the first column. The data columns contain the model species name, OH rate constant in ppm^{-1} min^{-1} units (or the rate constant units used in the model, if different), and a reactivity factor, using FORTRAN format code (A8,E10.3,F8.2,F8.0). The OH rate constant (kOH) input is used to estimate how much the compound reacts, based on the base case results for the "T." tracers as discussed above. If the reactivity factor is positive, the rate of reaction estimated using the OH rate constant is increased by this factor, as may be appropriate for compounds that react in other ways besides with OH to a non-negligible extent. It the factor is -1.0, it means that the model species should be treated as completely reacting in the scenario for the purpose of estimating the amount added, and the kOH input is ignored. This is appropriate for highly photoreactive model species or model species used for highly reactive compounds such as C_{3+} alkenes.

3.6.2. Model Scenarios for Reactivity Calculations

SCENARIO.PRM input. The inputs defining the model scenarios for reactivity calculations are given in the file SCENARIO.PRM, which must be in the local folder where REACTICAL is run. This is a plain text file containing options for running REACTCAL and model-independent input defining base case scenarios for reactivity simulations. These options and inputs are listed on Table 16, which also shows parameter values used for the reactivity scenarios of Carter (1994a, 2010e) as examples. These include tot total NOX and HC levels (values of the TOT.NOX and TOT.HC scenario inputs), the composition of the base ROG mixture used to represent anthropogenic VOC emissions, the composition of the aloft VOC mixture, the aloft O_3 and NO_2 and total VOCs, the file containing the photolysis rate constants, and the file containing all the other inputs for this type of scenario.

The REACTCAL options and optionally default scenario input parameters are given in single lines in the SCENARIO.PRM file, each containing the parameter name, one or more space, and the value

Table 16. List and description of scenario input parameters that are given in SCENARIO.PRM, and example parameter values. The example parameter values shown are representative those used to calculate the Carter (1994a, 2010e) reactivity scales.

Parameter [a]	Description
	The following parameters had the same values for all the Carter (1994a) scenarios.
	Program control parameters
O3	Tells REACTCAL to calculate reactivities relative to ozone
MOLRCT 0.0005	This is used to derive the amount of test compound to the simulation, given information about its reaction rate in EXPLKOH.PRM or TESTHC.INS. This is the target estimated amount of VOC reacting in the in units of millimoles m^{-2} per simulation. This was derived by determining the minimum amount of test compound to add in incremental reactivity simulation that is sufficiently large for numerical error to be minor but sufficiently small so incremental reactivities are not dependent on the amount added.
EPS 1.E-6	This is the error control parameter for the Gear solver when conducting the simulations, which REACTCAL passes to the model simulation program. Smaller values make the simulations more accurate but take more computer time. This was used in conjunction with MOLRCT to minimize effects of numerical errors without unnecessary increases in simulation time.
	Scenario conditions parameters
BASEHC *ARBMIX1*	This is used to specify the relative composition of the mixture used to represent anthropogenic VOC emissions, whose total amount is specified by TOT.HC. The Carter (1994a) scenarios use the mixture whose composition in terms of mole compound per mole carbon mixture is given in ARBMIX1.CMP, as indicated in Table 14. Change *ARBMIX1* to another mixture to use a different base VOC speciation.
ALOFTHC *ALOFT* ALO.HC 0.030	These specify the relative speciation and total concentration of VOCs aloft, excluding biogenic model species and CO. The Carter (1994a) scenarios use the mixture whose composition in terms of mole compound per mole carbon mixture is given in ALOFT.CMP, as indicated in Table 14. Change *ALOFT* to another mixture to use a different aloft VOC speciation. ALO.HC is the total VOC concentration aloft in ppmC units. The aloft concentrations of individual VOCs are given by ALO.HC multiplied by the relative concentration in the ALOFTHC .CMP file (ALOFT.CMP in this case).
HV 1.0	Allows the total light intensity to be varied. All photolysis rates are multiplied by this factor.
P(Z) *STD640*	Name of file giving parameters needed to calculate all photolysis rates in the mechanism as a function of the solar zenith angle. See Section 2.4.3. A file with this name and extension .FZS must be in the mechanism input folder where the mechanism input files are located.
ALO.NOX 0.0	Total NOx present aloft in ppm. It is assumed to be in the form of NO2.

Table 16 (continued)

Parameter [a]	Description
The following parameters had varying values for the Carter (1994a) scenarios. The values shown are for the "averaged conditions" scenario.	
SCENTYPE *AVGEPA*	The name of the *ScenType*.INP file containing the scenario conditions, as described in Table 17. The file is located in the folder specified by the INP= input in MODELING.PRM. There is a separate .INP file for each of the 39 city-specific scenarios, with names like ATLGA1.INP, AUSTX1.INP, etc.
E.HC 15.38 E.NOX 2.338	Total anthropogenic VOC and NOX input (initial + emitted) in millimoles C or N m^{-2}. See discussion of specification of reactant groups inputs in Section 3.2.10. Note that E.HC/E.NOX is the VOC/NO_x ratio, in units of moles carbon per mole nitrogen.
ALO.O3 0.0704	Ozone concentration aloft in ppm, which varies with scenario.
Specifications for individual scenarios.	
.SCEN [*parm1*] [*parm2*] … *ScenID* [*value1*] [*value2*] … .	Defines a scenario and optionally gives parameter values for the scenario. The other scenario parameter values used are the last values given previously in the file as a single input line. The first line optionally lists parameter values such as "SCENTYPE", "E.HC", "E.NOX", etc that can be specified in the following input lines. The following lines first give a scenario name (*ScenID*) and then the parameter values that are applicable only to that scenario. Multiple scenarios can be input this way, terminated by an input line containing only ".".

[a] The parameter values given are those used for the reactivity calculations for the Carter (1994a) scenarios. Other parameters may be used for other types of scenarios, but a complete discussion of this is beyond the scope of this document.

(if applicable). Lines with a "!" in the first column are treated as comments and are ignored. The scenarios themselves are defined by inputs starting with ".SCEN" that give the scenario names and optionally parameter values for one or more scenarios. If a scenario input parameter value is not given on the .SCEN line, the default value input using the .SCEN input with the scenario name is used. Note that subsequent input in SCENARIO.PRM can change default scenario input parameters for scenarios defined using subsequent .SCEN inputs.

Scenario type .INP file. One of the inputs that must be defined for each scenario in SCENARIO.PRM is the SCENTYPE parameter, which gives the name of an .INP file containing all the simulation inputs needed for the scenarios except for those given in SCENARIO.PRM. This includes, for example, the start and end times, parameters for calculating photolysis rates, temperature, humidity, background pollutants other than aloft O_3, HC, or NO_x, emissions and inversion height schedules, and other parameters needed for the simulations. These files are located in the folder named in the INP= input in MODELING.PRM (Table 3). Note that a given .INP file can be used for more than one named scenario if different values of the parameters specified in SCENARIO.PRM (e.g., total HC or NO_x inputs) are assigned to them. An example of such an input file is shown on Table 17, which contains the inputs used to define the "averaged conditions" scenario of Carter (1994a) The inputs for the other Carter (1994a) reactivity scenarios are similar. Not all of the input listed in Table 17 is necessary for very simple box model scenarios where all the reactants are present initially and there are no changes in the mixing height.

Table 17. List and description of inputs in the .INP files used to describe the city-specific or averaged conditions scenarios used for the Carter (1994a) reactivity scales.

Input	Description
	Calculation control parameters
NOOUT	Do not produce an output (.OUT) file unless there is an error.
SUN	Use solar photolysis. Photolysis rates are derived based on inputs in SCENARIO.PRM
SAVE	Save calculation results in format expected by REACTCAL
	Calculation conditions input
@MODDOC.IN	A "@" as the first character means take input from another file, in this case MODDOC.IN. This is created by REACTCAL to give a label for the calculation to go with the output file.
.LAT 36.22 .DEC 16.50	Latitude and declination to use when calculating the zenith angle as a function of time for the purpose of calculating solar photolysis rates.
TSTART -75.81.	Difference in minutes between simulation time and the solar time used to calculate zenith angles. -60 minutes is appropriate for daylight savings time.
.DPRN 60.	Time interval in minutes for data to be output. REACTCAL expects this to be 60 minutes.
.T0 480. .TEND 1200.	Start and stop time for the simulation in minutes after midnight. This simulation starts at 8:00AM local time and ends at 8:00PM.
CH4 1.790 CO 2.028 ISOP 0.00010 APIN 0.00010 UNKN 0.00010	Initial ground level pollutant levels. Note that biogenic emissions are represented in EKMA models by isoprene (ISOP), alpha-pinene (APIN) and unknown biogenic (UNKN) model species. In SAPRC-07 the lumped terpene model species TERP is used to represent APIN and UNKN. These do not include NO_x or initial or emitted anthropogenic VOCs.
A#CO 0.50	CO concentrations aloft. Other aloft pollutant levels are specified in SCENARIO.PRM, as discussed in Section 3.6.3.
INI.HC 0.6036 INI.THC 0.6036 INI.NOX 0.4567	The fraction of anthropogenic VOCs, the test VOC for reactivity calculations, and the NOx that is present initially. See discussion of specifications of pollutant concentrations in Section 3.2.10.
EMIT.FAC 24.6268	Conversion factor to convert flux emissions to ppm. See Section 3.2.10. Note that this is based on the expectation that total anthropogenic VOC or NOx inputs are given in units of millimoles m^{-2} units and inversion heights are in meters. See Section 3.2.10,
@ALOFT.IN @TOTHC.IN @BASEHC.IN @TESTHC.IN @NOX.IN	Aloft, total anthropogenic VOC, anthropogenic VOC composition, test VOC, and NOx pollutant inputs as specified in SCENARIO.PRM. These files are created by REACTCAL when the simulation is run. The specification of these inputs using separate files allows this .INP file to be used to represent scenarios with a variety of pollutant levels with all other conditions the same. See Section 3.6.3.

Table 17 (continued)

Input	Description
.VARYCO	The remainder of this table lists variable coefficient input, giving the name of the coefficients and codes indicating how values are derived for the times not specified. These are followed by times (in minutes after minute local time) and the values of the coefficients at that time. The prefix "L#" means that the values between the times given are determined by linear interpolation, and the "C#" prefix means that the value for the previous time is used until the next time is input. Note that these could be replaced by simply the coefficient name and value (outside of the .VARYCO input region) if these parameters are constant.
L#TEMPR L#H2O L#HEIGHT	These inputs give temperature in degrees K, H2O concentration in ppm and the inversion height in meters as a function of time. These also could be constant, but if the inversion height is constant there is no entrainment of pollutants from aloft.
C#EMR.HC C#EMR.THC C#EMR.NOX	These inputs give the relative emissions schedules for anthropogenic VOCs, the test VOC for reactivity calculations, and NOx inputs. These need to be properly normalized to be consistent with the corresponding .INI parameter. These emissions to change each hour but be constant between the hours, based on the treatment in the original EKMA scenarios. See discussion of specifications of pollutant emissions in Section 3.2.10.
C#E#CO C#E#ISOP C#E#APIN C#E#UNKN	The emissions rates of CO and the three model species used to represent biogenic pollutants as a function of time. These are given in units of millimoles m^{-2}, and are represented as changing hourly to constant values in the original input files used by Carter (1994b). Note that these pollutants are treated separately from anthropogenic VOC emissions in the "HC" group, as discussed in Section 3.6.1.
L#R1#NO L#R1#NO2 L#R1#HONO	The parameters R1#*NOxSpecies* give the speciation of the NOx emissions and must sum up to 1. In the EKMA scenarios developed by Carter (1994a) it is assumed that the initial NO_x consist of larger fractions of HONO and NO2 than are emitted, so these are treated as time-dependent parameters that change from their initial values after the first minute of the simulation. The speciation is treated as time-varying to allow for higher initial HONO fractions in the NO_x input than in emitted NO_x.
#END	

At a minimum the lines "@TOTHC.IN" @BASEHC.IN, @TESTHC.IN and "@NOX.IN" must be included in the .INP file so REACTACAL can pass the total HC, VOC composition, test compound input, and total NO_x levels to the simulation program. It does this by creating files of these names based on inputs obtained from SCENARIO.PRM. If there is entrainment from aloft, then the line "@ALOFT.IN is also needed to provide for input concerning aloft levels and compositions. The "SAVE" option also needs to be specified and the "SUN" option is needed if solar photolysis rates are to be input. The other inputs are the same as for any simulation input file for box model simulation.

Photolysis Rate Input. An additional requirement for scenario input preparation for ambient simulations is calculating the photolysis rates in the mechanism for the set of actinic fluxes that are to be used for the reactivity scenario. This is described in Section 3.2.8. Be sure to include in the .FZS input file parameters for photolysis files that may be used for any of the test compounds as well as those used in the base mechanism.

Base VOC Composition Input. Another additional requirement scenario input preparation that is required for REACTCAL simulations is to carry out the lumping processing for the base and aloft VOC inputs as discussed above in the subsection titled "Composition Input for Both Base and Test Models". This needs to be done for each mixture specified in BASEHC or ALOFTHC inputs for all scenarios in SCENARIO.PRM, or at least for those scenarios where calculations are to be carried out. However, this needs only be done once for each mixture, as long as the composition or the lumping assignments in the base model are not changed.

Distributed Reactivity Scenarios. Although many types of scenarios could be used for incremental reactivity calculations, the scenarios distributed with this software are those used for the reactivity scales of Carter (1994a), updated as described by Carter (2000), and used to derive the MIR and other scales given by Carter (2010a,e). These were derived based on am extensive set of idealized pollution scenarios have been developed for a number of urban areas for conducting EKMA model analyses of effects of ROG and NO_x controls on ozone formation (Carter, 1994a and references therein). The EKMA modeling approach involves use of single-cell box models to simulate ozone formation in one day episodes. Such models cannot represent realistic pollution episodes in great detail, and do not incorporate processes such as carry-over of pollutants from one day to the next. However, they can represent dynamic injection of pollutants, time-varying changes of inversion heights with entrainment of pollutants from aloft as the inversion height increases throughout the day, and time-varying photolysis rates, temperatures, and humidities. Thus they can represent a wide range of chemical conditions that may affect predictions of effects of ROG and NOxcontrol on ozone formation. These chemical conditions are the same as those affecting VOC reactivity. Therefore, at least to the extent they are suitable for their intended purpose, an appropriate set of EKMA scenarios should also be suitable for assessing methods to develop reactivity scales encompassing a wide range of conditions.

3.6.3. Use of the REACTCAL and MROPT2 Programs for Reactivity Calculations

The REACTCAL program provides a convenient means to carry out both base case and test compound reactivity simulations, including deriving scales for adjusted NO_x conditions for MIR (maximum incremental reactivity) and other adjusted NO_x reactivity scale. The use of this program, and the files it uses and creates, are discussed in this section. The inputs for adjusted NO_x scenarios are derived using the associated MROPT2 program, which is also described in this section.

Program Options in MODELING.PRM. REACTCAL reads the parameter file MODELING.PRM to obtain file locations and other parameters that control operation of the program. Unlike INTRUN, there is no command-line option to specify use of a different parameter file, so generally the MODELING.PRM file in the folder where REACTCAL is run contains information for the specific set of base and test models to use. Table 3 in Section 3.1 lists the specific inputs used by REACTCAL, of which a number are required for REACTCAL to run. Note that in addition to the parameters used only for reactivity calculation as shown in the last section of the table, it also requires the parameters needed by INTRUN and the simulation programs, such as the location of the prepared model file.

The distributed files assume that REACTCAL and the other reactivity processing programs (other than model preparation) are run in the REACT folder, so it is the MODELING.PRM in that folder that controls reactivity simulations using these files. It is set up for simulations using the SAPRC-07 mechanism as distributed in MECH\SAPRC07. The inputs in that file, which shows where distributed files are located for SAPRC07 reactivity calculations as well as the other REACTCAL parameters used, are listed in Example 10.

Command Line Options. For base case calculations the user specifies the just the scenario ID and recognized options for the base case scenario on the command line, and obtains information about the

Example 10. Examples of a parameter file for reactivity simulations using the SAPRC-07 mechanism.

File=REACT\MODELING.PRM
Parameters for chamber simulations using the SAPRC07 mechanism
(See comments in the distributed file for descriptions of specific inputs)

```
MDC=SAPRC07
LPC=..\MECH\SAPRC07\BASELUMP
LCC=..\MECH\SAPRC07\SAPRC07L
MECH=..\MECH\SAPRC07\
RCTLPC=..\MECH\SAPRC07\REACT
MOD=..\MECH\SAPRC07\
EXPLICIT-KOH=..\MECH\SAPRC07\EXPLKOH.PRM
MOD-BS=AIRLMPBS
MOD-S=AIRLMPG
MOD-G=AIRLMPG
MOD-CL=AIRLMPCL
MOD-SP=AIRLMPSP
MOD-MX=AIRLMPMX
DFLMOD=AIRLMPG
MODPRE=AIRLMP
DMS=..\MECH\SAPRC07\DMS.PRM
SCENPARM=SCENARIO.PRM
RXP=.\SAPRC07\
TESTHC=.\SAPRC07\TESTHC.INS
INP=.\INPFILES\
CMP=.\CMPFILES\
EMI=.\CMPFILES\
CAL=.\CDTFILES\
CLC=.\SAPRC07\CLCFILES\
RCT=.\SAPRC07\RCTFILES\
BASEXT=SAPRC07
MAXCAL=O3 PAN HO2H
FAC=IntOH 1.E+6
SDR=..\CHAMCALC\CHAR\LIGHT\
```

inputs for the base case from SCENARIO.PRM and files reference in it, as discussed below. Optional parameters allows use of inputs from a previously run base case scenario with different values of total NOX and HC inputs, with the results in a new base case scenario. For test calculations, the command line gives at a minimum the name of a scenario that has been previously simulated in a base case calculation and the name of the compound whose reactivity is to be calculated. Other options can be specified to modify the amount added, and whether a previously added VOC calculation is to be re-run. The program then prepares the simulation input file for the base or test calculation, runs the calculation, processes the results, and puts them in output files as discussed below.

The command lines and options for running REACTCAL for both base case and test simulations are summarized in Table 18. The first parameter on the command line is always the scenario ID, and the second parameter on the line is either the name of a test compound or a recognized base case specification option. Note that none of the options specify the chemical mechanism used, since parameters specifying the mechanism are taken from the MODELING.PRM file in the local folder where REACTCAL is run. Therefore, MODELING.PRM would need to be changed if a different mechanism is to be used for reactivity calculations, or REACTCAL can be run in a different folder with a different MODELING.PRM file. The parameters in MODELING.PRM that are read by REACTCAL and other reactivity processing programs are summarized in Table 19.

Table 18. Summary of command line options when running REACTCAL

Parameter	Description
	Base Case Simulations
	REACTCAL *ScenID* [option] [*ScenID2*]
ScenID	Required. Scenario ID for this base case calculation. Must be defined in SCENARIO.PRM unless an option is specified that gets the inputs from another reactivity scenario.
NOx-MIR NOx-MOR NOx-LIM	Optional. Get the total NO$_x$ input for the specified type of adjusted NO$_x$ scenario using the rest of the inputs from a previously calculated scenario specified as *ScenID2*. The base case calculations and then MROPT2 should be run for the scenario named *ScenID2* first. The scenario types are MIR, MOIR (MOR) and EBIR (LIM), referring to specified conditions of NO$_x$ availability as discussed by Carter (1994a).
N=*amount* C=*amount*	Optional. Specifies the total NO$_x$ (N=) or HC (C=) input, where *amount* is the TOT.NOX or TOT.HC value to use in the simulation, which should be the same units as used in SCENARIO.PRM (generally millimoles N or C per meter^2 per simulation). The other scenario conditions are as specified for *ScenID2*. It is necessary to run the base case calculation for *ScenID2* first, but it is not necessary to run MROPT2.
ScenID2	Need if an option specified. A scenario where the base case calculation has already been run. Only needed if "NOx-*type*", "N=", or "C=" are specified as options. Note that MROPT2 should also have also been run for *ScenID2* if "NOx-*type*" is used.
	Test Simulations
	REACTCAL *ScenID compound* [option] [*amount*] [%]
ScenID	Required. Scenario ID for this test calculation. Base calculation must be run first.
Compound	Required. The name of a compound for which the reactivity is to be calculated. The parameters or model species used to represent the compound are given in TESTHC.INS as discussed in the text. Information about the compound must be included in DMS.PRM or DMSRCT will fail (see Section 3.4.1).
NOCHECK	Optional. By default, REACTCAL checks the *ScenID*.CLC file to see if the test compound has already been done since the last base calculation, and if no it does not do the calculation. This over-rides this option, and is appropriate if the model for the test compound was changed since the last calculation.
F=*file*	Optional for compounds represented using an adjustable parameter model species. Takes the parameters for the test compound from a single-compound mechanism parameter file rather than the .LPC file. The full path name of the file, including the extension, must be given. See Section 3.4.6 for a discussion of the format of mechanism parameter files.
ADDED *Amount* [%] ("SUBST" no longer supported)	Optional. Specifies the amount of test compound to add, with the amount given by the *amount* parameter on the command line. If a "%" is specified after the amount (following a space), the amount added is given by 100 * *amount* * E.HC in the base case simulation. Otherwise, *amount* is the absolute amount added in the units used for E.HC in SCENARIO.PRM. Note that *amount* given on this line is moles, not moles carbon. The results do not go to the .CLC file. Instead, the ozone results are appended to a file in the local folder called SUBSTDAT.CSV. This is created if it does not already exist.

Table 19. Parameters in the MODELING.PRM file used to control reactivity calculations.

Code	Description, discussion, and defaults	Programs [a]
	Parameters used for reactivity calculations [a]	
MOD-BS= MOD-G= MOD-*XX*=	Specifies which mechanism executable file to use when conducting various types of reactivity simulations. Parameter value is the first part of the file name, without the extension (8-character max). MOD-BS refers to the mechanism file when conducting the base case simulations, and should be AIRLMPSB for SAPRC-07 reactivity calculations. The others refer to codes given for the test compound in the TESTHC.INS file (see Section 3.6.1).	REACTCAL
SCENPARM=	Name of the SCENARIO.PRM file which gives file names and various input parameters for the reactivity calculation scenarios. SCENPARM =SCENARIO.PRM. in this distribution.	REACTCAL
RXP= BASEXT=	Location of folder where reactivity results and processed input files for the current mechanism are saved. Normally both of these parameters are given the same value, which is the name of the mechanism, which is "SAPRC07" for SAPRC-07 calculations.	REACTCAL
TESTHC=	Location of the file giving the processed input defining mechanisms for the test compounds whose reactivities are to be calculated. For SAPRC07 this is .\SAPRC07\TESTHC.INS.	REACTCAL
EXPLICIT-KOH=	Location of a file giving the OH radical rate constant for all test compounds for which reactivities are to be calculated, except for compounds whose rate constants are given in the TESTHC input file (e.g., SAPRC07\TESTHC.INS) if this is a lumped parameter mechanism (as is SAPRC-07). Must be given for all compounds if this is not a lumped parameter mechanism. This information is used to compute how much test compound to add in the incremental reactivity simulations.	REACTCAL (not needed for VOCs represented using adjusted parameter model species)
CLC=	Location of folder for primary reactivity output. For SAPRC07 this is .\SAPRC07\CLCFILES.	REACTCAL DMSRCT MROPT2
RCT=	Location of folder for processed reactivity output. For SAPRC07 this is .\SAPRC07\CRCFILES.	DMSRCT
DMS=	Location of DMS.PRM file that has molecular weights for computing mass-based reactivity values (see Section3.4.1).	DMSRCT
RCTLPC=	Used to derive reactivities using "pure mechanism" parameter species reactivities. No longer supported [b].	DMSRCT

[a] The examples of parameter values those used for calculating reactivity scales for the SAPRC-07 mechanism, for which the necessary files are all included in the distribution. Parameter values for other mechanisms would be analogous or similar.

[b] If DMSRCT fails with output saying it can't open a blank .LPC file, then get a new version of the programs from the SAPRC file distribution. See Section 2.2.1. Alternatively, add the RCTLPC=*file* input to MODELING.PRM, where *file* is any .LPC file in the mechanism folder.

Normally, the model species and (if applicable) mechanism parameters for the test compound is derived from input in the lumping control files as discussed in Section 3.6.1. However, if the compound is represented in the model by an adjustable parameter model species, the test compound input can optionally be obtained from a single-compound mechanism parameter file, giving the name of the compound and its mechanistic parameters. This is indicated using the command line option "F=*filename*", where *filename* is the full path name (including extension) of a single compound parameter file, which by convention is given the extension .GN1. This file has the same format as multi-compound mechanism parameter (.GNA) files that are referenced in lumping control files used by LUMPINT, as discussed in Section 3.4.5. The only differences are (1) parameters for only a single compound can be input, (2) the input record with the compound name can optionally have additional information that is included in the REACTCAL output files, and (3) the carbon number and molecular weight of the compound can optionally be given as values for parameters called "NC" and "MW", respectively. These carbon number and molecular weight parameters are not included in the parameter output that is used for the simulation program, but if included the REACTCAL will output to the user console the incremental reactivity results in various units. The use of this method means that the model species assignments for this compound do not need to be in the TESTHC input file, though the compound still needs to be listed in DMS.PRM if its reactivity is to be processed using DMSRCT as discussed in Section 3.6.4. Its primary utility is obtaining reactivity estimates for new compounds, or effects of using different parameters for compounds, without having to update or edit the TESTHC input file.

Use of the MROPT2 Program. Many of the reactivity scenarios used in the Carter (1994a, 2010e) reactivity scales are "adjusted NO_x" scenarios, where NO_x inputs are adjusted to yield MIR, MOIR, or EBIR reactivity characteristics. As discussed by Carter (1994a) a MIR (maximum incremental reactivity) scenario represents NO_x inputs that yield the highest incremental reactivity of the base ROG mixture with all other inputs constant, a MOIR (maximum ozone incremental reactivity) scenario represents NO_x conditions that yield maximum ozone concentrations, and an EBIR (equal benefit incremental reactivity) scenario represents NO_x conditions where equal relative changes to total VOC or NO_x inputs have the same effect on maximum ozone yields. In this context, a "base case" scenario represents NO_x conditions that are initially estimated or established for the scenario, based on ambient data or other scenarios. Note that in this context the term "base case" refers to a type of a reactivity scenario, as opposed to a "base case" reactivity simulation, which could use adjusted NO_x as well as base case types of scenarios.

The MROPT2 program is used to find the MIR, MOIR, and EBIR NO_x levels of any scenario whose inputs are defined in SCENARIO.PRM. It is called using the command

MROPT2 *ScenID* [LOG=*LogFile*]

Where *ScenID* is the name of a scenario for which the base calculation has already been run, and *LogFile* is the name of a .LOG file that can optionally be output to summarize the process. The primary output is added to the *ScenID*.BAS file for the scenario, that is created during the base calculation (see discussion of output files, below). This includes NO_x inputs and resulting maximum O_3 concentrations and molar incremental reactivities of the NO_x and base ROG mixture at various NO_x levels, including those labeled "NOx-MIR", "NOx-MOR" and "NOx-LIM" for MIR, MOIR, and EBIR NO_x levels, respectively. Note that these values in the .BAS file are over-written if MROPT2 is run again for this scenario.

After this is run, then the adjusted NO_x scenarios can be defined and used in base case simulations using the commands

REACTCAL *AdjScenID* NOx-*type* *BaseScen*

Where *AdjScenID* is the name given to the adjusted NO_x scenario, *type* is the type of NO_x adjustment (MIR, MOR, or LIM for MIR, MOIR, and EBIR scenarios, respectively), and *BaseScen* is a scenario defined in SCENARIO.PRM for which a base case calculation has already been run. Note that the inputs for the adjusted NO_x scenarios are NOT in scenario.prm, because all their inputs taken from the *BaseScen*.INP file. Once this simulation is successfully run, then added test compound simulations can be conducted using this adjusted NO_x scenario.

REACTCAL Output Files. The output files produced by REACTCAL are summarized in Table 14. These include .CDA files giving the concentration-time simulation results, .BAS files giving results of base case simulations, and .CLC .CLD files with incremental reactivity results. These three files are discussed below.

The concentration-time simulation results are output in .CDA files, which is the same as the output files for normal simulations as discussed in Section 3.3.6. These files go into a subfolder named for the model (given in the MODELSET= or BASEXT= input in MODELING.PRM, which in turn is a subfolder of the folder named in the CAL= input in MODELING.PRM. For example, for the currently distributed files, the MODELING.PRM in the REACT folder has calculation results going into the MODEL\CDTFILES ("MOD=.\CDTFILES\") and the reactivity results going to a subfolder called SAPRC07, so the .CDA files go to the MODEL\CDTFILES\SAPRC07 folder. The base case results are saved in a file called *ScenID*.CDA, where *ScenID* is the name of the scenario as given in the REACTCAL command, and the test simulation results are saved in a file called REACTCAL.CDA. The latter are over-written when the next test calculation is carried out for any scenario, so should be renamed or saved before the next test calculation if it is to be retained.

To obtain the concentration-time data from a base case calculation in a .CSV file that can be imported into spreadsheets, use the command

CDA2CSV *ScenID ModelID* [*outputCSVname*]

for base calculations, or

CDA2CSV REACTCAL *ModelID* [*outputCSVname*]

for the latest test calculation, where *ScenID* is the scenario name, *ModelID* is the name of the model given in the "MODELSET=" or "BASEEXT= in MODELING.PRM (e.g., "SAPRC07"), and *outputCSVname* is the name of the CSV. Note that the scenario name is not needed, since the latest test calculation is used regardless of the scenario.

The results of the base case simulations to into a file called *ScenID*.BAS, located in a subfolder called "CLCFILES" the folder named in the "RXP=" input of modeling.prm. For example, the base case calculations using the distributed files and the SAPRC07 mechanism are in REACT\SAPRC07\ CLCFILES*ScenID*.BAS. This is a plain text file that has two or three parts, and no comment lines. The first part gives the scenario parameters taken from SCENARIO.PRM or from the inputs to the base simulation with this scenario. This is sufficient input for REACTCAL to derive the inputs for simulations using this scenario without having to read SCENARIO.PRM. This is terminated by an "ENDPARMS" input line. The second part gives selected results of the base case simulation. These include maximum, times-to-maximum, final, and sums over the simulation time for all species listed in the "MAXCAL= input in MODELING.PRM, and includes "MAXO3", "TM-O3", "ENDO3", and 'SUMO3" for the ozone results. In addition it includes the correction factor ("CORR") used to derive the total moles of O_3 and other products formed given their final concentrations, and the fractions reacted for the counter species with the various rate constants, as discussed in Section 3.6.1. The latter are input lines such as "FR-KOH *kOH Factor*", where *kOH* is the rate constant in ppm^{-1} min^{-1} units, and *Factor* is the calculated fraction reacted. The correction factor is used when deriving the incremental reactivity results, and the fractions reacted are used to estimate amounts of test compound to add given their OH rate constant.

If MROPT2 is run for this scenario, then the results from this program are appended to the *ScenID*.BAS file and constitute the third part of this file. These include total NO_x (E.NOX), maximum O_3

concentrations (O3MAX), the incremental reactivity for the base ROG and for NO_x [IR(ROG) and IR(NOX)], and a number indicating the closeness to the EBIR point (BD.NOX-HC) for the calculations where the NO_x is varied. The outputs corresponding to MIR, MOIR, and EBIR NO_x levels are indicated using the codes "NOx-MIR", "NOx-MOR", or "NOx-LIM", respectively, at the start of the lines. This can be used as input to REACTCAL to define the corresponding adjusted NO_x scenario. MROPT2 also outputs the incremental reactivities of NO_x and the base ROG for the base case scenario as IR(NOX) and IR(ROG).

The results of both the base and test simulations are also output in plain text files called *ScenID*.CLC and *ScenID*.CLD, in the same folder as used for the .BAS file. The base calculation creates the file if needed or appends to it if it exists, and includes headers indicating the data present and output with the base case results. The results of test simulations are appended to these files, which must exist before the test simulation is run.

The *ScenID*.CLC is the main file used for ozone reactivity results, which are used as inputs to the reactivity processing programs discussed above. One output line is produced for each calculation, and the data are in fixed column format. The data on each line include the test compound (or "BASE" for base resuls), the maximum ozone, and the maximum ozone corrected by the factor that gives the molar ozone yields from the maximum concentration. Test calculation output lines also contain the molar incremental ozone reactivity (moles ozone formed in the base calculation - moles ozone in the base calculation, divided by the moles test compound added), the mechanistic reactivity (incremental reactivity divided by the estimated fraction reacted), the percentage change in the maximum O_3 compared to the base case, and the kinetic reactivity (estimated fraction reacted, derived prior to the simulation). If the test calculation was initiated using a "F=*filename*" option, and the carbon number, molecular weight, and additional compound information is given in the mechanism parameter file, than this information is also included in the output line in the .CLC file at the end of the other data. Note that the file headers also have a column named "CalNo", but calculation numbers are no longer used in the current version of the program.

The *ScenID*.CLD file contains additional results from the base and test calculations, which can be used, for example, to derive incremental reactivities with respect to other results besides maximum ozone. Besides O_3, this file includes results for other species listed in the "MAXCAL=" input to MODELING.PRM, which includes PAN and HO2H in the distributed files for SAPRC07. The base calculation outputs header information identifying the data to be output, results for selected species, times where O_3 data are output, and the O_3 and integrated O_3 data for each of those times. The test calculations output the name of the test compound, the amounts added, the fraction reacted (kinetic reactivity), the results for the selected species, and the O_3 and integrated O_3 data for the times indicated for the base calculation. Use of these files is no discussed further in this document, though programs to utilize them to summarize reactivity results have been written. However, these files are not needed for calculation of reactivity scales based on effects of compounds on maximum ozone concentrations.

3.6.4. Processing and Summarizing Reactivity Calculation Output

Use of the DMSRCT program. The ozone reactivity results calculated by the REACTCAL program and saved in the *ScenID*.CLC file are in molar units, but usually tabulated reactivity scales are given in mass based units, i.e., grams O_3 per gram compound emitted. In addition, incremental reactivity test calculations are conducted only for single compounds (or single model species), not mixtures The program DMSRCT can be used to tabulate all the reactivity results for a scenario, including those of mixtures that have reactivity values for all components, with the reactivity values given in per carbon and (for compounds) per mass units. The program is normally called using the "DMSRCT *ScenID*" command. (The program has several options, but not all of these currently operate properly and are therefore not documented.) The molar reactivity data are taken from the .CLC file, and the molecular weights are taken from the DMS.PRM file discussed in Section 3.4.1. The results are output in the file *ScenID*.RCT file located in a subfolder called "RCTFILES" the folder named in the "RXP=" input of modeling.prm. Only those compounds named in both the scenario's .CLC file and the DMS.PRM file are output in the .RCT file. In addition, test calculation results in the .CLC file that precede output from base calculation results are ignored. This means that re-running a base calculation in effect previous test calculation results to be ignored if DMSRCT is used to summarize the results.

DMSRCT will also optionally read the lumping control file named in the "RCTLPC=" input in MODELING.PRM in order to find REPLACE commands that define simple mixtures in terms of model-independent species names. If the reactivities of the simple mixtures are output in the .RCT file if the reactivities for all of their components are found in the .CLC file.

When reading the DMS.PRM file, the program keeps a list of the number of species names that are associated with complex mixtures. These are indicated by the characters "Mix" starting at column 26 in the input line with the mixture name in DMS.PRM. The program opens the .CMP files with for the mixtures in the folder indicated by the "CMP=" input in MODELING.PRM. If it finds that the .CLC file had reactivity results for all the components, then it calculates its incremental reactivity from those of its components, including those that are simple mixtures, and includes the incremental reactivities per carbon it in the .RCT file.

The .RCT files contain the names of the compound, its kinetic reactivity (fraction reacted), its mechanistic reactivity per carbon (incremental reactivity / fraction reacted), and its incremental reactivities in per carbon (IR/C) and per mass (IR/G) basis. Only reactivities per carbon are given in .RCT for complex mixtures.

Use of the REACTTAB Program. The REACTTAB program provides a useful means to collect incremental reactivity results from the .RCT files created by DMSrct and import them into spreadsheets. The incremental reactivity results can be tabulated in mass (gm O_3 /gm VOC), molar (moles O_3 /mole VOC) or carbon (moles O_3 /mole carbon VOC) units, as selected by the user. Note that the program reads the per carbon reactivities from the .RCT files, which are given for mixtures as well as compounds. If needed, these are converted to mass-based or mole-based reactivities using the molecular weights or carbon numbers read by DMS.PRM (or the file named in the "DMS=" input in MODELING.PRM). Only results for compounds listed in DMS.PRM are output.

REACTTAB is usually run with all the information it needs on the command line, but it can also be run in interactive mode if all output is to go to the user terminal. It must be run in the REACT folder, or at least the same folder where DMSRCT is run. The command line options are

REACTTAB *VOCinfo ScenInfo* [*OutputFile*] [+|*][*Option*] [*StdVOC*]

and the command line parameters are as follows:

- *VOCinfo*: This gives either the name(s) of the compound(s) whose reactivities are to be output, or "F=" followed by the name a file containing a list of compounds. The compounds are identified using the model-independent names in DMS.PRM and the .RCT files. If it is a single compound, *VOCinfo* is just the compound name. If it is several compounds, it is a list of names separated by a space and surrounded by quotes, e.g., "ETHANE PROPANE", where the quotes (") are required. If the compounds are listed in a file, then *VOCinfo* is replaced by F=*filename*, where *filename* is the name of a file (including extension) with a list of VOC names, one name per line and no comments.
- *ScenInfo*: This gives either the name(s) of the scenarios(s) for which reactivities are to be output, or "F=" followed by the name a file containing a list of scenarios. The format is the same as

discussed above for *VOCinfo*. For example, the files EPASCN*type*.LIS in the REACT folder of the distributed files contain the names of all the 49 base case or adjusted NO_x scenarios used by Carter (1994a, 2000e), where *type* = "BS", "MR", "MO", or "NL" for base, MIR, MOIR, or EBIR scenarios, respectively.

- *OutputFile*: This optional parameter gives the full name of the output file, including the extension. If not given, or if *OutputFile*="X" the results are output to the user's terminal. Giving X as the file name allows for specification of non-default options on the command line.
- [+|*]*Option*: This gives a code indicating what data to be output and in what format. A "+" prefix indicates that the results are to be output in CSV format and with 4 figures after the decimal point and a "*" prefix indicates that the results are to be output in fixed-width format, also with 4 significant figures. However, if the output file name ends with the extension .CSV, it forces CSV format output even if the "+" code is not given. If neither prefix is given and the output file does not have the .CSV extension, the output is in fixed-width format but with only two significant figures, which is appropriate given the uncertainties and precision issues in reactivity calculations. The values of *Option* could be as follows:

Code	Output data
(blank)	Carbon based incremental reactivities (moles O_3 /moles carbon VOC) (default)
M	Mass based incremental reactivities (gm O_3 /gm VOC emitted)
MOL	Mole based incremental reactivities (moles O_3 /moles VOC emitted)
MR	Mole based mechanistic reactivities (moles O_3 /moles VOC reacted)
KR	Kinetic reactivities (fraction of emitted VOC that reacts)

Note that the code input is case-sensitive, and must be given in upper case. For example, "m" is an invalid option to get mass-based reactivities. If no option input is given, then the data output are per-carbon reactivities in tabulated format with 2-significant figures.

- *StdVOC*: If relative reactivities are desired, this optional input gives the incremental reactivities relative to the standard VOC, i.e., IR(input VOC) / IR (*StdVOC*). The units are determined by the option input, discussed above. For example, to get the reactivities of a compound relative to ethane in all the base Carter (1994a) scenarios, give the command

REACTTAB *Compound* F=EPASCNBS.LIS *OutputCSVfile* +M ETHANE

Where *Compound* is the name of the compound, EPASCNBS.LIS is the distributed file giving the names of all base scenarios, and *OutputCSVfile* is the output file, which is in CSV format.

Results with multiple scenarios and compounds could be output either with the scenarios specified in the columns and the compounds given in rows, depending on whether the output is to be in CSV format. If the output is in CSV format, then the output is given with the compounds in the columns and the scenarios in the rows, unless the number of compounds is greater than the number of scenarios, in which case the scenarios are given in the columns and the compounds in the rows. If the format is not CSV, the compounds are listed in columns and the scenarios in rows unless the number of compounds exceeds the number of columns. Note that the maximum number of columns is currently set at 40, so having more than 40 scenarios and 40 compounds is not allowed.

4. UPDATED INFORMATION AVAILABLE

The latest version of this document and the software described herein is currently available at the author's website at http://www.cert.ucr.edu/~carter/SAPRC/SAPRCfiles.htm, which currently mapped to https://intra.engr.ucr.edu/~carter/SAPRC/SAPRCfiles.htm. Although this web page should remain accessible using this address, sometimes ucr.edu changes how pages are addressed, so if it is not found, you should be able to find it by doing a web search for "SAPRC modeling software". This site also contains update history for the files and programs, and a link to the current SAPRC mechanism web page at http://www.cert.ucr.edu/~carter/SAPRC.

Although this software has been used by the author for many years, its associated documentation (this document) has only recently been completed and is still in draft form. In addition, the programs are not completely bug-free, especially for features that have not been used frequently. Some of the programs list options available to the user if needed command lines are missing, and some of the listed options are not documented, either because they have not been used recently nor tested with the current programs, or because they were found not to work properly. Therefore, use of undocumented options is not recommended, and output received using them should be checked. Please notify the author at carter@cert.ucr.edu (or the email address given at the web site, if different) if errors, omissions, or unclear writing is found in this document, if the software does not perform as documented, or errors or unexpected behavior is found. I will attempt to update the documentation and either correct the errors or remove options that causes errors as time permits. If corrections are made, the updated document and any updated programs and files will be uploaded to this web site.

Although it is not required, the author requests that people who download this software and files and intend to use it for ongoing work to send an email to the address at the web requesting being added to an email list to be notified of updates or corrections.

Currently the software website has files implementing the software for the SAPRC-07, SAPRC-11 and SAPRC-18 mechanisms. SAPRC-07 is the most widely used version of the mechanism and the most widely version of the MIR reactivity scale (Carter, 2010a,b,e), and is currently the only version where reactivity calculation files are included in the distribution. SAPRC-18 is the latest version of the SAPRC mechanism, but it has not yet been completely evaluated and is subject to change. SAPRC-18 makes extensive use of the SAPRC mechanism generation system (Carter, 2019, 2020a) currently available at http://mechgen.cert.ucr.edu (or do a search for "SAPRC MechGen", or look for links at the SAPRC web pages discussed above), and future version of this may make use of using output from that system as mechanism preparation. Although links between MechGen output and mechanism preparation input exist and are discussed in the draft SAPRC-18 documentation (Carter, 2020b), these are not discussed in the current document.

Future versions of this document and the distributed files may include files for updated or other mechanisms, documentation, programs for linking mechanism generation systems to mechanism preparation input, and information useful for training. People who asked to be on the email list for this software will be notified of significant updates, and others can check the web site periodically and check for changes, errata, or new links that may be of interest.

5. REFERENCES

Carter, W. P. L. (1990): “A Detailed Mechanism for the Gas-Phase Atmospheric Reactions of Organic Compounds,” Atmos. Environ., 24A, 481-518.

Carter, W. P. L. (1994a): "Development of Ozone Reactivity Scales for Volatile Organic Compounds," J. Air & Waste Manage. Assoc., 44, 881-899.

Carter, W. P. L. (1994b): "Calculation of Reactivity Scales Using an Updated Carbon Bond IV Mechanism," Report Prepared for Systems Applications International Under Funding from the Auto/Oil Air Quality Improvement Research Program, April 12. Available at http://www.cert.ucr.edu/~carter/absts.htm#cb4rct

Carter, W. P. L. (2000): "Documentation of the SAPRC-99 Chemical Mechanism for VOC Reactivity Assessment," Report to the California Air Resources Board, Contracts 92-329 and 95-308, May 8. Available at http://www.cert.ucr.edu/~carter/absts.htm#saprc99.

Carter, W. P. L. (2010a): "Development of the SAPRC-07 Chemical Mechanism and Updated Ozone Reactivity Scales," Final report to the California Air Resources Board Contract No. 03-318. January 27. Available at www.cert.ucr.edu/~carter/SAPRC.

Carter, W. P. L. (2010b): "Development of the SAPRC-07 Chemical Mechanism," *Atmospheric Environment*, in press. Accepted January 19.

Carter, W. P. L. (2010c): "Development of a Condensed SAPRC-07 Chemical Mechanism," Final Report to the California Air Resources Board, January 28. Available at http://www.cert.ucr.edu/~carter/absts.htm#csaprc07.

Carter, W. P. L. (2010d): "Development of a Condensed SAPRC-07 Chemical Mechanism, Atmospheric Environment, 44, 5336-5345.

Carter, W. P. L (2010e): “Updated Maximum Incremental Reactivity Scale and Hydrocarbon Bin Reactivities for Regulatory Applications,” Report to the California Air Resources Board, Contract 07-339, January 28, available at http://www.cert.ucr.edu/~carter/SAPRC/MIR10.pdf.

Carter, W. P. L. (2019). "Documentation of the SAPRC-16 Mechanism Generation System, Interim report to California Air Resources Board Contract 11-771, May 25. Available at http://www.cert.ucr.edu/~carter/SAPRC/18, May 25..

Carter, W. P. L. (2020a). "Gateway to the SAPRC Mechanism Generation System," Available at http://mechgen.cert.ucr.edu/. .

Carter, W. P. L. (2020b): "Documentation of the SAPRC-18 Mechanism," Report to California Air Resources Board Contract 11-761, May 29. Available at http://www.cert.ucr.edu/~carter/SAPRC/18. May 29.

Carter, W. P. L., A. C. Lloyd, J. L. Sprung, and J. N. Pitts, Jr. (1979): "Computer Modeling of Smog Chamber Data: Progress in Validation of a Detailed Mechanism for the Photooxidation of Propene and n-Butane in Photochemical Smog", Int. J. Chem. Kinet, 11, 45.

Carter, W. P. L., and F. W. Lurmann (1990): “Evaluation of the RADM Gas-Phase Chemical Mechanism,” Final Report, EPA-600/3-90-001.

Carter, W. P. L. and F. W. Lurmann (1991): “Evaluation of a Detailed Gas-Phase Atmospheric Reaction Mechanism using Environmental Chamber Data,” Atm. Environ. 25A, 2771-2806.

Carter, W. P. L. and Gookyoung Heo (2012): "Development of Revised SAPRC Aromatics Mechanisms," Report to the California Air Resources Board Contracts No. 07-730 and 08-326, April 12, 2012. Available at http://www.cert.ucr.edu/~carter/absts.htm#saprc11.

Carter, W. P. L. and G. Heo (2013): “Development of Revised SAPRC Aromatic Mechanisms,” Atmos. Eviron., 77, 404-414.

Derwent, R. G.,M. E. Jenkin, S. M. Saunders and M. J. Pilling (1998): "Photochemical ozone creation potentials for organic compounds in northwest Europe calculated with a master chemical mechanism," Atmos. Environ. 32, 2429-2441.

Gear, C. W. (1971): "DIFSUB for Solution of Ordinary Differential Equations," Comm of the ACM, 14, 185-190.

Hindmarsh, A. C. (1972): "Gear: Ordinary Differential Equation Systems Solver," Lawrence Livermore Lab Report UCID 2001, Rev. w, 1972.

Peterson, J. T. (1976): “Calculated Actinic Fluxes (290 - 700 nm) for Air Pollution Photochemistry Applications”, EPA-600/4-76-025, June.

TUV (2019). Tropospheric Ultraviolet and Visible (TUV) Radiation model. Available at https://www2.acom.ucar.edu/modeling/tropospheric-ultraviolet-and-visible-tuv-radiation-model