TOTAL-SYSTEM PERFORMANCE ASSESSMENT (TPA) COMPUTER CODE: DESCRIPTION OF EXECUTIVE MODULE, VERSION 2.0

Iterative Performance Assessment, Phase 2

Prepared for

Nuclear Regulatory Commission
Contract NRC-02-88-005

Prepared by

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

August 1993

CNWRA 93-017

TOTAL-SYSTEM PERFORMANCE ASSESSMENT (TPA) COMPUTER CODE: DESCRIPTION OF EXECUTIVE MODULE, VERSION 2.0

Iterative Performance Assessment, Phase 2

Prepared for

Nuclear Regulatory Commission
Contract NRC-02-88-005

Prepared by

B. Sagar and R.W. Janetzke

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

August 1993

ABSTRACT

The Total-System Performance Assessment (TPA) computer code is designed to perform calculations leading to an estimate of the Complementary Cumulative Distribution Function (CCDF) of the future performance of a geologic repository. The CCDF is called for in the Environmental Protection Agency (EPA) standard (40 CFR Part 191) as the primary performance measure for a geologic repository. Other performance measures such as dose to man may also be calculated. To be capable of analyzing a variety of scenarios, the TPA computer code consists of several independent modules, some performing general functions and others performing very specific functions. The Executive Module (Exec) of the TPA is that part of the computer code that controls flow of information between different parts of the TPA and does the final processing to produce the CCDF. The design of the Exec and its relationship to other modules of the TPA are explained in this document.

CONTENTS

CONTENTS (Cont'd)

CONTENTS (Cont'd)

FIGURES

TABLES

ACKNOWLEDGMENTS

Development of the Total-System Performance Assessment (TPA) computer code is a joint effort between the Center for Nuclear Waste Regulatory Analyses (CNWRA), the U.S. Nuclear Regulatory Commission (NRC), Office of Nuclear Material Safety and Safeguards (NMSS), and the Office of Nuclear Regulatory Research (RES). While we are thankful to all the participants in the Iterative Performance Assessment (IPA) task, of which development of TPA is a part, we especially acknowledge the help of N. Eisenberg (NRC), R. Codell (NRC), R. Wescott (NRC), T. McCartin (NRC), J. Park (NRC), R. Baca (CNWRA), J. Walton (CNWRA), and C. Freitas (SwRI). This document contains the description of only the Executive Module of the TPA. The TPA includes many Consequence Modules authored by different participants. Since each module is a complete computer program in itself, full details of each module will be available as separate documents.

The authors wish to thank C. Lin who provided a technical review and W. Patrick who conducted a programmatic review.

This report was prepared to document work performed by the CNWRA for the NRC under Contract No. NRC-02-88-005. The activities reported here were performed on behalf of the NRC Office of Nuclear Material Safety and Safeguards, Division of High-Level Waste Management, and the Office of Nuclear Regulatory Research, Division of Regulatory Applications. The report is an independent product of the CNWRA and does not necessarily reflect the views or regulatory position of the NRC.

FOREWORD

In accordance with the provisions of the Nuclear Waste Policy Act (NWPA, 1982), the U.S. Nuclear Regulatory Commission (NRC) has the responsibility of evaluating a license application and, if appropriate, granting a license for the first (and subsequent, if any) geologic repository for high-level nuclear waste (HLW). This Act was amended in 1987 to designate one site for detailed characterization. The designated site is in the unsaturated region of tuffaceous rocks of Yucca Mountain in Southern Nevada. The Center for Nuclear Waste Regulatory Analyses (CNWRA) at Southwest Research Institute (SwRI) is a Federally Funded Research and Development Center (FFRDC) created to support the NRC in its mission of evaluating and licensing the proposed HLW repository. To meet its licensing function, the NRC will review the application submitted by the U.S. Department of Energy (DOE). One of the critical sections of the license application will deal with the assessment of the future performance of the repository system, which has to meet certain minimum standards established by regulations.

In order to develop capabilities to review the Performance Assessment (PA) in DOE's license application, the NRC and CNWRA are engaged in developing and applying PA methods and models to existing data. Later, at the time of license application review, these methods may be used to conduct independent PA, if the NRC elects to do so.

Because of the large space and time scales involved in estimating repository performance, mathematical models encoded as computer codes are the chosen tools for PA. The repository system consists of designed (or engineered) barriers embedded in the natural geologic setting. Estimating performance of the total system requires that the behavior of these components be projected under possible future conditions. This is obviously a complex task that requires a variety of calculations. The development of the Executive Module (Exec described in this report is a step towards performing these calculations in a systematic manner.

1 INTRODUCTION

1.1 REGULATORY AND TECHNICAL BACKGROUND

The conduct of performance assessment (PA) has, in addition to its other functions, a very specific regulatory purpose - that of determining whether the geologic repository system satisfies the regulatory standards. This is done by comparing the estimated values of the regulatory performance measures with the minimum values of the same measures specified in the regulations. Thus, the PA models must be designed to estimate the regulatory performance measures. In addition to the regulatory function, PA will also be used to design [by the U.S. Department of Energy (DOE)] and judge the adequacy of [by the U.S. Nuclear Regulatory Commission (NRC)] the site characterization program. Because of their interdisciplinary nature, PAs are also used to integrate technical work across disciplines. Operational flexibility in the Total-System Performance Assessment (TPA) model is necessary to meet these varied objectives.

The primary regulations applicable to the high-level waste (HLW) geologic repository were promulgated by the NRC in 10 CFR Part 60 — Disposal of High-Level Radioactive Wastes in Geologic Repositories. Two sections of 10 CFR Part 60 pertain specifically to post-closure performance. These are: (i) Part 60.112 — Overall System Performance Objective for the geologic repository after permanent closure; and (ii) Part 60.113 — Performance of Particular Barriers after Permanent Closure. Part 60.112 makes reference to satisfying the generally applicable environmental standards for radioactivity established by the U.S. Environmental Protection Agency (EPA). These environmental standards referred to in Part 60.112 were promulgated by the EPA in 40 CFR Part 191 in 1985 (EPA, 1985). However, on litigation, certain provisions of these standards were remanded by a federal court. Proposed revisions of 40 CFR Part 191 were under review in early 1993. In late 1992, the U.S. Congress enacted a new law known as the Energy Policy Act according to which the EPA will develop standards applicable specifically to Yucca Mountain that may be different from those in 40 CFR Part 191. However, for the development of the TPA code, Version 2.0, the 1985 EPA standards are followed. The TPA code will be modified, as necessary, at the appropriate time to account for any changes in the EPA rule.

Three different performance measures are used in Part 191. These are: (i) Release of radioactivity over the entire (integrated over space) accessible environment boundary cumulated (integrated over time) over a 10,000 year period after closure must not exceed specified limits at specified levels (Part 191.13 - Containment Requirements), where the preferred method of representing this performance measure is through a Complementary Cumulative (Probability) Distribution Function (CCDF); (ii) Dose to humans in the first 1,000 years after repository closure must not exceed specified limit (Part 191.15 - Individual Protection Requirements), this requirement has no probability attached to it; and (iii) Concentration of alpha, beta, and gamma emitting radionuclides must not exceed specified limits (Part 191.16 - Groundwater Protection Requirements), there is no probability attached to this requirement. While the first performance measure is to consider all future credible scenarios, the other two apply only to undisturbed performance.

In addition, three other performance measures are used in 10 CFR Part 60.113 to define performance of individual barriers (in contrast to total system). These are: (i) Life of the waste package must exceed specified limits [Part 60.113(a)(1)(ii)(A) - Substantially Complete Containment Requirement]; (ii) Release from engineered barriers must be less than specified limits [Part 60.113(a)(1)(ii)(B) - Gradual Release Requirement]; and (iii) Groundwater travel time (GWTT) must be greater than specified limits [Part 60.113(a)(2) - Ground Water Travel Time Requirement].

In all, therefore, there are six distinct performance measures. In general, a TPA code must allow for estimation of the three measures related to 40 CFR Part 191 and preferably, but not necessarily, for the other three related to 10 CFR Part 60.113. Figure 1-1 depicts the six performance measures and lists the steps for their assessment. These steps include conceptualization of processes, as well as assembly of data suitable for input to the mathematical models.

1.2 PURPOSE OF SOFTWARE

The objective of the TPA software is to provide computational algorithms for estimating values of various performance measures described in Section 1.1. Such computations will take into account the interactions among subsystems, components, future states, and processes. When fully developed, the TPA code will permit estimates of system performance without unnecessary conservative assumptions.

The TPA software is designed to simulate the behavior of an HLW repository located in a partially saturated geologic medium. Both the natural system and engineered barriers are included. The ultimate goal of TPA simulations is to estimate the long-term performance of the repository where the performance is measured by release of radiation at a specified boundary or dose commitment to humans affected by the release, or both. The evolutionary change in the geologic setting is described in terms of disruptive scenarios which, in addition to a parametric description of the changed state, also has a probability of occurrence attached to it. The NRC will use the TPA to review critical aspects of the PA contained in DOE's license application.

The TPA software is a collection of several independent modules linked together to perform as a unit. The Executive Module (Exec) is the module that performs the linkage function.

1.3 REPORT CONTENT

A brief overview of the models embodied in the TPA Version 2.0 software is provided in Chapter 2. Features of the Exec module are described in Chapter 3, which includes a detailed description of the input/output (I/O) files. Chapter 4 contains the input instructions for the Exec module. The output options are discussed in Chapter 5. The verification and validation status of the software is discussed in Chapter 6. In Appendix C, requirements of the TPA code, which were decided prior to its development, are included. A sample problem is included in Appendix A.

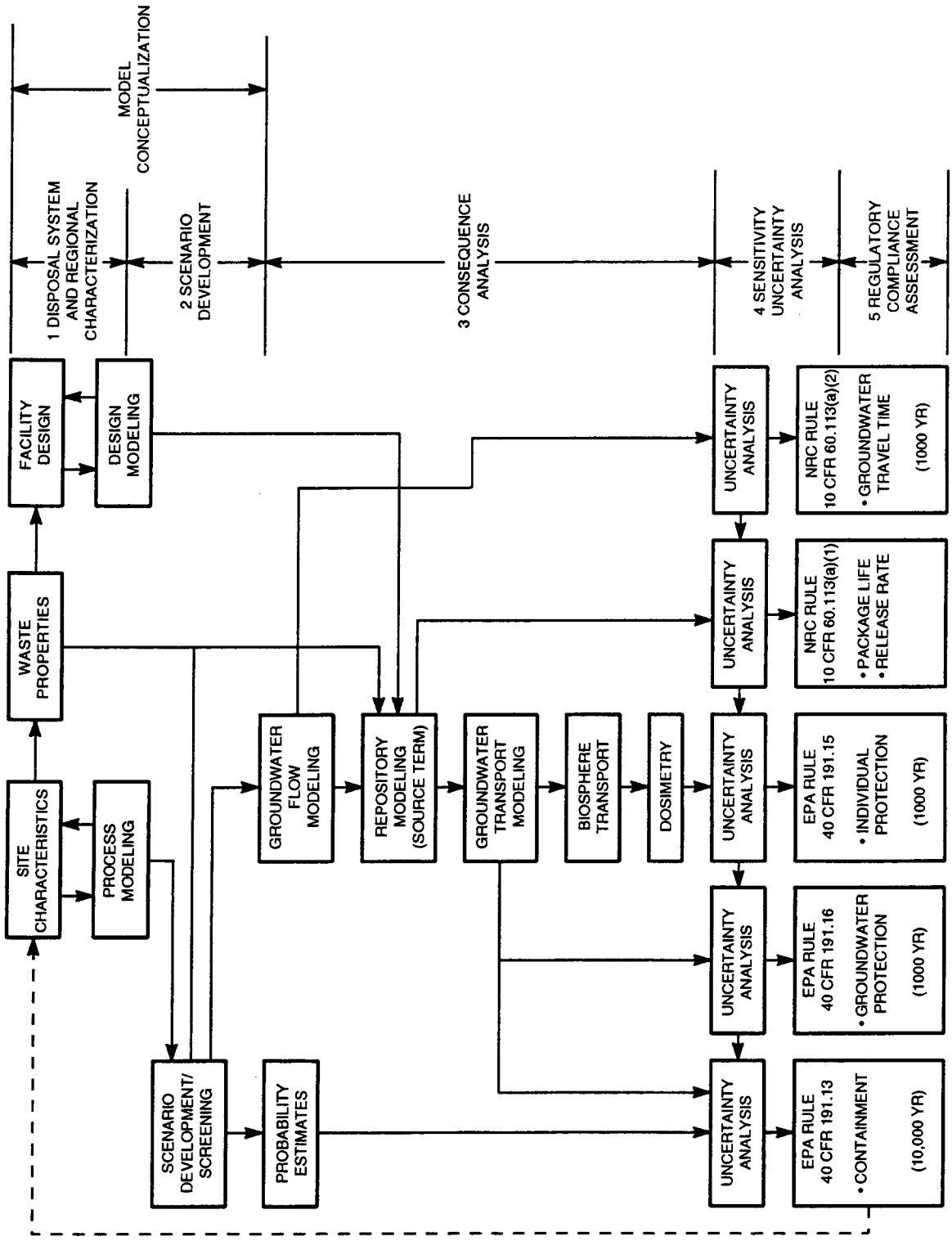

Figure 1-1. Regulatory performance measures

2 MODEL DESCRIPTION

2.1 OVERVIEW

The diverse nature of the performance measures, discussed in Chapter 1, requires that many technical disciplines be involved in their estimation. Consequently, it is desirable to design the TPA as a set of Consequence Modules (CM) which are largely independent computational units with their execution controlled by an Exec. The Exec acts as the manager and assures that CMs are executed in the desired sequence and that appropriate values of the common parameters are passed to CM. Automated features included in the TPA system (TPA system, TPA software, and TPA code are used interchangeably in this document; these terms represent the sum of all modules including the Exec - see Chapter 3) will facilitate the unattended execution of a set of multiple scenarios with associated output.

The requirements for the TPA code were developed early in the design process. These requirements are given in Appendix C.

2.2 THE Exec CONCEPTUAL MODEL

Almost all of the conceptualization necessary to model a repository system is included in modules other than the Exec. The Exec controls the sequence of execution of various modules, transfers data to other modules, and controls data transfer from one module to another. However, the implementation of these actions is kept flexible so that varying conceptualizations of a scenario may be simulated. In other words, no specific conceptual model is embedded in the Exec except for the fact that the general approach of scenario analysis is adopted in contrast to the environmental simulation approach (for a contrast of these two approaches see Thompson and Sagar, 1993[1]). The scenario approach adopted in Iterative Performance Assessment (IPA) Phase 2 is discussed in greater detail in the IPA Phase 2 final report that will be published in the near future.

In the scenario approach, the future state of the repository system is conceptualized to be defined by a set of parameters, some or all of which may be random variables with known probability distributions. This set of parameters is assumed to be time independent for a particular scenario. A different scenario is defined, if parameter values change within the time span of interest (e.g., 10,000 years). In essence, the time change in parameters, if any, is discretized with each discrete value defining another scenario. In the analyses conducted so far, disruptions defining scenarios occur at a specified time and the disturbed state then remains constant. For example, in the climate scenario, precipitation (and hence infiltration rate) may change at time, t, and then remain constant at the new rate.

---

1 Thompson, B.G.J., and B. Sagar. 1993. The development and application of integrated procedures of post-closure assessment based upon Monte Carlo simulation: The probabilistic system assessment (PRA) approach. *Journal of Reliability Engineering & System Safety*. In press.

The TPA code is written to analyze any arbitrary number of scenarios. The actual number of scenarios that can be analyzed, however, may be limited by the attributes of the hardware, e.g., memory, disk storage, and speed of computations.

2.3 THE Exec NUMERICAL MODEL

The random nature of some of the parameters characterizing a scenario is accommodated in calculation of consequences by using the Monte Carlo method. A set of parameter vectors is first created by sampling from the probability distributions of the random variables. The sampling method is implemented in a separate module called Latin hypercube sampling (LHS). The sampling method in LHS ensures that all parameter vectors in a set are equally probable. Thus, if N vectors are included in a set, then each vector has a probability of occurrence of $1/N$. Simulations are performed with each vector in turn resulting in calculation of N consequences. It is then obvious that each consequence is also equi-probable with a probability of $1/N$.

The only calculations performed in the Exec consist of: (i) sorting of consequences with respect to magnitude within a scenario and plotting a CCDF for that scenario; and (ii) obtaining a weighted sum of the consequences to obtain an overall CCDF that includes the effect of all scenarios. The first calculation is straightforward sorting of consequences in a sequence with the lowest as the first value and the highest as the last. Once the sorting is complete, the cumulative probabilities are assigned to each consequence. Since calculated consequences are statistically independent of each other, the estimate of the conditional probability (conditioned on the assumption that scenario, S, occurs) that the nth consequence will be equalled or exceeded is $(N\text{-}n+1)/N$. For example, if 500 samples are used ($N = 500$) and the 100th consequence ($n = 100$) in the sorted sequence is r, then $p\{R > r|\text{Scenario}\} = (500 - 100 + 1)/500 = 0.802$, where R is the random variable representing the actual consequence. The probability estimates improve as N increases. This is especially true of large consequences which are expected to have small probabilities. Illustratively, if $N = 500$, then the best resolution that can be obtained in probability value is 1/500. In addition, a large number of samples increases the probability that parameter vectors leading to extreme consequences are included in the simulation. Normally, a convergence test should be performed to find the number of samples required such that the shape of the CCDF does not alter significantly by the addition of more samples.

For the second calculation, the probability, $p\{R\}$, of a consequence R is obtained as

$$P\{R \le r\} = \sum_{i=1}^{N} P\{R_i \le r | S_i\} P\{S_i\}, \qquad (2\text{-}1)$$

where S_i denotes the scenario and $p\{R|S_i\}$ is the probability of R given that S_i occurs.

For sake of completeness, a brief overview of other modules of the TPA computer code is given in the next section.

2.4 BRIEF DESCRIPTION OF TPA MODULES

The TPA modules that are linked together by the Exec are programs that can be executed by themselves (stand-alone) without the aid of the TPA system. These programs are also referred to as subprocesses in the sense that together they provide the complete process for describing the behavior of the repository. The standard documentation prepared for these programs (or subprocesses) will also apply to their use in the TPA system, with minor modifications to input and output procedures, which will be explained in this document. A brief description of the purpose of these modules is provided.

2.4.1 LHS

The TPA user can specify various parameters pertaining to any number of CMs to be uncertain where the uncertainty is represented through statistical distributions. The LHS module utilizes the LHS method (Iman and Shortencarrier, 1984) to create equally likely parameter vectors. While only uncorrelated variables were used in IPA Phase 2 calculations, the LHS module is designed to sample from correlated variables also. In short, the LHS method is based on dividing the probability space into N number of equal and discrete subspaces. In the case of a single variable, for example, the probability distribution is divided into N parts, where the area under each part is the same. A sample is then drawn randomly from each discrete part. Once the N samples are obtained, their order is randomized to obtain a vector made up of unordered components. The theory for the correlated variables is more complex and can be seen in Iman and Shortencarrier (1984). The LHS program was obtained from the Sandia National Laboratories (SNL). The output of the LHS is written to a file whose size depends upon the number of samples (or vectors) and the number of values in each sample. For instance, if there are n uncertain variables and N samples are required, then the output file will contain N vectors, each vector containing n values, or a total of $N \times n$ values. For further analysis, one vector at a time is read from this file. Two aspects are to be noted: (i) all uncertain parameters, irrespective of which CM they belong to, are sampled at one time; and (ii) for the analysis of any one scenario, a single call to LHS provides all the samples.

2.4.2 CANT2

The time-dependent temperature of the waste container surface is calculated in CANT2. Developed by R. Codell (NRC), the CANT2 module is based upon an analytic solution of the linear heat conduction equation by the principle of superposition assuming a finite number of heat sources. In IPA Phase 2, the repository is assumed to consist of seven cells, each cell comprised of several panels. The temperature of a representative container in each of the seven cells as a function of time is calculated in CANT2. The output is written to a file, which is read by the SOurce TErm Code (SOTEC) module. In IPA Phase 2, none of the parameters used by CANT2 is considered random; these parameters also did not vary with disruptive scenarios. Consequently, CANT2 is executed only once and the resulting temperatures are used by all the vectors of all the scenarios.

2.4.3 FLOWMOD

Simplified one-dimensional (1D) liquid flow calculations in the unsaturated domain are performed in the FLOWMOD module. This CM was developed by T. McCartin (NRC). The flow domain is assumed to consist of a certain number (seven in IPA Phase 2) of vertical columns. The infiltration rate at the top varies from column to column. Once liquid water enters the column at the top, its flow is partitioned between fractures and matrix through a factor that depends upon the rate of infiltration. The rates of infiltration for the columns are determined by detailed two-dimensional (2D) analyses using the double porosity code - DCM3D (Updegraff et al., 1991). For each column, 1D analyses are performed, also using DCM3D, for various infiltration rates to obtain a relationship between infiltration rate and the fracture-matrix flow partitioning factor. FLOWMOD uses these relationships to determine the mass fluxes and particle travel times for each of the stratigraphic units comprising a column. Note that the DCM3D is not a part of the TPA code. It is executed separately to create the input data for FLOWMOD.

2.4.4 C14

In IPA Phase 2, ^{14}C is considered to be the only radionuclide that can be transported in a gaseous phase. The C14 CM calculates the transport rates of $^{14}CO_2$. Authored by R. Wescott (NRC) and R. Codell (NRC), this module calculates its own far-field temperature field, which is then used to determine time-dependent gas velocities. The amount of $^{14}CO_2$ released from the repository, as calculated by the SOTEC module, is provided to C14 as an input. Far-field migration is then calculated by solving numerically the convection-diffusion equation. Further details on C14 module will be provided in its users manual.

2.4.5 SOTEC

SOTEC deals with the calculation of aqueous and gaseous radionuclide time- and space-dependent source terms. As mentioned above, ^{14}C is the only radionuclide that is treated in the gaseous phase in IPA Phase 2. However, any number of radionuclides can be dealt with in the aqueous phase. Three primary calculations are done in SOTEC: (i) failure of waste containers due to a combination of corrosion processes and mechanical stresses; (ii) the leaching of spent fuel; and (iii) the release of ^{14}C gas from the oxidation of UO_2 and other components in the spent fuel and hardware. In Version 1.0 of SOTEC (Sagar et al., 1993), general corrosion, pitting, and crevice corrosion are modeled based on a temperature-dependent corrosion potential. The temperatures obtained in CANT2 are provided as inputs to SOTEC. Leaching rates are considered to be either solubility limited or congruent to UO_2 rates. More details are available in Sagar et al. (1992).

2.4.6 NEFTRAN

Far-field transport of radionuclides in the aqueous phase, from repository to the accessible environment boundary, is coded in the NEFTRAN II code (Olague et al., 1991). This code was obtained from SNL. The pore velocities calculated by FLOWMOD are provided as input to NEFTRAN. In

NEFTRAN, the transport domain is considered to be made up of 1D transport paths (or tubes or legs), along which the convection-diffusion equation is solved semi-analytically by a method called the distributed velocity method (DVM). Details of the method are available in Olague et al. (1991).

2.4.7 CLIMATO

This is a place holder for a future climate-related module of the disruptive event. In IPA Phase 2, climate change is treated by specifying climate-dependent infiltration rate and water table position for use in FLOWMOD.

2.4.8 SEISMO

SEISMO is a module for calculating consequences of a seismic event. The probability of occurrence of an event of certain magnitude is considered to be time-dependent. To simplify, a seismic hazard curve representing time-dependence of earthquake magnitudes (peak accelerations) at a certain probability level (e.g., 95 percent) is first obtained. This curve is obtained from a family of postulated plots between the occurrence probability versus earthquake magnitude for a set of fixed time periods. Based on the structural properties of the container material, a fragility curve representing a relation between peak acceleration and the critical container wall thickness is derived. The actual container wall thickness as affected by corrosion processes is obtained from SOTEC as a function of time. Any time the critical wall thickness obtained from the fragility curve is greater than the actual thickness produced by the SOTEC, failure occurs. The number of such failures is fed back to SOTEC for calculation of the source term.

2.4.9 DRILLO1

The human intrusion scenario is stipulated to consist of drilling in and around the repository. The location of drill holes and the timing of drilling is assumed to be random (see NUREG-1327). The drill bit can either directly hit a waste package or it may only penetrate rock which may be contaminated. Radioactive material may be brought to the surface in either case. The calculated number of direct hits are supplied to the SOTEC models for inclusion in the calculation of source term.

2.4.10 DRILLO2

Consequences from the drilling scenario are calculated in DRILLO2. A drill bit directly hitting a waste package or penetrating contaminated rock is assumed to lift a certain portion of the radionuclide inventory to the ground surface. The inventory in a waste package and in the rock surrounding waste packages is tracked in SOTEC as a function of time, and is supplied to DRILLO2. A small percentage of the radioactive material brought to the surface is assumed to be particulate material which becomes airborne. This information is then provided to the AIRCOM module for calculation of human dose.

2.4.11 VOLCANO

Consequences due to magmatic events are calculated in the VOLCANO module. This module is based on the work of Margulies et al. (1992). In the geometric approach followed in VOLCANO, Monte Carlo sampling is used to generate a volcanic event randomly in a rectangular region surrounding the repository horizon. Random sampling is used to decide: (i) the location of the sampled volcanic eruption; (ii) the nature (intrusive leading to dike formation and extrusive leading to cone formation) of the volcanic event; and (iii) the dimensions of the dike or cone. From the area of the repository intercepted by the dikes and cones, the numbers of waste packages failed by the magmatic event are determined. This information is used in SOTEC to calculate the source term. When the volcanic event leads to the formation of a cone, release of radioactivity to air is calculated, which is then used in AIRCOM to calculate human dose and in Exec to calculate the total release. A more detailed description of the VOLCANO module is given in Magmatic Scenario Code User Guide by Lin et al., (1993)[2].

2.4.12 AIRCOM

This module is mainly utilitarian in nature and does not perform any calculations relative to the physics of the TPA system except for the introduction of respirable fractions for the drilling and volcanic disruptive events. Its main purpose is to merge the various gaseous release data files into one file in the proper format for use by the Dose in Ten Thousand Years (DITTY) module. The gaseous release files are generated by VOLCANO, DRILLO, and C14.

2.4.13 DITTY

Transport of radioactivity to a biosphere is modeled in the DITTY module. Developed originally for the Hanford site, this code was obtained from the Battelle Pacific Northwest Laboratories (PNL) in Richland, Washington. This module considers both air and liquid transport pathways and calculates both the individual and population doses. The module is designed to deal with both acute and chronic releases, and, annual, committed, or accumulated doses can be calculated. Several of the Hanford site-specific data coded in DITTY were modified. A second generation of dose calculation codes (GENII), which includes the original DITTY code, is currently available from PNL. For conceptual models included in DITTY and its user's manual, see Napier et al. (1988).

2.5 SENSITIVITY AND UNCERTAINTY ANALYSES

Sensitivity and uncertainty analyses are not integral to TPA Version 2.0. Instead, these analyses are performed in a post-processing phase using the SPLUS statistical package and other available

---

2 Lin, C.S., R.G. Baca, and R. Drake. 1993. *Magmatic Scenario Code User's Guide.* CNWRA 93-010. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses. In preparation.

statistical routines. Among other products, the sensitivities of the final results (i.e., CCDF) to selected parameters are generated. The actual method used for IPA Phase 2 is explained in the final IPA Phase 2 report.

3 Exec SOFTWARE DESCRIPTION

The TPA Version 2.0 computer code is about 20,000 lines of code and, with data areas, requires about 8 megawords of memory. This code must control and interface with up to 13 subprocesses (see Chapter 2). As stated before, each subprocess (or module) is an independent computer program with its own set of input and output files. In addition to those standard input/output (I/O) files, the Exec creates several files for transfer of data from one module to another. In this chapter, the organization of the TPA code is described first. Thereafter, the mechanics of data transfer between various modules is explained. At the end, hints for code maintenance are provided.

3.1 TPA CODE ORGANIZATION

Organizationally, the TPA code can be thought of as made up of four parts: (i) the executive (or manager); (ii) algorithm(s) to sample from probability distributions; (iii) algorithm(s) to compute consequences; and (iv) algorithm(s) to compute sensitivities and perform uncertainty analyses. The parts listed under (ii) and (iii) are called subprocesses and were briefly described in Chapter 2. Version 2.0 of TPA does not include any module for sensitivity/uncertainty analyses; these analyses are performed externally by post-processing TPA output. The Exec of the TPA directs data flow between different subprocesses and controls their execution. Figure 3-1 shows schematically the organization of Version 2.0 of the TPA code. The shaded parts of Figure 3-1 represent the Exec.

The TPA system uses a dedicated subroutine to handle the setup and initiation of each subprocess. The subprocess is the result of FORTRAN CALL statement invoking a utility routine, which is specific to the operating system. This CALL starts the subprocess. Control is returned to the Exec at the end of the execution of the subprocess.

3.2 DATA HANDLING AND CONTROL

The TPA I/O files can generally be divided into four different types: (i) input; (ii) temporary; (iii) reusable; and (iv) output. The input files include the TPA input file, LHS standard input file, and standard input files for each of the CM.

The temporary files include the files generated by the TPA system for the purpose of transmitting control parameters to a CM. These typically contain global parameters (parameters that are common to more than one module) which can override the parameters read from the standard input file of the module. The temporary files are named inside the TPA and are deleted at the end of the execution. In general, the end user does not need to prepare or make arrangements for the temporary files, other than preventing possible file name conflicts in the default directory. A data flow diagram indicating inter-module communication interfaces is shown in Figure 3-2, and all file names are listed in Appendix E.

The reusable files are those in which intermediate results from various CMs are written. The data in these files may be processed later by other modules. For example, the releases of radionuclides

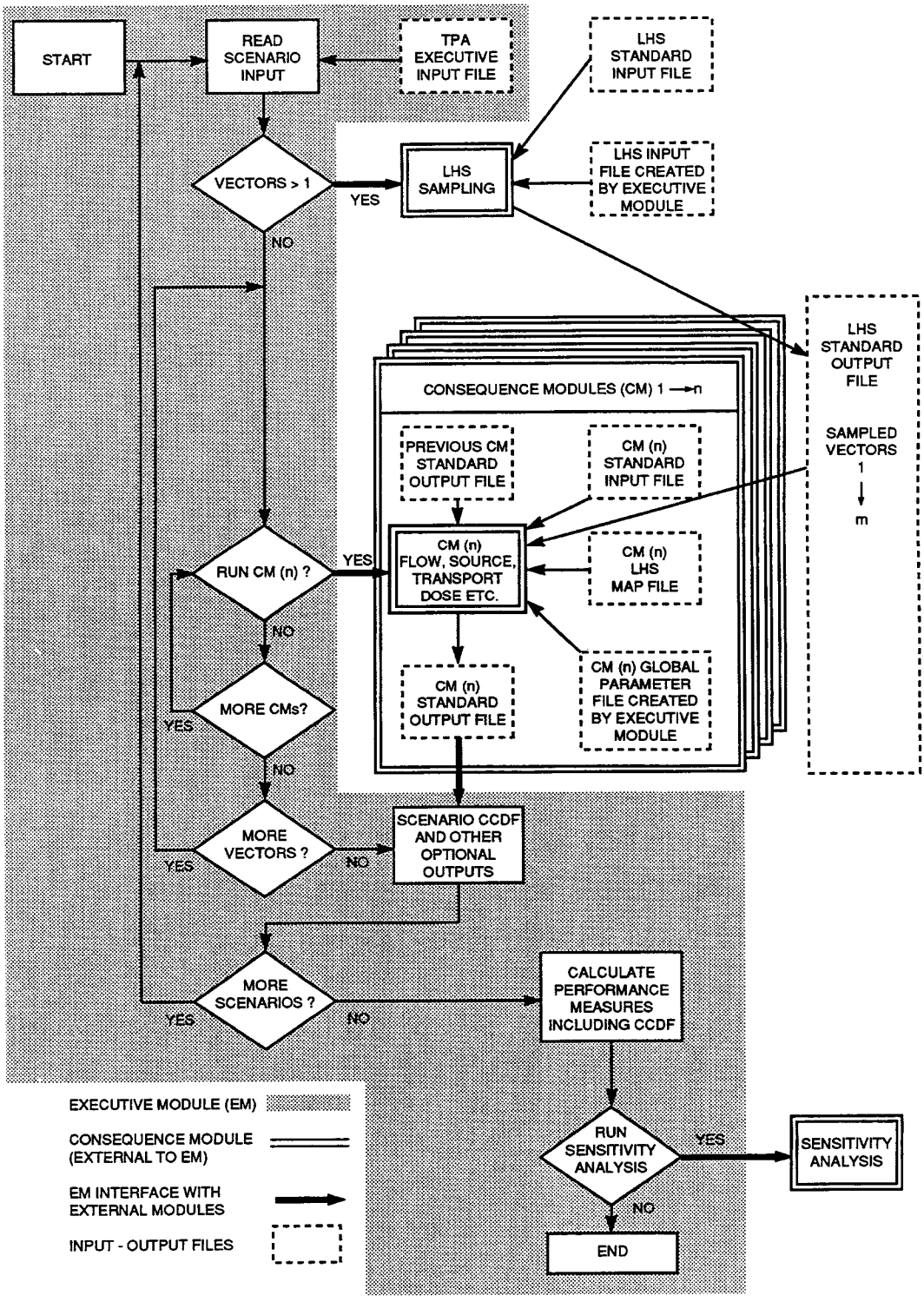

Figure 3-1. Organization of the TPA computer code

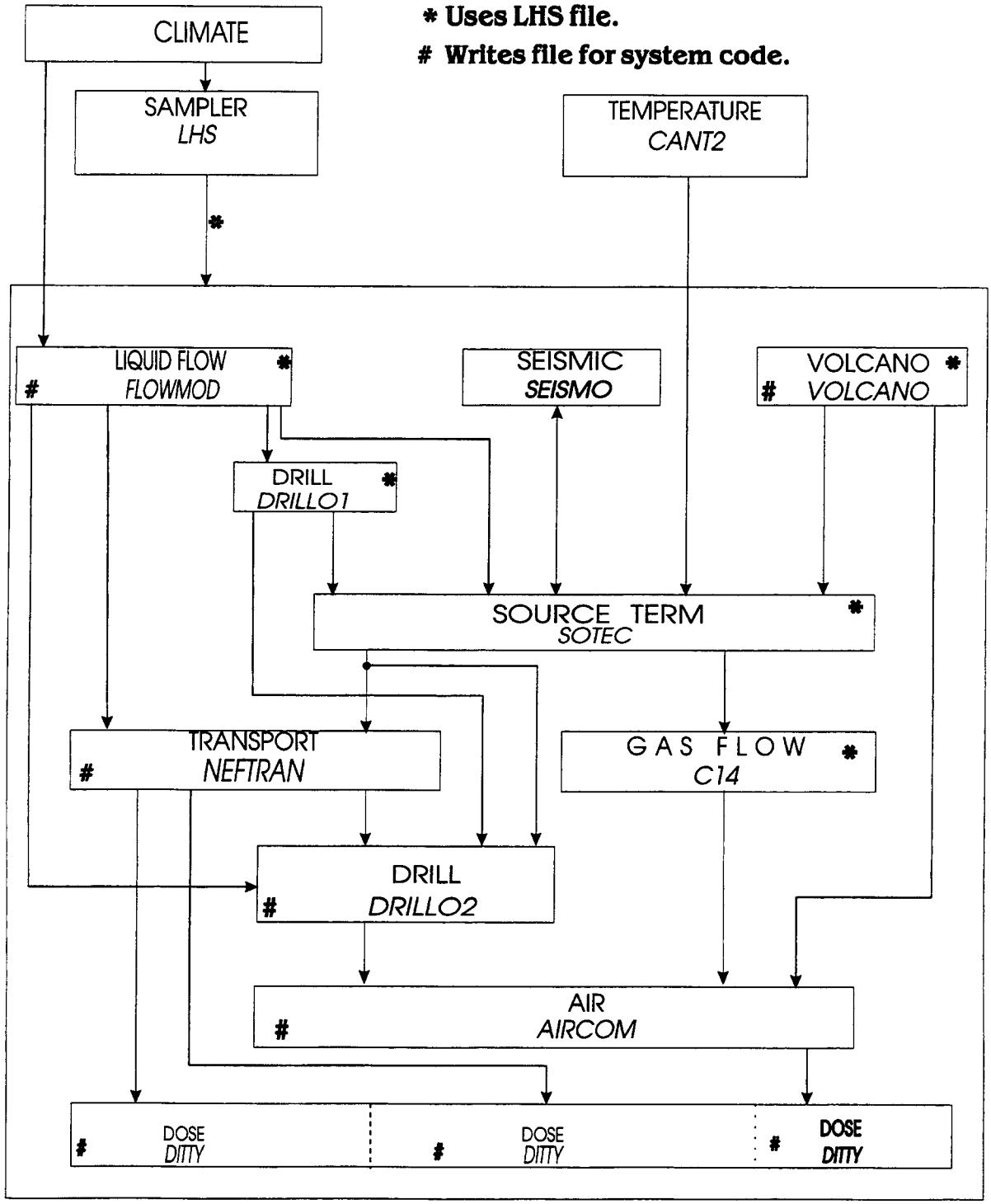

Figure 3-2. TPA system flow diagram

calculated by a CM (e.g., DRILLO) will be written in a reusable file. These files are saved and can be used at the end to analyze intermediate results.

The output files include the output file of the TPA, standard output files of each one of the CMs, error log files, and specially formatted files for external utilities, such as the TECPLOT graphics utility (TECPLOT, 1990).

The names of most files except the temporary files can be read as part of the input. The format-free input process of the TPA Exec is explained in detail later. This format-free input is based on the use of keywords. Keywords used to read the names of the files are found in Chapter 4.

3.2.1 Global Data

One of the primary requirements of the TPA is that parameters that are common to many subprocesses be consistent among these subprocesses. It is also desirable to keep the codes in a form where they could be run independently as well. The resolution of these generally conflicting goals was to maintain consistency through the use of temporary global data files.

Global data are supplied by the analyst in the main TPA input file and also generated automatically by the Exec. These data are then distributed to the appropriate CMs as needed through the use of temporary files. Obviously, the values of corresponding variables contained in the subprocess standard input file are to be overwritten by the values in the global data file. The names of these temporary files have been standardized to facilitate the control of the CMs. The first four characters of each name are TPA_. The remainder of the name is unique for each CM. Incidentally, the presence of a module's global data file is used as a flag to tell the CM to read the LHS file for sampled data and to generate files required by the Exec in addition to its normal output files. The sections below describe the contents of each of the global data files.

3.2.2 Concept of an LHS Map File

There are certain sampled variables that are specified in the standard LHS input file. A map file giving the correspondence between such variables in the LHS output file and the variables in the standard input file of a particular subprocess is provided by the analyst and additionally serves to indicate to the subprocess the variables to be overwritten. The map file indicates the location of variables that are sampled by LHS and whose sample vectors therefore exist in the LHS output file.

3.3 DETAILED DESCRIPTION OF I/O FILES

3.3.1 Description of the TPA Exec Input File (TPA.INP)

The standard input file for the TPA Exec consists of three sections: (i) header information; (ii) scenario data; and (iii) output instructions. The header section contains two data lines: the first line specifies the title of the problem; and the second line specifies the identification of the analyst. The

scenario section contains the specific scenario information for each scenario to be run. There will be as many subsections in the scenario section as there are scenarios to be analyzed. The output instructions are given in the calculation section and specify the disposition of output data items.

The construction of the standard input file for the TPA Exec is based on the format-free RDFREE (Janetzke and Sagar, 1991b) library of subroutines. Even though the input is referred to as format-free, there are, nevertheless, a few rules to follow when composing an input line. These rules are much less restrictive than the usual FORTRAN input statements and can be found in the RDFREE utility User's Guide (Janetzke and Sagar, 1991b). The RDFREE set of subroutines is used to scan the input for recognizable words and numbers and assigns the appropriate values to the internal variables.

The descriptions of the specific commands used for TPA Exec are given in Chapter 4 of this document. An example of an input file for the TPA Exec is provided below.

EXAMPLE OF TPA Exec STANDARD INPUT FILE

```
  \\ Header Section \\
TITLE: TPA test scenario input file.
ANALYSt 'Ron Janetzke CNWRA'|


\\ Scenario section.\\
SCENARio NAME 'Scenario 00 - base case'
         write OUTPUT on 'scen00.cum' file
         write CCDF results on 'scen00.cdf' file
         PROBABility of this scenario is .98
         START time of this scenario is 0 yrs after closure
         STOP time of this scenario is 10000 yrs
         calculate rel of NUCLIDes 'AM243' 'PU239' 'U235' |


 \***** LHS is to be executed first *****\
LHS vectors = 5
    read INPUT from file named 'lhs_00.inp'
    write OUTPUT on 'lhs_00.oul' |

 \***** execute NEFTRAN next *****\
NEFTRAn INPUT file name is     'nefii_00.inp'
        LHSMAP file name is   'nefmap_00.dat'
        write standard OUTPUT on    'nefii_00.out'
        write RELEAS output on  'nefil_00.rel'
        write DISCHArge output on 'nefii_00.dis'
        nuclide SOURCE input is on     'nefii_00.src' |

 \ ===== end of input for scenario 00 =====
run |


\\  Input for scenario no. 2 \\
SCENARio NAME 'scenario 01'
         OUTPUT file name = 'scen_01.cum'
```

```
          CCDF output file name = 'scen_01.cdf'
          PROBABility of scenario = .01
          START time of scenario = 3500 yrs
          STOP time of scenario = 10000 yrs
          NUCLIDes of interest are 'AM243' 'PU239' 'U235' ¦
LHS vectors = 5
    standard INPUT file name 'lhs_01.inp'
    standard OUTPUT file name 'lhs_01.ou2' ¦
NEFTRAn REUSE release file  'nefil_01.rel'¦
run ¦

\\ Input data for scenario no. 3 \\
SCENARio NAME 'scenario 11'
          PROBABility = 0.01
          REUSE output file named 'scen_00.cum'¦
run ¦
END of all scenarios ¦

\\ Output instructions \\
OUTPUT
      show on SCREEN
      print CUMREL, PATHREl, NUCREL, SCENREl, EPASUM,
      and CCDF ¦

END of all input¦
```

(Note: The keywords/modifiers are identified in uppercase. These need not be entered in uppercase in the actual input file.)

3.3.2 Description of the Standard LHS Input File

The method for constructing the LHS input file is described in the LHS documentation (Iman and Shortencarrier, 1984). Please refer to this document for definitions of FORTRAN variable names used below. Compared to the format-free style of the TPA Exec input file described previously, the LHS standard input file is of fixed-format type.

The first line in the LHS input provides the title of the problem. This title will be written in the standard LHS output file. Preferably, this title should be the same as the one in the TPA Exec file. The remaining lines in the LHS input file consist of keywords (beginning in the first column) followed by numeric values in fixed format.

Note that while using the LHS as part of the TPA system, the NOBS keyword may be omitted if the VECTORS modifier is used in the standard TPA input file (see previous example). If both NOBS and VECTORS are used, the VECTORS value obtained from the TPA Exec will overwrite the NOBS value. This is accomplished through a slightly modified input section in the LHS program that looks for this global parameter in the LHS global data (TPA_LHS.LGD) file. The value for NREPS should be always set to 1, since the LHS data set for the next scenario will be generated during the next call to the LHS program. For details of statistical distributions that can be invoked, the documentation of LHS should be consulted (see example of file lhs20n.inp in Appendix A). Note that variable names are not

required, but are included to reflect the names used in the CM. Notice also that some variables are followed by additional text for clarity.

3.3.3 Description of the LHS Global Data Input File (TPA_LHS.LGD)

The LHS global data input file is a temporary file created by the TPA program and is not supplied by the analyst. It contains the values of variables that have been selected to overwrite the standard LHS input values. These include the I/O file names and the number of vectors requested, as shown in the following example.

EXAMPLE OF LHS GLOBAL DATA TEMPORARY INPUT FILE

```
TPA temporary file for LHS global parameters.
lhs.inp
lhs.out
       50
```

3.3.4 Description of the C14 LHS Parameter Map File

The input file that specifies a mapping (or link) between the members of the sampled vector from the LHS and the variables in the standard input of C14 is named C14MAP.DAT.

The C14MAP.DAT file contains a title line, a column header line, and a line for each variable sampled in the LHS output file. Each data line contains three fields. The first field is six columns wide for the FORTRAN name of the variable, and the second field is an integer which points to the element of the LHS vector corresponding to the variable named in the first field. The third field is a count of the number of consecutive locations in the LHS vector which apply to the variable. The count is one for scalers, but may be larger for arrays. A period (.) in column one can be used to ignore the sampled value of that variable for a particular run. This feature permits a map file to be set up for a subprocess in general with all the names of the random variables included in it. Then simply putting a period in front of any of these variables indicates that variable is not to be replaced from the LHS.OUT file. Note that it is possible to map either a scaler or an array.

EXAMPLE OF MAP FILE C14MAP.DAT

```
TITLE: A map file for C14 variables in the LHS output file.

VARIABLE INDEX COUNT
ALPHA      1     1
AKR        2    10
PORK      12    10
RETARD    22    10
```

3.3.5 Description of the C14 Global Data Temporary Input File (TPA_C14.CGD)

This is a temporary file created by the TPA to supply global data to the C14 module.

Line 1: title
Line 2: temperature data input file
Line 3: temporary file for velocity data
Line 4: LHS map file
Line 5: LHS sampled data file
Line 6: input file from SOTEC with C14 rates
Line 7: output file for AIRCOM
Line 8: current LHS vector number
Line 9: first time step, time step size, last time step

EXAMPLE OF GLOBAL DATA FILE TPA_C14.CGD

```
TPA temporary file for C14 global parameters.
/c14h/fdtemp2a.out
c14h.vel
/c14h/c14map.dat
lhsoooo.out
sotc14.dat
c14air.dat
       50
        0        50     10080
```

3.3.6 Description of the NEFTRAN Global Data Temporary Input File (TPA_NEF.NGD)

The NEFTRAN global data input file is created by the TPA system and is not supplied by the analyst. The file contains the I/O file names required by NEFTRAN II and two scenario global variables. One is the scenario vector number currently being executed by the TPA program. The other is the time of the end of the simulation. This variable is called TUB in the NEFTRAN II code and is supplied by the analyst with the NEFTRAN STOP keyword in the TPA Exec input file.

EXAMPLE OF NEFTRAN GLOBAL DATA TEMPORARY INPUT FILE (TPA_NEF.NGD)

```
TPA temporary file for NEFTRAN global parameters.
flonef.dat
nefrep.out
neftpa.dat
nefrep.dis
nefrep.src
- blank -
lhsoooo.out
```

```
     50
10080.0
```

3.3.7 Description of the Reusable NEFTRAN Nuclide Release Data File

The reusable NEFTRAN release data file is created by NEFTRAN II and is not supplied by the analyst. However, the disposition of the file is under the control of the analyst since the file is not deleted by the TPA program. The file contains the release values for all the input vectors of a given scenario. The file is complete at the end of a scenario and the data in it may be used in the next scenario if desired. The file may also be stored by the analyst in another directory for use at a later time in another run of the TPA system. When a file is to be reused, it must be generated by a scenario with the same number of input vectors as the scenario in which it is to be reused.

The release data calculated by NEFTRAN are initially presented to the TPA system via standard formatted files. The file name is specified with other global parameters in the TPA.INP file. As NEFTRAN is invoked for each vector, it appends the section of data for that vector onto the end of the file, and then closes the file. After all vectors are finished the TPA Exec process then opens the file and reads the release data for all of the vectors. However, only data for nuclides that are requested when the SCENARIO NUCLIDES keyword are transferred, the others are ignored and remain zeroes.

After the title record the file is divided into sections, one section for each vector. The first record of a section identifies the vector by the number for which it applies. The release data may contain any number of records. Each release record contains the nuclide name in a CHARACTER*6 field followed by a floating point release value.

EXAMPLE OF A NEFTRAN REUSABLE DATA FILE

```
NEFTRAN release (Ci) data file to be read by the main process.
Vector                1
AM243    5.9539E+02
PU239    5.2631E+05
U235     2.7302E+02
Vector                2
AM243    9.0037E+02
PU239    7.8607E+05
U235     4.0625E+02
Vector                3
AM243    5.5569E+02
PU239    4.8643E+05
U235     2.5160E+02
Vector                4
AM243    2.1601E+02
PU239    1.8767E+05
U235     9.6841E+01
Vector                5
AM243    7.4751E+02
PU239    6.6262E+05
U235     3.4403E+02
```

3.3.8 Description of the FLOWMOD Parameter Map File

This file is created by the TPA for use by the FLOWMOD module. The data in this file indicates the location of FLOWMOD-specific parameters that were sampled by the LHS module. In the example below PERMM represent matrix permeability which has seven values (one for each layer) and the first value is at location number 32 in the sampled vector.

Line 1: title
Line 2: header
Line 3: matrix permeability pointer
Line 4: fracture permeability pointer
Line 5: matrix porosity pointer
Line 6: fracture porosity pointer
Line 7: matrix beta pointer
Line 8: fracture beta pointer
Line 9: gradient pointer
Line 10: dispersion pointer
Line 11: infiltration pointer
Line 12: kd pointer
Line 13: area parameter pointer

EXAMPLE OF FLOMAP.DAT FILE

```
TITLE: FLOWMOD map file for undisturbed sampled variables from LHS.
VARIABLE INDEX COUNT
PERMM     32     7
PERMF     39     7
PORM      46     7
PORF      53     7
BETAM     60     7
BETAF     67     7
GRADIE    74     7
DISPER    81     1
TINFIL    82     1
KDM       83   126
AREAO    209     1
```

3.3.9 Description of the FLOWMOD Global Data Temporary Input File (TPA_FLO.FGD)

Global data required by FLOWMOD is written to this temporary file by the TPA.

Line 1: title
Line 2: infiltration file name
Line 3: matrix flux file name

Line 4: LHS map file name
Line 5: LHS output file name
Line 6: current vector number
Line 7: stop time
Line 8: variable floare
Line 9: variable flossat
Line 10: variable flouns
Line 11: number of chains
Line 12: number of nuclides
Line 13: number of elements
Group 1: members per chain
Group 2: nuclide names
Group 3: nuclide mass
Group 4: nuclide element number
Group 5: nuclide parent 1 list
Group 6: nuclide parent 2 list
Group 7: parent 1 branching fraction
Group 8: parent 2 branching fraction
Group 9: nuclide inventory
Group 10: nuclide half-life
Group 11: nuclide EPA weighting factor
Group 12: inlet area data
Group 13: outlet area data
Group 14: unsaturated leg lengths
Group 15: saturated leg lengths
Group 16: matrix grain data

EXAMPLE TPA_FLO.FGD file

```
TITLE: TPA temporary file for FLOWMOD global parameters.
[rjanetzke.flowmod]infil.tbl
[rjanetzke.flowmod]flux.tbl
flomap.dat
lhsrep.out
         2
  10000.0
         7
         7
         7
         2
         5
         4
         3         2
 NP237  U233    TH229   PU240   U236
    237.0     233.0     229.0     240.0     236.0
         1         2         3         4         2
         0         1         2         0         1
         0         0         0         0         0
```

```
     0.0       1.0       1.0       0.0       1.0
     0.0       0.0       0.0       0.0       0.0
  0.217E+05  0.266E+01  0.196E-02  0.315E+08  0.154E+05
  0.214E+07  0.162E+06  0.734E+04  0.658E+04  0.239E+08
  0.100E+01  0.100E+01  0.100E+01  0.100E+01  0.100E+01
  0.127E+07  0.144E+07  0.148E+06  0.218E+06  0.199E+06  0.786E+06  0.134E+07
  0.262E+06  0.262E+06  0.262E+06  0.262E+06  0.262E+06  0.262E+06  0.262E+06
  0.600E+02  0.700E+02  0.400E+02  0.300E+02  0.600E+02  0.000E+00  0.000E+00
0.600E+02  0.000E+00
  0.130E+03  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.600E+02  0.000E+00
0.700E+02  0.000E+00
  0.000E+00  0.000E+00  0.000E+00  0.600E+02  0.200E+02  0.140E+03  0.000E+00
0.000E+00  0.000E+00
  0.000E+00  0.600E+02  0.700E+02  0.000E+00  0.600E+02  0.000E+00  0.000E+00
0.000E+00  0.600E+02
  0.140E+03  0.000E+00  0.000E+00  0.900E+02  0.100E+03  0.400E+02  0.600E+02
0.300E+02  0.100E+03
  0.200E+02  0.000E+00  0.000E+00  0.000E+00
  0.150E+04  0.000E+00  0.200E+04  0.125E+04  0.000E+00  0.000E+00  0.000E+00
0.150E+04  0.000E+00
  0.200E+04  0.125E+04  0.000E+00  0.000E+00  0.000E+00  0.150E+04  0.000E+00
0.200E+04  0.125E+04
  0.000E+00  0.000E+00  0.000E+00  0.150E+04  0.000E+00  0.200E+04  0.125E+04
0.000E+00  0.000E+00
  0.000E+00  0.150E+04  0.000E+00  0.200E+04  0.125E+04  0.000E+00  0.000E+00
0.000E+00  0.150E+04
  0.000E+00  0.200E+04  0.125E+04  0.000E+00  0.000E+00  0.000E+00  0.150E+04
0.000E+00  0.200E+04
  0.125E+04  0.000E+00  0.000E+00  0.000E+00
  0.258E+04  0.237E+04  0.223E+04  0.259E+04  0.227E+04  0.263E+04  0.227E+04
```

3.3.10 Description of the SOTEC Parameter Map File

Location of sampled parameters used by SOTEC is indicated in this file. The index (second column) is the location of the first value of an array and the count (third column) provides the number of values in that array.

Line 1: title
Line 2: header
Line 3: ecorr pointer
Line 4: carbon pointer
Line 5: forward pointer
Line 6: warea pointer
Line 7: rpor pointer
Line 8: rdiff pointer
Line 9: volmax pointer
Line 10: rde pointer
Line 11: sol pointer
Line 12: funnel factor pointer

EXAMPLE OF SOTMAP.DAT FILE

```
TITLE: SOTEC map file for sampled variables from LHS.
VARIBL INDEX COUNT
ECORR    210     8
CARBON   218    15
FORWAR   233     7
WAREA    240     7
RPOR     247     7
RDIFF    254     4
VOLMAX   258     7
RDE      265    36
SOL      301    36
FUNNEL   337     7
```

3.3.11 Description of the SOTEC Global Data Temporary Input File (TPA_SOT.SGD)

This file is read by SOTEC to obtain global data from TPA.

Line 1: title
Line 2: SOTEC standard input file name
Line 3: LHS map file name
Line 4: LHS output file name
Line 5: scenario name using list directed string format
Line 6: current vector number
Line 7: stop time
Line 8: waste inventory for repository
Line 9: number of areas
Line 10: number of cans per area
Line 11: canister length, radius, thickness
Line 12: number of chains
Line 13: number of nuclides
Line 14: number of elements
Group 1: members per chain
Group 2: nuclide names
Group 3: nuclide mass
Group 4: nuclide element number
Group 5: nuclide first parent
Group 6: nuclide second parent
Group 7: first parent branching fraction
Group 8: second parent branching fraction
Group 9: nuclide inventory
Group 10: nuclide half-life

EXAMPLE OF TPA_SOT.SGD FILE

```
TITLE: TPA temporary file for SOTEC global parameters.
../sotec/sotec.inp
../sotec/sotmap.dat
lhsoooo.out
'OOOO'
       50
    10080
     70.0
        7
     2335     6150     4875     3675     1275     5625     1073
  0.4700000E+01  0.3300000E+00  0.1000000E-01
       13
       20
       15
        2        3        2        1        4        1        1        1
        1        1        1        1        1
  'CM246 ', 'U238  ', 'CM245 ', 'AM241 ', 'NP237 ',
  'AM243 ', 'PU239 ', 'PU240 ', 'U234  ', 'TH230 ',
  'RA226 ', 'PB210 ', 'CS137 ', 'CS135 ', 'I129  ',
  'TC99  ', 'NI59  ', 'C14   ', 'SE79  ', 'NB94  ',
    246.0    238.0    245.0    241.0    237.0    243.0    239.0    240.0
234.0
    230.0    226.0    210.0    137.0    135.0    129.0     99.0     59.0
14.0
     79.0     94.0
      1        2        1        3        4        3        5        5
2
      6        7        8        9        9       10       11       12
13
     14       15
      0        1        0        1        2        0        1        0
0
      1        2        3        0        0        0        0        0
0
      0        0
      0        0        0        0        0        0        0        0
0
      0        0        0        0        0        0        0        0
0
      0        0
    0.0      1.0      0.0      1.0      1.0      0.0      1.0      0.0
0.0
    1.0      1.0      1.0      0.0      0.0      0.0      0.0      0.0
0.0
    0.0      0.0
    0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
0.0
    0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
0.0
    0.0      0.0
0.258E-01  0.318E+00  0.126E+00  0.164E+04  0.288E+00  0.155E+02  0.308E+03
0.508E+03  0.189E+01
0.129E-03  0.367E-06  0.471E-07  0.766E+05  0.350E+00  0.295E-01  0.123E+02
0.356E+01  0.100E-26
```

```
0.381E+00  0.793E+00
0.473E+04  0.447E+10  0.850E+04  0.432E+03  0.214E+07  0.738E+04  0.241E+05
0.654E+04  0.245E+06
0.770E+05  0.160E+04  0.223E+02  0.300E+02  0.230E+07  0.157E+08  0.213E+06
0.800E+05  0.573E+04
0.650E+05  0.203E+05
```

3.3.12 Description of the VOLCANO Parameter Map File

Location of sampled parameters used by VOLCANO module is indicated in this file. The index (second column) is the location of the first value of an array and the count (third column) provides the number of values in that array.

Line 1: title
Line 2: header
Line 3: event time pointer
Line 4-11: u1 through u8 pointers

EXAMPLE OF VOLMAP.DAT FILE

```
VOLCANO MAP FILE
VAR     INDEX    COUNT
time      344      1
u1        345      1
u2        346      1
u3        347      1
u4        348      1
u5        349      1
u6        350      1
u7        351      1
u8        352      1
```

3.3.13 Description of the VOLCANO Global Data Temporary Input File (TPA_VOL.VGD)

Global data for the VOLCANO module is contained in this temporary file created by the TPA.

Line 1: title
Line 2: LHS output file name
Line 3: LHS map file for VOLCANO parameters
Line 4: current vector number
Line 5: simulation stop time
Line 6: invocation flag
Line 7: waste inventory for repository
Line 8: number of zones in repository

Line 9: number of canisters in each zone
Line 10: number of chains
Line 11: number of nuclides
Line 12: number of elements
Group 1: number of nuclides per chain
Group 2: nuclide names
Group 3: nuclide atomic mass
Group 4: nuclide element numbers
Group 5: nuclide parent 1 position in chain
Group 6: nuclide parent 2 position in chain
Group 7: parent 1 branching fraction
Group 8: parent 2 branching fraction
Group 9: nuclide inventory
Group 10:nuclide half-life

EXAMPLE OF TPA_VOL.VGD FILE

```
TITLE: TPA temporary file for VOLCANO global parameters.
/oooo/lhsoooo.out
/volcano/volmap.dat
      400
  10080.0
        1
     70.0
        7
     2335      6150      4875      3675      1275      5625      1073
       13
       20
       15
        2         3         2         1         4         1         1         1
        1         1         1         1         1
  'CM246 ', 'U238  ', 'CM245 ', 'AM241 ', 'NP237 ',
  'AM243 ', 'PU239 ', 'PU240 ', 'U234  ', 'TH230 ',
  'RA226 ', 'PB210 ', 'CS137 ', 'CS135 ', 'I129  ',
  'TC99  ', 'NI59  ', 'C14   ', 'SE79  ', 'NB94  ',
    246.0     238.0     245.0     241.0     237.0     243.0     239.0     240.0
234.0
    230.0     226.0     210.0     137.0     135.0     129.0      99.0      59.0
14.0
     79.0      94.0
      1         2         1         3         4         3         5         5
2
      6         7         8         9         9        10        11        12
13
     14        15
      0         1         0         1         2         0         1         0
0
      1         2         3         0         0         0         0         0
0
      0         0
```

```
      0         0         0         0         0         0         0         0
0
      0         0         0         0         0         0         0         0
0
      0         0
      0.0       1.0       0.0       1.0       1.0       0.0       1.0       0.0
0.0
      1.0       1.0       1.0       0.0       0.0       0.0       0.0       0.0
0.0
      0.0       0.0
      0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
0.0
      0.0       0.0       0.0       0.0       0.0       0.0       0.0       0.0
0.0
      0.0       0.0
0.258E-01 0.318E+00 0.126E+00 0.164E+04 0.288E+00 0.155E+02 0.308E+03
0.508E+03 0.189E+01
0.129E-03 0.367E-06 0.471E-07 0.766E+05 0.350E+00 0.295E-01 0.123E+02
0.356E+01  0.154E+01
0.381E+00 0.793E+00 0.473E+04  0.447E+10  0.850E+04  0.432E+03  0.214E+07
0.738E+04 0.241E+05 0.654E+04 0.245E+06
0.770E+05 0.160E+04 0.223E+02 0.300E+02 0.230E+07 0.157E+08 0.213E+06
0.800E+05  0.573E+04
0.650E+05 0.203E+05
```

3.3.14 Description of the DRILLO1 Parameter Map File

Location of sampled parameters specific to DRILLO1 are contained in this file. In the example below, in the vector of values created by LHS, the 353rd value belongs to Nbor which is an array of size one.

Line 1: title
Line 2: header
Line 3: nbor pointer
Line 4: radius pointer
Line 5: regn pointer
Line 6: td pointer
Line 7: hit pointer

EXAMPLE OF DR1MAP.DAT FILE

```
DRILLO1 MAP FILE
VAR      INDEX    COUNT
Nbor       353      1
radius     354      1
Regn       355      30
Td         385      30
hit        415      30
```

3.3.15 Description of the DRILLO1 Global Data Temporary Input File (TPA_DR1.CGD)

Global data for DRILLO1 is read from this file which is created by the TPA.

Line 1: title
Line 2: LHS output file
Line 3: LHS map file for DRILLO1
Line 4: current vector number
Line 5: simulation stop time
Line 6: number of repository zones
Line 7: canister radius
Line 8: number of canisters in each zone

EXAMPLE OF TPA_DR1.DGD FILE

```
TITLE: TPA temporary file for DRILLO1 global parameters.
/oooo/lhsoooo.out
/drill/dr1map.dat
      400
  10080.0
        7
   0.3300000E+00
 2335,  6150,  4875,  3675,  1275,  5625,  1073
```

3.3.16 Description of the DRILLO2 Global Data Temporary Input File (TPA_DR2.CGD)

This temporary file created by the TPA contains data for use by the DRILLO2 module.

Line 1: title
Line 2: file name for NEFTRAN input from SOTEC
Line 3: file name for input from NEFTRAN
Line 4: file name for FLOWMOD groundwater travel time output
Line 5: file name for output to AIRCOM
Line 6: current vector number
Line 7: simulation stop time
Line 8: canister length
Line 9: total number of canisters
Line 10: waste inventory for repository
Line 11: number of chains
Group 1: number of nuclides per chain
Group 2: nuclide names
Group 3: nuclide inventory

Group 4: nuclide half-life

EXAMPLE OF TPA_DR2.DGD FILE

```
TITLE: TPA temporary file for DRILLO2 global parameters.
sotnef.dat
nefdr2.dat
flotpa.dat
dr2air.dat
      400
  10080.0
  0.4700000E+01
  25008.0
     70.0
       13
        2         3         2         1         4         1         1         1
        1         1         1         1         1
  'CM246 ', 'U238  ', 'CM245 ', 'AM241 ', 'NP237 ',
  'AM243 ', 'PU239 ', 'PU240 ', 'U234  ', 'TH230 ',
  'RA226 ', 'PB210 ', 'CS137 ', 'CS135 ', 'I129  ',
  'TC99  ', 'NI59  ', 'C14   ', 'SE79  ', 'NB94  ',
  0.258E-01  0.318E+00  0.126E+00  0.164E+04  0.288E+00  0.155E+02  0.308E+03
0.508E+03 0.189E+01
0.129E-03 0.367E-06 0.471E-07 0.766E+05 0.350E+00 0.295E-01 0.123E+02
0.356E+01  0.154E+01
0.381E+00 0.793E+00
0.473E+04 0.447E+10 0.850E+04 0.432E+03 0.214E+07 0.738E+04 0.241E+05
0.654E+04  0.245E+06
0.770E+05 0.160E+04 0.223E+02 0.300E+02 0.230E+07 0.157E+08 0.213E+06
0.800E+05  0.573E+04
0.650E+05 0.203E+05
```

3.3.17 Description of the SEISMO Global Data Temporary Input File (TPA_SEI.SGD)

This file contains global data for the SEISMO module.

Line 1: title
Line 2: SOTEC output file for container wall thickness
Line 3: file name for output to SOTEC
Line 4: current vector number
Line 5: simulation stop time
Line 6: canister length
Line 7: canister radius
Line 8: canister wall thickness
Line 9: total number of canisters
Line 10: waste inventory for repository

EXAMPLE OF TPA_SEI.SGD FILE

```
TITLE: TPA temporary file for SEISMO global parameters.
/usr/tmp/zke/oooo20n400v/sotsei.dat
seisot.dat
      400
  10080.0
  0.4700000E+01
  0.3300000E+00
  0.1000000E-01
  0.2500800E+05
  0.7000000E+02
```

3.3.18 Description of the AIRCOM Global Data Temporary Input File (TPA_AIR.SGD)

Global data for the AIRCOM module is contained in this file.

Line 1: title
Line 2: file name for input from C14
Line 3: file name for input from DRILLO2
Line 4: file name for input from VOLCANO
Line 5: file name for AIRCOM standard input
Line 6: current vector number
Line 7: scenario 4 character name

EXAMPLE OF TPA_AIR.AGD FILE

```
TITLE: TPA temporary file for AIRCOM global parameters.
c14air.dat
dr2air.dat
volair.dat
airrel.in
      400
CSDV
```

3.3.19 Description of the DITTY Global Data Temporary Input File (TPA_DIT.CGD)

Global data for the DITTY module is contained in this temporary file.

Line 1: title
Line 2: file name for DITTY standard input
Line 3: file name for input from AIRCOM
Line 4: not used
Line 5: file name for population data

Line 6: file name for wind data
Line 7: file name for DITTY standard output
Line 8: file name for plot data
Line 9: file name for dose output data for TPA executive
Line 10: not used
Line 11: not used
Line 12: current vector number
Line 13: number of nuclides
Group 1: nuclide names
Line 14: simulation start time
Line 15: simulation stop time
Line 16: 'iwat' flag
Line 17: 'iair' flag
Line 18: 'ipath' flag
Line 19: 'luw' flag
Line 20: 'lua' flag
Line 21: 'ipopl' value
Line 22: 'ipop' value

EXAMPLE OF TPA_DIT.DGD FILE

```
TPA temporary file for DITTY_AIR global parameters.
/ditty/ditapoc.in
airrel.in

/ditty/poppop.in
/ditty/jointfre.in
dittya.out
dittec.plt
dittpa.dat
ditmap.dat
/oooo/lhsoooo.out
      400
       20
CM246
U 238
CM245
AM241
NP237
AM243
PU239
PU240
U 234
TH230
RA226
PB210
CS137
CS135
I 129
TC99
```

```
NI59
C 14
SE79
NB94
        0
    10080
        0
        2
        1
        0
        2
        0
        2
```

3.3.20 Description of the Reusable Scenario Nuclide Release Data File

The reusable scenario data file is created by the TPA program and is not supplied by the analyst. However, the file is not deleted by the TPA system and may be moved to another directory for use in a later run. The file contains the portion of the CUMREL internal data array that applies to the current scenario. This array contains the nuclide release data from all of the vectors, CMs, zones (cells), pathways (forms), and nuclides. The file is complete at the end of the scenario and may be reused immediately in the next scenario. When the file is reused it must match the number of nuclides, pathways, zones, CMs, and number of input vectors of the scenario. The data is not formatted with headers in the file since it is for internal use only and is a straight dump of the data array.

EXAMPLE OF A REUSABLE SCENARIO FILE

```
 TPA Version 2.0 on 04/05/93  at 13:57:00
  4/5 oodo201k run vector 153 only.

 TPA scenario cumulative release data file.
 Nuclides Pathways   Areas   Modules Vectors
       20        3       7       9       1
        1 vector
    0.000000E+00   0.000000E+00   8.505572E-02   0.000000E+00   0.000000E+00
    0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
    0.000000E+00   0.000000E+00   0.000000E+00   7.518714E+01   0.000000E+00
    0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
    0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
    0.000000E+00   0.000000E+00   0.000000E+00   3.900285E-02   0.000000E+00
    0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   1.286243E-01
    0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
    0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
    1.122957E+02   0.000000E+00   0.000000E+00   0.000000E+00   0.000000E+00
      .
      .
      .
```

3.3.21 Description of the Standard TPA Output File (TPA.OUT)

This formatted file reproduces the input contained in the scenario and calculation sections of the TPA Exec input file. Additionally, the file contains output requested by the user using the keyword calculation OUTPUT. During the execution of the TPA, radionuclide releases are calculated for each radionuclide in three phases (as solute dissolved in water, as gas, and as exposed material on the surface — sometimes these phases are referred to as liquid, gas, and direct pathways, respectively) for each of the input vectors sampled from their statistical distributions. Thus, radionuclide releases are functions of properties of the radionuclides, pathways, input vectors, and scenarios. While the CCDF, which is the sum of releases over all the radionuclides through all paths over all of the scenarios, is the final output of interest, the intermediate outputs may also be of some interest. The TPA Exec is designed to provide these intermediate outputs in a tabular form, if desired, by the user. The specific outputs to be printed by the TPA Exec can be selected by choosing any number of keywords from: CCDF, SCENREL, NUCREL, PATHREL, and CUMREL. Any keyword may be truncated to six characters in actual usage. The meaning of these terms is explained in the following.

The CCDF keyword causes the summation of normalized radionuclide releases for all of the scenarios. This summation is obtained by weighting the probability of release from an individual scenario with its probability of occurrence, that is,

$$P\{R \le r\} = \sum_{i=1}^{N} P\{R_i \le r | S_i\} \; P\{S_i\} \; . \qquad (3\text{-}1)$$

The output for the CCDF is printed in ascending order of probabilities with their respective release values.

The SCENREL keyword prints releases for each input vector of a scenario. The release from the individual radionuclides is added and normalized by using the allowable limits EPA. In the output produced, these releases are arranged in ascending order with their corresponding probabilities of occurrence.

The NUCREL data set is the total normalized radionuclide release from all paths, areas, and CM for each nuclide, vector, and scenario.

The PATHREL data set is the total normalized radionuclide release from all nuclides, areas, and CM for each path, vector, and scenario.

The CUMREL data set contains the radionuclide releases generated by the CM and normalized using the EPA release limits for a repository with an initial waste inventory of 70,000 MTHM. These results are arranged by scenario, vector, pathway (form), CM, and radionuclide.

Other information may also be displayed depending on the settings of the xUNIT variables in the TPA source code. For example, if EUNIT=6, then the error messages generated by TPA will be listed in the TPA standard output file.

3.3.22 Description of the TPA Exec Log File (TPA.LOG)

This formatted file contains a log of the major events during the execution of TPA. The first line of this file contains the date and title. The remaining lines contain the time and a message describing the event. Events such as the opening of a file and the starting of a CM are included.

EXAMPLE OF TPA Exec LOG FILE

```
17-JUN-91 TPA version 2.0 process log, CNWRA:05/27/91
17:25:33 TPA output file opened with name on next line.
17:25:33 USER4:[RJANETZKE.TPA2]TPA.OUT;27
17:25:33 TPA input file opened with name on next line.
17:25:33 USER4:[RJANETZKE.TPA2]TPA.INP;12
17:25:34 Analyst is Ron Janetzke
17:25:34 Scenario name: Name_1
17:25:34 Scenario cumulative release file name is name_1.cum

17:25:34 LHS input file is [rjanetzke.tpa]lhs.inp
17:25:34 LHS output file is lhs.oul

17:25:34 NEFTRAN input file is [rjanetzke.tpa]nefii.inp

17:25:34 NEFTRAN map file is [rjanetzke.tpa]nefmap.dat

17:25:34 NEFTRAN output file is nefii.out

17:25:34 NEFTRAN release file is nefil.rel

17:25:34 NEFTRAN discharge file is nefii.dis

17:25:34 NEFTRAN source file is nefii.src

17:27:45 NEFTRAN release data file is on next line.
17:27:45 nefil.rel
17:27:46 Scenario CCDF file name is TPAS01.CDF

17:27:46 File on next line is opened for scenario CDF file.
17:27:46 USER4:[RJANETZKE.TPA2]TPAS01.CDF;1
17:27:46 File on next line is opened for cumrel file.
17:27:46 USER4:[RJANETZKE.TPA2]NAME_1.CUM;1
17:27:47 Scenario name: Name_2
17:27:47 Scenario cumulative release file name is name_2.cum

17:27:47 LHS input file is [rjanetzke.tpa]lhs.inp

17:27:47 LHS output file is lhs.ou2

17:27:47 NEFTRAN input file is [rjanetzke.tpa]nefii.inp
```

```
17:27:47 NEFTRAN map file is [rjanetzke.tpa]nefmap.dat

17:27:47 NEFTRAN output file is nefii.out

17:27:47 NEFTRAN release file is nefil.rel

17:27:47 NEFTRAN discharge file is nefii.dis

17:27:47 NEFTRAN source file is nefii.src

17:27:47 Reusing NEFTRAN files this scenario.
17:28:04 NEFTRAN release data file is on next line.
17:28:04 nefil.rel
17:28:04 Scenario CCDF file name is TPAS02.CDF

17:28:04 File on next line is opened for scenario CDF file.
17:28:04 USER4:[RJANETZKE.TPA2]TPAS02.CDF;1
17:28:05 File on next line is opened for cumrel file.
17:28:05 USER4:[RJANETZKE.TPA2]NAME_2.CUM;1
17:28:05 Scenario name: Name_3
17:28:05 Scenario cumulative release file name is name_1.cum

17:28:05 LHS input file is [rjanetzke.tpa]lhs.inp

17:28:05 LHS output file is lhs.ou3

17:28:05 NEFTRAN input file is [rjanetzke.tpa]nefii.inp

17:28:05 NEFTRAN map file is [rjanetzke.tpa]nefmap.dat

17:28:05 NEFTRAN output file is nefii.out

17:28:05 NEFTRAN release file is nefs3.rel

17:28:05 NEFTRAN discharge file is nefii.dis

17:28:05 NEFTRAN source file is nefii.src

17:28:05 File on next line is opened for scenario cumulative release reuse.
17:28:05 USER4:[RJANETZKE.TPA2]NAME_1.CUM;1
17:28:21 Normal TPA system exit.
```

3.3.23 Description of the TPA Exec Error Message File (TPA.ERR)

This formatted file is normally empty. If the TPA system fails for some reason, there may be one or two messages in this file that would reflect the possible trouble area. See Appendix B for error messages.

3.3.24 Description of the Scenario CCDF Plot File

This file contains calculated data on cumulative releases for each scenario. This file is specially formatted for use with the TECPLOT graphics utility (TECPLOT, 1990). See the TECPLOT user's guide for details of the contents.

EXAMPLE OF SCENARIO CCDF PLOT FILE

```
 TITLE="Scenario CCDF for                                    TPA test run.
"
 VARIABLES=Entry,Relea,Proba,Compl
 ZONE T="CCDF",F=POINT
    1      7.0556E+00   2.5000E-01   1.0000E+00
    2      6.2449E+01   2.5000E-01   7.5000E-01
    3      9.1399E+01   2.5000E-01   5.0000E-01
    4      1.0301E+02   2.5000E-01   2.5000E-01
```

3.3.25 Description of the TPA CCDF Plot File

This file contains net cumulative releases from all scenarios and can be used to plot the final CCDF. Again, this file is formatted to meet the requirements of the TECPLOT graphics utility. See the TECPLOT user's guide for details about the format of the file.

EXAMPLE OF REPOSITORY CCDF PLOT FILE

```
TITLE="CCDF for  TPA scenario input file.
 VARIABLES=Entry,Relea,Proba,Compl
 ZONE T="CCDF",F=POINT
      1  0.2685E+02   0.1980E+00   0.1000E+01
      2  0.6961E+02   0.1980E+00   0.8020E+00
      3  0.7531E+02   0.1980E+00   0.6040E+00
      4  0.9482E+02   0.1980E+00   0.4060E+00
      5  0.1125E+03   0.6600E-01   0.2080E+00
      6  0.1125E+03   0.1320E+00   0.1420E+00
```

3.3.26 Software Limitations

The TPA system was originally designed to run on any platform supporting standard FORTRAN 77. The TPA code for the VAX/VMS and CRAY/UNICOS systems was implemented successfully in Phase 2. Later phases will incorporate additional platforms under the design goals. However, due to the large number of authors working independently on the consequence modules, not all of the subprocess modules run on both the VAX and CRAY. Specifically the SEISMO module will only run on the CRAY due to file names that are hard-coded. Normally this would not be a problem, but in this particular case the UNIX specific path is required with the file name. This limitation will be removed in later versions.

DRILLO1 also has a hard-coded file name for the FLOWMOD input file (flowmod.inp), but this works for both the VAX and the CRAY.

When the TPA system is invoked, it assumes the existence of various input files and the absence of critical files that are not to be overwritten. The analyst should move all critical output files to another directory before running the TPA system since the TPA Exec will halt if it attempts to overwrite them during execution. The files assumed to be critical by TPA Exec are, 'tpa.log', 'tpa.out', 'flotpa.dat', 'sotsei.dat', 'neftpa.dat', 'maxrel.dat', 'maxc14.dat', '*_TLY.DAT', and 'SCN*'. A file assumed critical by the LHS module is 'lhsoooo.out'. A file assumed critical by SOTEC is 'confail.out'.

Several quality control checks are appropriate before making a TPA run. They are:

- Check the MAXNUC, MAXVEC, and MAXSCN parameters in the TPA source code for their ability to accommodate the desired problem size
- Check the saturated leg lengths of the FLOWMOD input file; they should correspond to the leg lengths in the TPA WATER keyword input values
- Check the infiltration pointer in the FLOWMOD map file for appropriate climatic scenario values
- Check the inventories in the TPA input file, they should all be nonzero
- Check the file name for the data file that transfers data from SOTEC to SEISMO; this file should exist for all seismic scenarios, but is not needed for other disruptive scenarios, and should be absent for the undisturbed scenario
- Check the element ordering for the kd's, retardation, and solubilities in the LHS input file, they should be consistent with the TPA input file and the FLOWMOD input file
- Check the scenario probability in the TPA input file for appropriate value for a given scenario
- Check the nuclide names and ordering for consistency in both the TPA input file and the FLOWMOD input file

3.3.27 Hardware Requirements and Installation Procedures

The "cumrel" is the largest array in the TPA Exec and will determine the memory requirements during execution, at least 8 megawords should be requested on the INEL CRAY when running a 400-vector job. CRAY execution time is estimated to be 1.5 minutes for an undisturbed case vector and 2.0 minutes for a fully disturbed case vector.

For space considerations the TPA unit 6 output and the 'cumrel.fmt' file should be written to a temporary disk.

Typically, each consequence module is stored in its own subdirectory, with a separate subdirectory for the Exec. The primary source files are maintained as FORTRAN preprocessor input files. The preprocessor used is called preFOR and is discussed in section 3.4.1. Once the preFOR files are copied to disk an operating system procedure can be written to automatically generate the executable files. Each module will then have an executable file in its subdirectory. These are the modules that will be spawned by the TPA Exec during execution.

3.3.28 User Support

For technical assistance contact:

Ron Janetzke
Center for Nuclear Waste Regulatory Analyses
Southwest Research Institute
P. O. Drawer 28510
San Antonio, TX 78228-0510
e-mail: rjanetzke@swri.edu
(210) 522-3318

3.3.29 Code Structure

Table 3-1 gives a brief description of each subroutine for the TPA Exec. A Calling Tree diagram is presented in Figure 3-3.

Table 3-1. Listing of subroutines and their functions

Subroutine	Function
airexc	Creates a file of global parameters and transfers control to the AIRCOM subprocess.
blocks	Generates block letters for the first 18 letters of the input string, and writes them to the FORTRAN logical unit given in the first argument.
c14exc	Creates a file of global parameters and transfers control to the C14H subprocess.
c14rd	Reads and normalizes the C14 release file produced by C14H.
calcrd	Reads the calculation section keyword input for the main TPA module.

Subroutine	Function
center	Centers an input string in an output string with blanks on both sides.
cnsqnc	Calls the various executive routines for the consequence modules for the current vector.
combin	Combines elements of the scenario normalized release array into an array of unique valued elements with their corresponding probabilities.
comprd	Compresses the dose values for each scenario in the 'dossum' array.
compre	Compresses the release values for each scenario in the 'epasum' array.
comprs	Compresses a character string by compressing all successive blanks to the maximum allowed as specified by the argument list.
diaexc	Creates a file of global parameters for air dose and transfers control to the DITTY subprocess.
difexc	Creates a file of global parameters for food dose and transfers control to the DITTY subprocess.
diwexc	Creates a file of global parameters for water dose and transfers control to the DITTY subprocess.
dmpfil	Dumps an ASCII file to the supplied FORTRAN logical unit.
dosrd	Reads the dose files sent to TPA by DITTY.
dr1exc	Creates a file of global parameters and transfers control to the DRILLO1 subprocess.
dr2exc	Creates a file of global parameters and transfers control to the DRILLO2 subprocess.
dr2rd	Reads and normalizes the nuclide direct release file produced by DRILLO2.
dtcomp	Compresses the total system dose arrays.
epanrm	Normalizes a release value for a particular nuclide with respect to the EPA limit for that nuclide.
epaprb	Fills the 'epasum' and 'dossum' arrays and their corresponding probability arrays 'prbsum' and 'dprbsm'.
errmsg	Writes the string argument to the error file.
esort	Sorts the 'epasum' array and its corresponding probability array.

Subroutine	Function
floexc	Creates a file of global parameters and transfers control to the FLOWMOD subprocess.
init	Initializes the 'cumrel' array to zero.
lhsexc	Sets up and controls the environment for the LHS subprocess.
logmsg	Writes the string argument with a time tag to the TPA log file.
nefexc	Creates a file of global parameters and transfers control to the NEFTRAN subprocess.
nefrd	Reads and normalizes the NEFTRAN release file produced by NEFTRAN.
opnfil	Opens a formatted file on the requested logical unit with the arguments supplied in the argument list.
opninp	Opens the TPA standard input file. This file contains free form input suitable for reading by routines in the RDFREE library.
opnlog	Opens the TPA log file. The log file keeps a log of the major events in the running of the TPA process.
opnout	Opens the TPA standard output file.
outfmt	Writes the requested intermediate arrays to formatted disk files.
output	Writes the requested intermediate arrays to the output file.
paghdr	Outputs a form feed for a new page and prints a header with the run title.
pagttl	Outputs a page title for a new page.
paring	Puts both Exec and RDFREE code PARAMETER items on the second title page.
rdinhd	Reads the header section keyword input from the standard input file.
scdfwr	Opens and writes the scenario CCDF file.
scninp	Reads the keyword input for the main TPA module.
scumrd	Opens and reads the scenario cumulative release file.
scumwr	Opens and writes the scenario cumulative release file.
seiexc	Creates a file of global parameters and transfers control to the SEISMO subprocess.

Subroutine	Function
shsort	Performs a shell sort of the input array into ascending numerical order.
sotexc	Creates a file of global parameters and transfers control to the SOTEC subprocess.
split	Splits one part nuclide names into nuclide names with two parts as required by DITTY.
sums	Totals the appropriate columns and rows of the 'cumrel' array and stores them in intermediate arrays.
tcompr	Compresses the total system release arrays.
temexc	Sets up and controls the environment for the TEMPERATURE subprocess.
titpag	Puts the program title page on the output device with block letters.
tpa	The main driver for the TPA system.
tsdf	Fills the total system release arrays and the total system dose arrays and their corresponding probability arrays.
tsort	Sorts the total system release arrays and the total system dose arrays in ascending order of release.
volexc	Creates a file of global parameters and transfers control to the VOLCANO subprocess.
volrd	Reads and normalizes the nuclide direct release file produced by VOLCANO.

3.4 HINTS FOR CODE MAINTENANCE

Hints for code maintenance are provided in the following.

3.4.1 FORTRAN Preprocessor (preFOR)

Program preFOR (Janetzke and Sagar, 1991a) is written in standard FORTRAN 77 language. It is a preprocessor for FORTRAN codes, and its use can provide some flexibility in developing such codes. Included in preFOR are various utility modules to find the length of strings, fix tabs, perform I/O, etc. There is only one parameter in this code that an individual user may have to change: the parameter MXCOM defines the maximum number of temporary files allowed to be generated by preFOR and may be changed to something other than 250 (default). This parameter declares the array space for the temporary file names of the code fragments. If the number of code fragments exceeds MXCOM, the parameter should be increased to match the number of fragments.

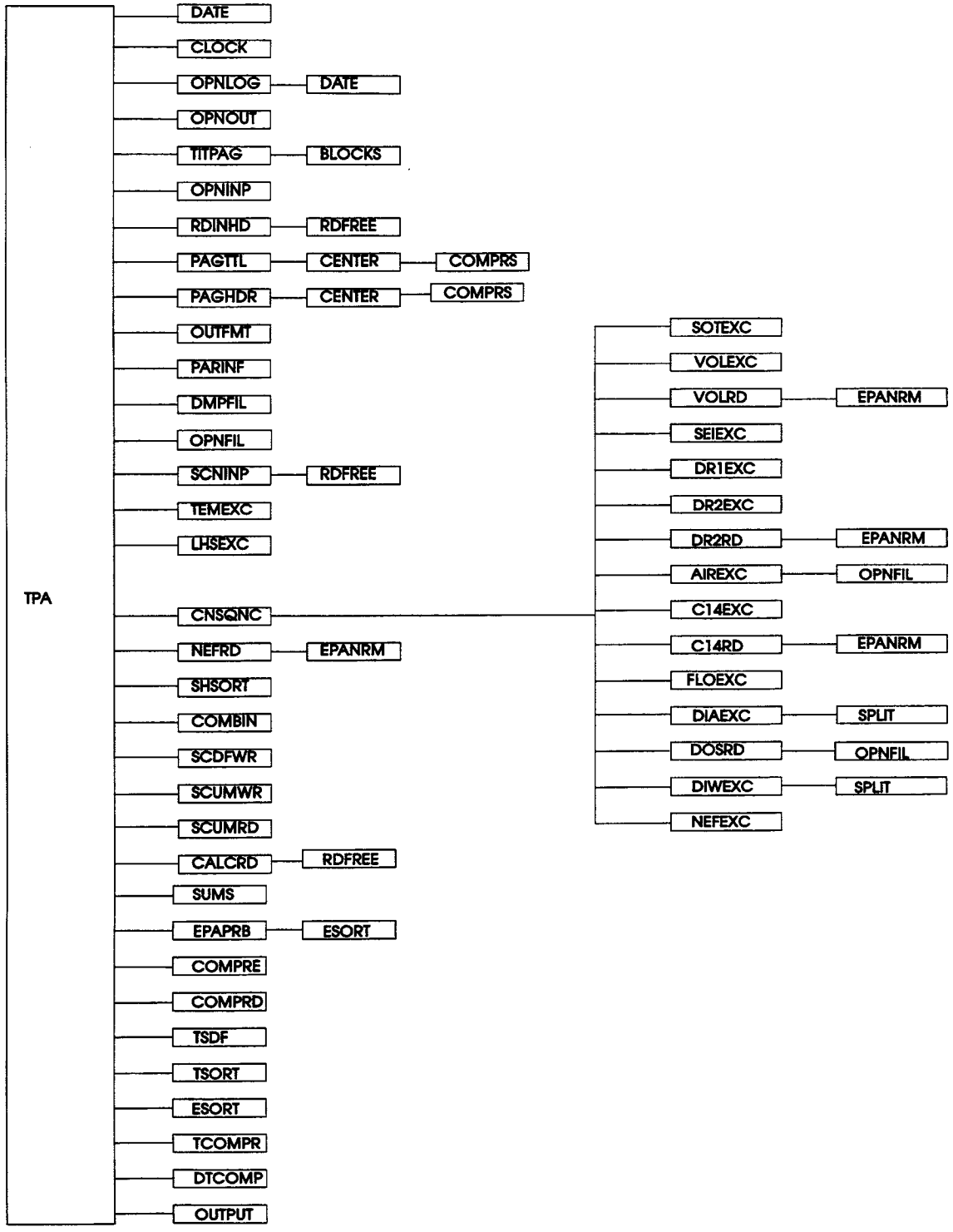

Figure 3-3. Calling sequence of subroutines in TPA

When invoked, preFOR will prompt for an input file name. The input file is a source code in which preFOR commands have been embedded. The program will then prompt for an output file name. The output file produced by preFOR is a standard FORTRAN file that can be compiled. The preFOR commands are summarized following and details can be read in Janetzke and Sagar (1991a).

The source code for TPA Exec includes a number of the preFOR commands. Before compilation, therefore, the preFOR utility should be invoked to produce a standard FORTRAN file.

3.5 COMPILATION AND LINKING OF THE TPA Exec CODE

The following commands can be used for compiling and linking the TPA Exec code on a VAX computer using the TPA.FOR and TPA.OPT files:

Command for Compilation
$ FORTRAN TPA
Command for Linking
$ LINK TPA/OPT

3.6 SYSTEM SPECIFIC CALLS IN TPA Exec

The TPA Exec computer code is configured for the VAX and CRAY computer system. The following are system calls used in the TPA system: (i) VAX/VMS — CALL DATE, CALL TIME, and CALL LIB$SPAWN; and (ii) CRAY — CALL DATE, CALL CLOCK, and I = ISHELL.

3.7 ARRAY SIZES IN TPA Exec

The size of the program is greatly affected by the size of the CUMREL array. This array has six dimensions: CUMREL (MAXNUC, MAXFRM, MAXARE, MAXMDL, MAXVEC, and MAXSCN), where,

MAXNUC = maximum number of nuclides, corresponds to the NN array in BLOCK DATA EPANRM

MAXFRM = maximum number of pathways (forms) of release

MAXARE = maximum number of areas (zones) in repository

MAXMDL = maximum number of CMs providing release values

MAXVEC = maximum number of vectors

MAXSCN = maximum number of scenarios

All of these may have a minimum setting of 1 except MAXNUC, which should remain at 20.

3.8 UNITS IN TPA

All units in TPA are metric [i.e., meter (length), kilogram (mass), and year (time) (MKY)], except for radiation, which is in terms of curies, and initial inventory, which is in kilo metric tons heavy metal (KMTHM).

3.9 CHANGING THE ENVIRONMENTAL PROTECTION AGENCY (EPA) LIMITS

The EPA release limits are defined in array RL in BLOCK DATA EPALIM. The values have units of curies/MTHM, and were obtained from the cumulative release limits in 40 CFR Part 191.

4 INPUT COMMANDS FOR TPA Exec

The input to TPA Exec is from a data file that is created using appropriate keywords. These keywords and the numeric and alpha data associated with them are explained in detail in this chapter. Section 3.8 discusses the appropriate units for input values.

The RDFREE library of subroutines is utilized to read and interpret the TPA Exec input data. RDFREE subroutines are designed to read all characters (alpha and numeric) in a record of any length, and analyze this record to assign input values to various parameters associated with the keyword of that record. Details about RDFREE are available in a separate document (Janetzke and Sagar, 1991b), and will not be repeated here. However, it should be mentioned that this form of input allows liberal use of comment lines and blank lines, as well as composing input in English-like constructs. The entry of numeric values in a record is free of any rigid format requirements.

4.1 DETAILED DESCRIPTION OF TPA KEYWORDS

This section defines the TPA keywords in alphabetical order. The keyword characters which appear in lower case on the following pages may be omitted, since the input routine only recognizes the first six characters. The modifiers are grouped under their respective keywords, also in alphabetical order. A detailed description of the syntax of the input mechanism is given in Appendix B. Note that all logical records must end with a special symbol which is the vertical bar (|).

The descriptions of keywords is given in five parts: PURPOSE, SYNTAX, MODIFIERS, COMMENTS, and EXAMPLE.

The PURPOSE part gives the section (either header, scenario, or calculation), in which the keyword may appear, and gives a functional description of the keyword indicating the action implied by its use.

The SYNTAX part shows the general form of the input record along with the position of any data fields. The keyword or modifier is given in uppercase and the generic data identification is given in lowercase, if present.

The MODIFIER part gives the meaning of the modifiers and any data associated with them. The data fields may be one of two types: (i) string; or (ii) numeric. The numeric data values can be input in any form (integer, fixed point, or exponent form), and any column following the keyword or modifier. String values must be enclosed in single quotes (').

The COMMENTS part gives further details of KEYWORD and data fields and hints on their usage.

The EXAMPLE part gives an example of a logical record that uses the keyword.

AIRCOM

PURPOSE This keyword of the scenario section schedules the AIRCOM subprocess for execution. This module accepts air release data from C14, VOLCANO, and DRILLO2 and reformats them for DITTY.

SYNTAX AIRCOM ¦

MODIFIERS None.

COMMENTS There are no data items for this keyword.

EXAMPLE `AIRCOM ¦`

ANALYSt

PURPOSE This header section keyword identifies the analyst associated with the current input file. As a required parameter, it enhances the quality assurance feature of the TPA system. This data item appears in the standard TPA printed output.

SYNTAX ANALYST s1 ¦

s1: Analyst name.

MODIFIERS None.

COMMENTS Only one string field is allowed, so include the first name, last name, and any initials in the string.

EXAMPLE `ANALYST 'Ron Janetzke' ¦`

C14

PURPOSE This scenario section keyword specifies the control parameters for the C14 subprocess.

SYNTAX C14 AIRCOM s1 DELTA n1 FIRST n2 LHSMAP s2 SOTEC s3 TEMPERATURE s4 VELOCITY s5 |

MODIFIERS AIRCOM

s1: The file name to be used for holding release for AIRCOM input.

DELTA

n1: The time step delta to be used for generating output values.

FIRST

n2: The time of the first time step.

LHSMAP

s2: The filename of the LHS output data set.

SOTEC

s3: The file name of the SOTEC output file to be used as input to C14.

TEMPERATURE

s4: The name of the file containing temperature data from a companion module called FDTEMP2A described in the C14 Documentation.

VELOCITY

s5: The name of the temporary file used for velocity data.

COMMENTS None.

C14 (Cont'd)

EXAMPLE

```
C14  lhsmap       = 'c14map.dat'
     temperature  = 'fdtemp2a.out'
     velocity     = 'c14h.vel'
     sotec        = 'sotc14.dat'
     aircom       = 'c14air.dat'
     first        0
     delta        50 |
```

DOSE_Air

PURPOSE This keyword of the scenario section specifies the control parameters for running the DITTY subprocess for dose values related to air release.

SYNTAX DOSE_AIR AIR s1 DATA s2 INPUT s3 OUTPUT s4 PLOT s5 POPULATION s6 RELEASE n1 STOP n2 WIND s7 ¦

MODIFIERS AIR

s1: The name of the data file from AIRCOM.

DATA

s2: The name of the file used to send release data to the TPA Exec.

INPUT

s3: The name of the standard input file for DITTY.

OUTPUT

s4: The name of the standard output file from DITTY.

PLOT

s5: The name of the plot file from DITTY.

POPULATION

s6: The name of the population input file for DITTY.

RELEASE

n1: The time of the release.

STOP

n2: The simulation stop time.

WIND

s7: The name of the wind data input file for DITTY.

DOSE_Air (Cont'd)

COMMENTS None.

EXAMPLE

```
DOSE_AIR
     input      = 'ditapoc.in'
     air        = 'airrel.in'
     population = 'poppop.in'
     wind       = 'jointfre.in'
     output     = 'dittya.out'
     plot       = 'dittec.plt'
     data       = 'dittpa.dat
     release    0
     stop       10080 ¦
```

DOSE_Food

PURPOSE This keyword of the scenario section specifies the control parameters for running the DITTY subprocess for dose values related to food consumption.

SYNTAX DOSE_FOOD DATA s1 INPUT s2 OUTPUT s3 PLOT s4 POPULATION s5 RELEASE n1 STOP n2 WATER s6 WIND s7 |

MODIFIERS DATA

s1: The name of the file used to send release data to the TPA Exec.

INPUT

s2: The name of the standard input file for DITTY.

OUTPUT

s3: The name of the standard output file from DITTY.

PLOT

s4: The name of the plot file from DITTY.

POPULATION

s5: The name of the population input file for DITTY.

RELEASE

n1: The time of the release.

STOP

n2: The simulation stop time.

WATER

s6: The name of the file for water release data.

WIND

s7: The name of the wind data input file for DITTY.

DOSE_Food (Cont'd)

COMMENTS None.

EXAMPLE

```
DOSE_FOOD
     input       = 'ditapoc.in'
     water       = 'watrel.in'
     population  = 'poppop.in'
     wind        = 'jointfre.in'
     output      = 'dittya.out'
     plot        = 'dittec.plt'
     data        = 'dittpa.dat
     release     0
     stop        10080 ¦
```

DOSE_Water

PURPOSE This keyword of the scenario section specifies the control parameters for running the DITTY subprocess for dose values related to water release.

SYNTAX DOSE_WATER DATA s1 INPUT s2 OUTPUT s3 PLOT s4 POPULATION s5 RELEASE n1 STOP n2 WATER s6 WIND s7 |

MODIFIERS DATA

s1: The name of the file used to send release data to the TPA Exec.

INPUT

s2: The name of the standard input file for DITTY.

OUTPUT

s3: The name of the standard output file from DITTY.

PLOT

s4: The name of the plot file from DITTY.

POPULATION

s5: The name of the population input file for DITTY.

RELEASE

n1: The time of the release.

STOP

n2: The simulation stop time.

WATER

s6: The name of the file for water release data.

WIND

s7: The name of the wind data input file for DITTY.

DOSE_Water (Cont'd)

COMMENTS None.

EXAMPLE

```
DOSE_WATER
     input       = 'ditapoc.in'
     water       = 'watrel.in'
     population  = 'poppop.in'
     wind        = 'jointfre.in'
     output      = 'dittya.out'
     plot        = 'dittec.plt'
     data        = 'dittpa.dat
     release     0
     stop        10080 ¦
```

DRILL1

PURPOSE This keyword of the scenario section schedules the DRILLO1 module for execution.

SYNTAX DRILL1 LHSMAP s1 ¦

MODIFIERS LHSMAP

s1: The name of the LHS output data file.

COMMENTS Consequences from drilling into the repository horizons are calculated using two modules. The location and timing of drilling and the possibility of hitting waste canisters is determined in DRILLO1. This information is passed to SOTEC, where it is used to compute the source term. The DRILLO2 module calculates consequences that result from bringing radioactive material to the surface.

EXAMPLE

```
DRILL1
     lhsmap = 'dr1map.dat' ¦
```

DRILL2

PURPOSE This keyword of the scenario section schedules the DRILLO2 module for execution.

SYNTAX DRILL2 AIRCOM s1 ¦

MODIFIERS AIRCOM

s1: The name of the output file for AIRCOM.

COMMENTS Consequences from drilling into the repository horizons are calculated using two modules. The location and timing of drilling and the possibility of hitting waste canisters is determined in DRILLO1. This information is passed to SOTEC where it is used to compute the source term. The DRILLO2 module calculates consequences that result from bringing radioactive material to the surface.

EXAMPLE

```
DRILL2
     aircom = 'dr2air.dat' ¦
```

END

PURPOSE This calculation section keyword signifies the end of the input data stream. When END is encountered in the input file, reading of input is stopped and calculations are begun. Therefore, it must be the last data record of the input file.

SYNTAX END ¦

MODIFIERS None.

COMMENTS If the END keyword is not the last record in the input file, the remainder of the file is ignored.

EXAMPLE END ¦

LHS

PURPOSE This scenario section keyword is used to specify the parameters that are to be passed to the LHS module. The data that is passed from the TPA Exec to the LHS generally consist of the names of the I/O files and the number of sample vectors to be generated.

SYNTAX LHS INPUT s1 OUTPUT s2 VECTORS n1 ¦

MODIFIERS INPUT

s1: The name of the LHS standard input file.

OUTPUT

s2: The name of the LHS output file with the sampled data.

VECTORS

n1: The number of vector data sets to generate. This value overrides the value for the NOBS parameter in the LHS standard input file.

COMMENTS Since the LHS parameters can affect all of the CMs, it should appear before them in the input file. Only one number should follow the keyword; any numbers appearing after the first number will be ignored. This number must not exceed MAXVEC because this is the FORTRAN parameter that controls the size of many of the internal data arrays. MAXVEC can be set to any value compatible with the host machine.

EXAMPLE LHS

```
vectors 50
input = 'lhsoooo.inp'
output = 'lhsoooo.out' ¦
```

NEFTRAn

PURPOSE This scenario section keyword is used to specify the parameters that TPA must pass to the NEFTRAN II subprocess. This keyword also indicates that the user wants the NEFTRAN II subprocess to be executed. Generally, the data passed to NEFTRAN II consist of names of various files, as indicated in the example below.

SYNTAX NEFTRAN DISCHARGE s1 INPUT s2 OUTPUT s3 RELEASE s4 SOURCE s5 ¦

MODIFIERS DISCHARGE

s1: The name of the discharge information output file. The discharge file contains the release value of each nuclide for each time step.

INPUT

s2: The name of the NEFTRAN standard input file prepared by FLOWMOD.

OUTPUT

s3: The name of the NEFTRAN standard output file.

RELEASE

s4: The name of the file of release data to be read by the TPA Exec.

SOURCE

s5: The name of the source information output file.

COMMENTS None.

EXAMPLE

```
NEFTRAN
     input      = 'flonef.dat'
     output     = 'nefrep.out'
     release    = 'neftpa.dat'
     discharge  = 'nefrep.dis'
     source     = 'nefrep.src' ¦
```

OUTPUT

PURPOSE This calculation section keyword specifies that formatted output files of internal arrays are to be generated by the TPA module. Several normalized release sum arrays will be printed. The output is directed to the standard TPA output file and printed in ASCII text. The main internal data array is the CUMREL array. It contains normalized nuclide release values (Ci) for each nuclide, consequence module, zone (area), pathway (liquid or gas), vector, and scenario. The other arrays are derived from it as sums over a given grouping of release values. A CCDF is also generated and is the total CCDF for the run.

SYNTAX OUTPUT CCDF CUMREL EPASUM NUCREL PATHRE SCENRE SCREEN |

MODIFIERS CCDF generates an ASCII file of the total CCDF for the run (ccdf.fmt).

CUMREL generates an ASCII file with the entire 'cumrel' array in a formatted listing (cumrel.fmt).

EPASUM generates an ASCII file with normalized release values summed by vector and sorted by ascending release value with probabilities assigned (epasum.fmt).

NUCREL generates an ASCII file with normalized release values summed from all paths, areas, and modules for each nuclide, vector, and scenario (nucrel.fmt).

PATHREL generates an ASCII file with normalized release values summed from all nuclides, areas, and modules, for each path, vector, and scenario (pathre.fmt).

SCENREL generates an ASCII file with normalized release values summed by vector (scnrel.fmt).

SCREEN schedules all information generated by the OUTPUT keyword to be displayed on unit 6 also.

COMMENTS CAUTION: Use of CUMREL and SCREEN can generate large files approaching 100 Megabytes each. All of the arrays that can be printed are sized by the TPA sizing parameters MAXSCN, MAXVEC, MAXFRM, MAXMDL, MAXARE, and MAXNUC. For most cases only the data that applies to a given scenario are printed, so the whole array is not printed. Even so the CUMREL array can be very large and generate many pages of output.

OUTPUT (Cont'd)

EXAMPLE

```
OUTPUT
      cumrel
      pathrel
      nucrel
      scenrel
      epasum
      ccdf ¦
```

RUN

PURPOSE This scenario section keyword starts the execution loop of all CMs scheduled for this scenario.

SYNTAX RUN ¦

MODIFIERS None.

COMMENTS The use of the RUN keyword signals the end of the specification of a given scenario. For single scenario runs the SCENARIO END command will usually follow immediately.

EXAMPLE RUN ¦

SCENARio

PURPOSE This scenario section keyword indicates the end of parameter input for a given scenario and requests the CMs to be executed in the order in which they were defined in the input file. This keyword introduces a logical record of modifiers that defines the parameters for a given scenario. Some of the modifiers define data values that are passed to some or all of the CMs. Scenario name and probability are specified as well as CCDF plot files and temporary storage files for the release sums are specified. There is the capability to use release data generated in a previous run rather than invoking the CMs.

SYNTAX SCENARIO n1 BRAN1 n2 BRAN2 n3 CANISTERS n4 CANLEN n5 CANRAD n6 CAN_WA n7 CCDF s1 CHAINS n8 ELEMENT n9 HALF_LIFE n10 ID_ELE n11 INVENT n12 LHS s2 MASS n13 MEMBER n14 NAME s3 NUCLIDE n15 NUCNAME s4 OUTPUT s5 PAR1 n16 PAR2 n17 PROBABILITY n18 START n19 STOP n20 VECTORS n21 WASTE n22 |

OR:

SCENARIO REUSE s6 PROBABILITY n18 |

OR:

SCENARIO END |

n1: Scenario number. This number indicates the position the results will be placed in any data arrays that are indexed by scenario.

MODIFIERS BRAN1

n2: Nuclide branching fractions from parent 1. One value for each nuclide. Data for this modifier is linked intimately with the CHAIN modifier. It is assumed that a daughter radionuclide may form from at most two parents. The fraction formed from the first parent is read by the modifier BRAN1; that formed by the second parent is read by BRAN2. The first member of a chain should always be assigned a value of 0.

BRAN2

n3: Nuclide branching fractions form parent 2. One value for each nuclide.

SCENARio (Cont'd)

CANISTERS

n4: Canisters per zone. One value for each zone. The number of waste canisters in various cells of the repository is read through the use of this modifier. For purposes of modeling, the repository is conceptualized as made of regions or cells. Conditions affecting the canisters within a cell are assumed to be uniform while varying from cell to cell. The number of cells can range from 1 to maxare. Maxare is a dimensioning parameter of TPA Exec whose value is currently set to 7.

CANLEN

n5: Canister length.

CANRAD

n6: Canister radius.

CAN_WA

n7: Canister wall thickness.

CCDF

s1: Name of the file to be used for the scenario CCDF plot data. The CCDF generated is for the scenario only and not for the entire run. This modifier must be present for each scenario for which a plot file is desired.

CHAINS

n8: The number of radionuclide chains to be considered in processing the scenario is specified by this modifier. Single radionuclides with no parent and daughter are considered a chain with one member.

ELEMENT

n9: The number of chemically distinct elements to be considered in processing the scenario is specified by this modifier. Note that an element may have several radioisotopes whose names are read by the use of keyword NUCLIDES.

SCENARio (Cont'd)

END This modifier is used alone with no others. It terminates the scenario section of the TPA input file and must appear after all other scenario input commands. The calculation section is entered immediately upon the receipt of the END modifier for the SCENARIO keyword.

HALF_LIFE

n10: Half-life of nuclides in nuclide list.

ID_ELE

n11: Nuclide element number referencing the order in which they appear in the nuclide list. For each nuclide name specified by NUCNAME, an element identification number is associated. For example, if the first four nuclides specified by NUCNAME are CM246, PU242, U238, U234, then the corresponding ID-ELEMENT data will be 1, 2, 3, 3. That is the first element is CM, the second, PU, and the third U. Note that the total number of elements is specified through the use of modifier ELEMENTS.

INVENT

n12: Nuclide inventories. The initial inventory of the radionuclides is provided by this modifier. Inventory is assumed to be in curie per MTHM in waste. Inventories are specified in the same order as the nuclide names in the modifier NUCNAME.

LHS

s2: Name of the LHS output file that holds the sampled data for this scenario.

MASS

n13: Nuclide atomic mass. The atomic mass is normally a part of the nuclide name, for example, U235 (Uranium with atomic mass of 235). Thus the atomic mass can be obtained from the name; this modifier will, therefore, be removed at a later date.

SCENARio (Cont'd)

MEMBER

n14: Number of members for each chain. Data of this modifier are linked intimately with the data of the CHAIN modifier. The number of members in a chain is specified with this modifier. Note that if a radionuclide is being considered by itself (i.e., daughter products are being omitted), then the number of members in this chain should be read as 1. Number of members must be specified for all chains, including for those chains that have only one member.

NAME

s3: Four character designation for this scenario. The order of the disruptive events represented in the name is climate, seismic, drilling, and volcano (e.g., CSDV). Omitted letters are replaced with an O. All letters are upper case only.

NUCLIDE

n15: Number of nuclides in nuclide name list.

NUCNAME

s4: Nuclide name list. This modifier reads the name of radionuclides for processing during the scenario. The names are passed to various modules including FLOWMOD, VOLCANO, and SOTEC via temporary files which are created by the TPA Exec. These are global parameters and may be used by other CMs as well. The number of nuclides entered should not exceed the TPA FORTRAN parameter 'maxnuc.'

OUTPUT

s5: This modifier identifies the name of the file to be used for storing the CUMREL array for the scenario. The CUMREL array is a large array that contains the release values for each nuclide, pathway, consequence module, and vector of the scenario. This file may then be used by another scenario to present the identical data to the TPA system without executing any of the code for the scenario.

PAR1

n16: Index in chain of parent 1.

SCENARio (Cont'd)

PAR2

n17: Index in chain of parent 2.

PROBABILITY

n18: Scenario probability. If a scenario CCDF is requested, this should be 1.

REUSE

s6: This modifier identifies a file that contains the scenario release data from a previously executed scenario. The scenario may be from this run or another TPA run. The release data is read from the file and used in the total CCDF calculation for this TPA run. The only other modifier that has any meaning when used with REUSE is PROBABILITY. All others are ignored.

START

n19: Simulation start time.

STOP

n20: Simulation stop time.

VECTORS

n21: Number of vectors to run for this scenario.

WASTE

n22: Total repository inventory in kilo metric tons heavy metal (KMTHM).

COMMENTS The END modifier is used on a mutually exclusive basis with all of the other modifiers.

In Version 2.0 of TPA Exec, canister length is assumed to be the same in all cells of the repository.

Inventories are later converted to curies per canister in some of the modules such as SOTEC or into mass (kg) per container as in SEISMO.

SCENARio (Cont'd)

EXAMPLE

```
scenario 1     \ Begin scenario.
         name 'CSDV'\ Scenario name.
         output   '/usr/tmp/zke/csdv20n400v/csdv.cum'\   Scenario
output file name.
         ccdf   'ccdf.dat'\ Produce CCDF output.
         waste       70                  \ kMTHM.
         canisters
         2335 6150 4875 3675 1275 5625 1073 \per zone.
         can_length 4.7                   \ canister length.
         can_radius 0.33                  \ canister radius.
         can_wall   0.01                \ canister  wall thickness.
         probability 1.0             \ scenario probability.

         start time=0\ Simulation start time.
         stop time=10080\ Simulation stop time.
         lhs file = '../oooo/lhsoooo.out' \ LHS output file for
this scenario,
         vectors  = 400               \ Number of vectors for this
scenario.
        chains  13
      nuclides 20
        elements 15
      members 2, 3, 2, 1, 4, 1, 1, 1, 1, 1, 1, 1, 1
      nucnam
            'CM246'          'U238'
            'CM245' 'AM241' 'NP237'
            'AM243' 'PU239'
            'PU240'
                    'U234'  'TH230' 'RA226' 'PB210'
            'CS137'
            'CS135' 'I129'          'TC99'
                    'NI59'  'C14'   'SE79'  'NB94'
      mass
            246             238
            245     241     237
            243     239
            240
                    234     230     226     210
            137
            135     129             99
                    59      14      79      94
```

SCENARio (Cont'd)

```
id_ele
      1                 2
      1        3        4
      3        5
      5
               2        6         7         8
      9
      9        10                 11
               12       13        14        15
par1
      0                 1
      0        1        2
      0        1
      0
               0        1         2         3
      0
      0        0                  0
               0        0         0         0
par2
      0                 0
      0        0        0
      0        0
      0
               0        0         0         0
      0
      0        0                  0
               0        0         0         0
bran1
      0                 1
      0        1        1
      0        1
      0
               0        1         1         1
      0
      0        0                  0
               0        0         0         0
bran2
      0                 0
      0        0        0
      0        0
      0
               0        0         0         0
      0
      0        0                  0
```

SCENARio (Cont'd)

```
                0       0       0       0
    inventory
            0.0258          0.318
            0.126   1640.   0.288
            15.5    308.
            508.
                    1.89    1.29e-4  3.67e-7 4.71e-8
            7.66e04
            0.35    0.0295          12.3
                    3.56    1.54    0.381   0.793
    half_life
            4.73E03         4.47E09
            8.5E03  4.32E02 2.14e6
            7.38E03 2.41E04
            6.54e3
                    2.45E05 7.70E04 1.6E03  22.3
            30.
            2.3E06 1.57E07          2.13E05
                   8.0E04  5.73E03  6.5e04 2.03E04
```

SEISMIc

PURPOSE This keyword of the scenario section schedules the Exec of the SEISMO module.

SYNTAX SEISMIC INPUT s1 OUTPUT s2 SOTEC s3 ¦

MODIFIERS INPUT

s1: The name of the standard input file for the SEISMO module.

OUTPUT

s2: The name of the file of container failure data to generate for the SOTEC module.

SOTEC

s3: The name of the SOTEC generated file holding container wall thickness data as a function of time.

COMMENTS None.

EXAMPLE

```
SEISMIC
     input = 'seismo.inp'
     output = 'seisot.dat'
     sotec = 'sotsei.dat' ¦
```

SOURCE

PURPOSE This keyword of the scenario section schedules the source term module SOTEC for execution.

SYNTAX SOURCE INPUT s1 LHSMAP s2 STOP n1 ¦

MODIFIERS INPUT

s1: The name of the SOTEC RDFREE-formatted input file.

LHSMAP

s2: The name of the LHS sampled data output file.

STOP

n1: The simulation stop time.

COMMENTS None.

EXAMPLE

```
SOURCE
     input  = 'sotec.inp'
     lhsmap = 'sotmap.dat'
     stop 10080 ¦
```

TEMPERature

PURPOSE This keyword of the scenario section schedules the temperature module CANT2 for execution.

SYNTAX TEMPERATURE |

MODIFIERS None.

COMMENTS This module is usually run external to the TPA Exec rather than including it in the loop of modules running for many vectors. This is because CANT2 does not use sampled variables or output from modules that do use sampled variables.

EXAMPLE `TEMPERATURE |`

TITLE

PURPOSE This header section keyword provides the title for the run and appears on the top of each output page. The title string is limited to 74 characters. This parameter is required to satisfy quality assurance concerns.

SYNTAX TITLE s1

s1: An unquoted string.

MODIFIERS None.

COMMENTS Note that the vertical bar is not required to terminate the title. This keyword always takes the next 74 characters on the line as the title string.

EXAMPLE

```
TITLE: Test #1.
```

VOLCANo

PURPOSE This keyword of the scenario section schedules the execution of the VOLCANO module for this scenario.

SYNTAX VOLCANO FLAG n1 LHSMAP s1 OUTPUT s2 ¦

MODIFIERS FLAG

n1: Invocation flag for the VOLCANO module. This should always be set to 1.

LHSMAP

s1: The name of the LHS sampled data output file.

OUTPUT

s2: The name of the VOLCANO standard output file.

COMMENTS None.

EXAMPLE

```
VOLCANO
     output = 'volcano.out'
     lhsmap = 'volmap.dat'
     flag 1 ¦
```

WATER

PURPOSE This keyword of the scenario section schedules the execution of the FLOWMOD module for this scenario.

SYNTAX WATER AREA n1 DENSITY n2 INFILTRATION s1 INLET n3 INPUT s2 LEGLEN n4 LHSMAP s3 MATRIX s4 OUTLET n5 SATLEN n6 SATURATED n7 UNSATURATED n8 WEIGHT n9 |

(See the comments for information on additional modifiers.)

MODIFIERS AREA

n1: Number of repository zones (areas).

DENSITY

n2: Matrix grain density for each layer.

INFILTRATION

s1: The name of the input file holding the infiltration table.

INLET

n3: The inlet areas for each zone.

INPUT

s2: The name of the FLOWMOD standard input file.

LEGLEN

n4: The unsaturated leg lengths for each layer of each zone.

LHSMAP

s3: The name of the FLOWMOD mapping file for LHS sampled variables.

MATRIX

s4: The name of the FLOWMOD input file containing the flux table.

WATER (Cont'd)

OUTLET

n5: The outlet areas for each zone.

SATLEN

n6: The leg lengths for each layer of each zone.

SATURATED

n7: The number of saturated layers.

UNSATURATED

n8: The number of unsaturated layers.

WEIGHT

n9: The EPA weighting factor for each nuclide in the nuclide name list as specified in the SCENARIO keyword.

COMMENTS The following modifiers may also be used with the WATER keyword. They are not described here since their meaning and syntax is identical to the presentation in the SCENARIO keyword section (CHAINS, NUCLIDE, ELEMENT, MEMBER, NAME, MASS, ID_ELE, PAR1, PAR2, BRAN1, BRAN2, INVENTORY, and HALF_LIFE).

EXAMPLE

```
WATER
      input 'flowmod.inp'
      infiltration '../flowmod/infil.tbl'
      matrix '../flowmod/flux.tbl'
      lhsmap '../flowmod/flomapcooo.dat'
      area 7
      saturated 7\ Layers
      unsaturated 7\ Layers
      weight
            0.1                 0.1
            0.1       0.1       0.1
            0.1       0.1
            0.1
```

WATER (Cont'd)

```
                0.1       0.01      0.1       1.
      1.
      1.      0.1                  10.
                1.        0.1       1.        1.
  inlet 3.1e5  1.4e6  1.1e6  6.6e5  2.6e5  1.2e6  2.0e5
outlet 2.62e5 2.62e5 2.62e5 2.62e5 2.62e5 2.62e5 2.62e5
leglength
          30.    0.    0.    0.    0.   0.0    0.0 \pluvial
          60.    0.   30.    0.    0.   0.0    0.0
          60.   30.   20.    0.    0.   0.0    0.0
          60.  140.    0.    0.   90.  40.0    0.0
          60.   30.    0.    0.    0.   0.0    0.0
          60.   70.   30.    0.    0.   0.0    0.0
          60.   20.   40.    0.    0.   0.0    0.0
satlength
        4950.    0. 1350.    0.  50.0    0.0    0.0 \pluvial
        5570.  380.  990.    0.  50.0    0.0    0.0
        5620.  280.    0.    0.   0.0    0.0    0.0
        5000. 1200.    0.    0.   0.0    0.0    0.0
        5400. 1220.    0.    0.   0.0    0.0    0.0
        5780.  900.    0.    0.   0.0    0.0    0.0
        5520.  100. 1350.    0.  60.0    0.0    0.0
density   \ Matrix grain density for each layer.
        2580. 2370. 2230. 2590. 2270. 2630. 2270.
```

5 DESCRIPTION OF OUTPUTS

The files described here are regularly produced by the TPA Exec for all runs. An example of the files is provided in the description of the sample problem in Appendix A. The OUTPUT keyword of the calculation section of the TPA input file affects only the contents of the 'tpa.out'. The form and content of the other files is fixed.

5.1 xxxx_TLY.DAT

This file is an execution tally file that logs the success or failure of each subprocess module execution for each vector of a scenario. The first four letters of the file name are reserved for the scenario name.

The file contains a simple 2D matrix of modules versus vector number with an integer in each position indicating the status of the run. A -1 indicates that the module was not scheduled to run for that particular vector. A 0 indicates that the run was unsuccessful. A 1 indicates that the run was successful. The scheme implemented is neither comprehensive nor robust but does detect any module abnormal termination from the operating system point of view (e.g., divide by zero).

5.2 ccdf.dat

The name of this file is user supplied with the SCENARIO CCDF keyword in the TPA input file. It is uniquely specified here to identify it in the sample problem output listings. The file contains the scenario release CCDF in TECPLOT format. Note it does not contain CCDF information from more than one scenario. An example of the plots available with the TECPLOT software is given in Figure 5-1.

5.3 ccdf.fmt

This file contains the release CCDF information for all scenarios invoked during a given run of the TPA system. The contents are also in a TECPLOT compatible format.

5.4 doseccdf.fmt

This file contains the dose CCDF information for all scenarios invoked during a given run of the TPA system. The contents are also in a TECPLOT compatible format.

5.5 epasum.fmt

This file contains the release CCDF information for all scenarios invoked during a given run of the TPA system. The contents are in standard printer output format.

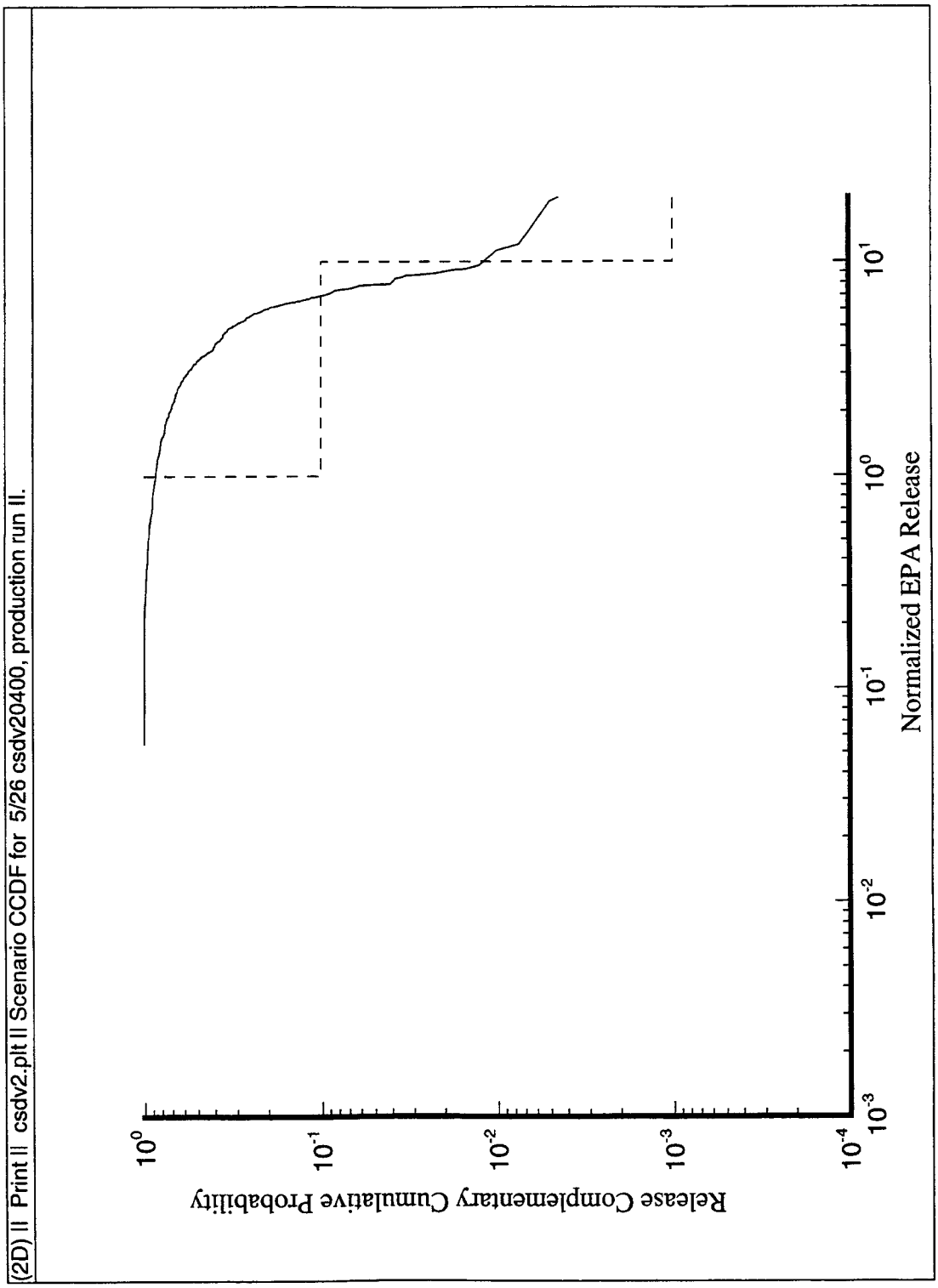

Figure 5-1. TECPLOT software examples

5.6 nucrel.fmt

This file contains the release information by nuclide. The nuclide information appears in alphabetical order as specified in the EPANRM common block in the TPA code. The values are summed over pathway, area, and modules.

5.7 pathre.fmt

This file contains the release information by pathway. The pathways are listed in their numerical order, 1=liquid, 2=air, and 3=direct. The values are summed over nuclide, area, and module.

5.8 scnrel.fmt

This file contains the release information by vector. The release values are listed in their vector number order. They are summed over nuclide, pathway, area, and module.

5.9 tpa.log

This file contains the log entries from the TPA module 'logmsg'. The messages indicate the times at which certain events in the Exec took place. By examining the start times of the subprocess an estimate of wall clock time can be derived for each module.

In addition, some process failure conditions may also be indicated, such as the abnormal return of a subprocess.

5.10 tpa.out

This is the standard output file for the TPA Exec code. A title is displayed in block letters followed by the software version number. Various configuration parameters are listed followed by an ASCII dump for the input file for this run. The scenario input section of the input file is then repeated before each scenario is run. If LHS is invoked its input file is also listed in this file.

After all scenario output is complete, the OUTPUT keyword of the calculation section is considered. The modifiers to this keyword determine which, if any, of the data arrays to list in this file. The output for each modifier will start at the top of a new page. For a detailed description of the OUTPUT keyword and its modifiers see Chapter 4.

6 VERIFICATION, BENCHMARKING, AND CONFIGURATION CONTROL

The TPA verification procedure consists of checking the data values transferred to the CM via the global data files. These files are the only control the Exec has on the subprocesses. The information is easily checked since it is written to an external ASCII file and can be inspected after the run is complete. The data returned by the modules to the Exec are much more voluminous and were spot checked using a debugger.

The modules are able to run in stand-alone mode or as part of the TPA system. This permits the output from both modes to be compared using a differencing utility to ensure common results. The debugger was again used to check the global data values after the global data file was read in the CM. The stand-alone version of the modules is considered the benchmark version.

All of the Exec were developed on the VAX/VMS system, and all new versions of the files were kept online using the version extension capability of the VMS system for file names. In addition, all changes to the CM, while integrating them to the TPA system, were also kept online using the version extension feature. Additional information is provided in the change request forms, a hard copy of which has been maintained since November 13, 1992.

The TPA code is managed under procedures set out in CNWRA Technical Operating Procedure (TOP)-018. The production of this user's manual is one of the requirements of TOP-018. A new version of the TPA will likely be produced during IPA Phase 3, at which time the user's manual will also be updated.

7 REFERENCES

Environmental Protection Agency. 1985. Environmental radiation protection standards for management and disposal of spent-nuclear fuel, high-level and transuranic radioactive wastes; final rules, Title 40, Code of Federal Regulations, part 191. *Federal Register*. Washington, DC: U.S. Environmental Protection Agency 20: 38065.

Iman, R.L., and M.J. Shortencarrier. 1984. *A Fortran 77 Program and User's Guide for the Generation of Latin Hypercube Samples for Use with Computer Models*. SAND83-2365, NUREG/CR-3624. Albuquerque, NM: Sandia National Laboratories.

Janetzke, R.W., and B. Sagar. 1991a. *preFOR: A Pre-Processor for Fortran Files, User's Manual*. CNWRA 91-003. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

Janetzke, R.W., and B. Sagar. 1991b. *RDFREE: A Fortran Utility for Format Free Input, User's Guide, CNWRA 91-005*. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

Margulies, T., L. Lancaster, N. Eisenberg, and L. Abramson. 1992. *Probability Analysis of Magma Scenarios for Assessing Geologic Waste Repository Performance*. American Society of Mechanical Engineers: Winter Annual Meeting, November 8-13, 1992, Anaheim, CA.

Napier, B.A., R.A. Peloquin, D.L. Streng, J.V. Ramschell. 1988. *GENII - The Hanford Environmental Radiation Dosimetry Software System, Volumes 1, 2, and 3*. PNL-6584. Richland, WA: Pacific Northwest Laboratory.

Nuclear Waste Policy Act. 1982. *Public Law 97-425, 96 Stat 2201, 42USC 10101, 1983*. Washington, DC.

Olague, N.E., D.E. Longsine, J.E. Campbell, and C.D. Leigh. 1991. *User's Manual for the NEFTRAN II Computer Code*. NUREG/CR-5618, SAND90-2089. Washington, DC: U.S. Government Printing Office: Superintendent of Documents.

Sagar, B., R.B. Codell, J. Walton, and R.W. Janetzke. 1992. *SOTEC: A Source Term Code for High-Level Nuclear Waste Geologic Repositories. User's Manual: Version 1.0*. CNWRA 92-009. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

TECPLOT. 1990. *TECPLOT User's Manual, Version 4. Interactive Plotting for Scientists and Engineers*. Bellevue, WA: Amtec Engineering, Inc.

Updegraff, C.D., C.E. Lee, and D.P. Gallegos. 1991. *DCM3D: A Dual Continuum, Three-Dimensional, Ground-Water Flow Code for Unsaturated, Fractured, Porous Media*. NUREG/CR-5536. Washington, DC: U.S. Nuclear Regulatory Commission.

U.S. Nuclear Regulatory Commission. 1990. Disposal of high-level radioactive wastes in geologic repositories. *Code of Federal Regulation, Part 60, Chapter 1, Title 10, "Energy."* Washington, DC: U.S. Nuclear Regulatory Commission.

APPENDIX A

SAMPLE PROBLEM

SAMPLE PROBLEM

See microfiche for text of Appendix A, Sample Problem.

APPENDIX B

ERROR MESSAGES

ERROR MESSAGES

All TPA error messages are generated by the subroutine "errmsg," which has two arguments. The first gives the routine name in which the error occurred and the second gives the error message to be printed in the output. The list below gives the actual arguments of all calls to "errmsg" for the TPA system. Some of the error messages are completed at run time and will appear different than listed here. For the purposes of this list, the FORTRAN variable name is preserved in these cases.

('AIREXC','AIRCOM input file does not exist.')
('AIREXC','AIRCOM status code ='//code)
('AIREXC','DRILLO2 air file does not exist.')
('AIREXC','VOLCANO air file does not exist.')
('C14EXC','C14 status code ='//code)
('C14EXC','C14 temperature file does not exist.')
('C14EXC','LHS map file does not exist.')
('CALCRD','Unrecognized data for keyword CCDF.')
('CALCRD','Unrecognized data for keyword CUMREL.')
('CALCRD','Unrecognized data for keyword EPASUM.')
('CALCRD','Unrecognized data for keyword FORMAT.')
('CALCRD','Unrecognized data for keyword NUCREL.')
('CALCRD','Unrecognized data for keyword OUTPUT.')
('CALCRD','Unrecognized data for keyword PATHRE.')
('CALCRD','Unrecognized data for keyword PATHREL.')
('CALCRD','Unrecognized data for keyword SCENRE.')
('CALCRD','Unrecognized data for keyword SCREEN.')
('CALCRD','Unrecognized data for keyword SEPASU.')
('DIAEXC','DITTY AIR input file does not exist.')
('DIAEXC','DITTY_A air file does not exist.')
('DIAEXC','DITTY_A population file does not exist.')
('DIAEXC','DITTY_A status code ='//code)
('DIAEXC','DITTY_A wind file does not exist.')
('DIFEXC','DITTY_F input file does not exist.')
('DIFEXC','DITTY_F population file does not exist.')
('DIFEXC','DITTY_F status code ='//code)
('DIFEXC','DITTY_F water file does not exist.')
('DIWEXC','DITTY_W input file does not exist.')
('DIWEXC','DITTY_W population file does not exist.')
('DIWEXC','DITTY_W status code ='//code)
('DIWEXC','DITTY_W water file does not exist.')
('DOSRD','File does not exist.')
('DOSRD','Invalid form: '//ctemp)
('DR1EXC','DRILLO1 status code ='//code)
('DR1EXC','LHS map file does not exist.')
('DR2EXC','DRILLO2 input file does not exist.')
('DR2EXC','DRILLO2 status code ='//code)
('FLOEXC','FLOWMOD infiltration file does not exist.')
('FLOEXC','FLOWMOD matrix file does not exist.')
('FLOEXC','FLOWMOD status code ='//code)
('FLOEXC','LHS map file does not exist.')
('LHSEXC','LHS input file does not exist.')
('NEFEXC','NEFTRAN input file does not exist.')
('NEFEXC','NEFTRAN status code ='//code)
('NEFRD','Number of vectors in the release file does not match the number of vectors for this scenario.')

('NEFRD','Unrecognized second line of nefreu',' file.')
('SCNINP','Invalid nuclide name = '// nucnam(nuc))
('SCNINP','No other modifiers are permitted when the END modifier is used.')
('SCNINP','Number of consequence modules exceeds MAXMDL processing AIRCOM.')
('SCNINP','Number of consequence modules exceeds MAXMDL processing C14.')
('SCNINP','Number of consequence modules exceeds MAXMDL processing DOSE_A.')
('SCNINP','Number of consequence modules exceeds MAXMDL processing DOSE_F.')
('SCNINP','Number of consequence modules exceeds MAXMDL processing DOSE_W.')
('SCNINP','Number of consequence modules exceeds MAXMDL processing DRILL1.')
('SCNINP','Number of consequence modules exceeds MAXMDL processing DRILL2.')
('SCNINP','Number of consequence modules exceeds MAXMDL processing NEFTRAN.')
('SCNINP','Number of consequence modules exceeds MAXMDL processing SEISMO.')
('SCNINP','Number of consequence modules exceeds MAXMDL processing SOURCE.')
('SCNINP','Number of consequence modules exceeds MAXMDL processing VOLCANO.')
('SCNINP','Number of vectors exceeds MAXVEC.')
('SCNINP','SCENAR keyword requires at least one modifier.')
('SCNINP','Unrecognized data for keyword ANALYS.')
('SCNINP','Unrecognized data for keyword C14 AIRCOM.')
('SCNINP','Unrecognized data for keyword C14 DELTA.')
('SCNINP','Unrecognized data for keyword C14 FIRST.')
('SCNINP','Unrecognized data for keyword C14 LAYERS.')
('SCNINP','Unrecognized data for keyword C14 LHSMAP.')
('SCNINP','Unrecognized data for keyword C14 NUMBER.')
('SCNINP','Unrecognized data for keyword C14 SKIP.')
('SCNINP','Unrecognized data for keyword C14 SOTEC.')
('SCNINP','Unrecognized data for keyword C14 STOP.')
('SCNINP','Unrecognized data for keyword C14 TEMPER.')
('SCNINP','Unrecognized data for keyword C14 TIMES.')
('SCNINP','Unrecognized data for keyword C14 VELOCI.')
('SCNINP','Unrecognized data for keyword CAN_WA.')
('SCNINP','Unrecognized data for keyword CANIST.')
('SCNINP','Unrecognized data for keyword CANLEN.')
('SCNINP','Unrecognized data for keyword CANRAD.')
('SCNINP','Unrecognized data for keyword DOSE_A AIR.')
('SCNINP','Unrecognized data for keyword DOSE_A DATA.')
('SCNINP','Unrecognized data for keyword DOSE_A INPUT.')
('SCNINP','Unrecognized data for keyword DOSE_A LHSMAP.')
('SCNINP','Unrecognized data for keyword DOSE_A NUCLID.')
('SCNINP','Unrecognized data for keyword DOSE_A OUTPUT.')
('SCNINP','Unrecognized data for keyword DOSE_A PLOT.')
('SCNINP','Unrecognized data for keyword DOSE_A POPULA.')
('SCNINP','Unrecognized data for keyword DOSE_A RELEAS.')
('SCNINP','Unrecognized data for keyword DOSE_A STOP.')
('SCNINP','Unrecognized data for keyword DOSE_A WATER.')
('SCNINP','Unrecognized data for keyword DOSE_A WIND.')
('SCNINP','Unrecognized data for keyword DOSE_F AIR.')
('SCNINP','Unrecognized data for keyword DOSE_F DATA.')
('SCNINP','Unrecognized data for keyword DOSE_F INPUT.')
('SCNINP','Unrecognized data for keyword DOSE_F LHSMAP.')
('SCNINP','Unrecognized data for keyword DOSE_F NUCLID.')
('SCNINP','Unrecognized data for keyword DOSE_F OUTPUT.')
('SCNINP','Unrecognized data for keyword DOSE_F PLOT.')
('SCNINP','Unrecognized data for keyword DOSE_F POPULA.')
('SCNINP','Unrecognized data for keyword DOSE_F RELEAS.')
('SCNINP','Unrecognized data for keyword DOSE_F STOP.')
('SCNINP','Unrecognized data for keyword DOSE_F WATER.')

('SCNINP','Unrecognized data for keyword DOSE_F WIND.')
('SCNINP','Unrecognized data for keyword DOSE_W AIR.')
('SCNINP','Unrecognized data for keyword DOSE_W DATA.')
('SCNINP','Unrecognized data for keyword DOSE_W INPUT.')
('SCNINP','Unrecognized data for keyword DOSE_W LHSMAP.')
('SCNINP','Unrecognized data for keyword DOSE_W NUCLID.')
('SCNINP','Unrecognized data for keyword DOSE_W OUTPUT.')
('SCNINP','Unrecognized data for keyword DOSE_W PLOT.')
('SCNINP','Unrecognized data for keyword DOSE_W POPULA.')
('SCNINP','Unrecognized data for keyword DOSE_W RELEAS.')
('SCNINP','Unrecognized data for keyword DOSE_W STOP.')
('SCNINP','Unrecognized data for keyword DOSE_W WATER.')
('SCNINP','Unrecognized data for keyword DOSE_W WIND.')
('SCNINP','Unrecognized data for keyword DRILL1 LHSMAP.')
('SCNINP','Unrecognized data for keyword DRILL2 AIRCOM.')
('SCNINP','Unrecognized data for keyword LHS INPUT.')
('SCNINP','Unrecognized data for keyword LHS OUTPUT.')
('SCNINP','Unrecognized data for keyword LHS VECTOR.')
('SCNINP','Unrecognized data for keyword NAME.')
('SCNINP','Unrecognized data for keyword NEFTRAN DISCHA.')
('SCNINP','Unrecognized data for keyword NEFTRAN INPUT.')
('SCNINP','Unrecognized data for keyword NEFTRAN LHSMAP.')
('SCNINP','Unrecognized data for keyword NEFTRAN OUTPUT.')
('SCNINP','Unrecognized data for keyword NEFTRAN RELEASE.')
('SCNINP','Unrecognized data for keyword NEFTRAN SOURCE.')
('SCNINP','Unrecognized data for keyword NUCLID.')
('SCNINP','Unrecognized data for keyword PROBAB.')
('SCNINP','Unrecognized data for keyword SCENARIO BRAN1.')
('SCNINP','Unrecognized data for keyword SCENARIO BRAN2.')
('SCNINP','Unrecognized data for keyword SCENARIO CHAINS.')
('SCNINP','Unrecognized data for keyword SCENARIO ELEMEN.')
('SCNINP','Unrecognized data for keyword SCENARIO HALF_L.')
('SCNINP','Unrecognized data for keyword SCENARIO ID_ELE.')
('SCNINP','Unrecognized data for keyword SCENARIO INVENT.')
('SCNINP','Unrecognized data for keyword SCENARIO LHS.')
('SCNINP','Unrecognized data for keyword SCENARIO MASS.')
('SCNINP','Unrecognized data for keyword SCENARIO MEMBER.')
('SCNINP','Unrecognized data for keyword SCENARIO NAME.')
('SCNINP','Unrecognized data for keyword SCENARIO NUCLID.')
('SCNINP','Unrecognized data for keyword SCENARIO PAR1.')
('SCNINP','Unrecognized data for keyword SCENARIO PAR2.')
('SCNINP','Unrecognized data for keyword SCENARIO VECTOR.')
('SCNINP','Unrecognized data for keyword SCENARIO.')
('SCNINP','Unrecognized data for keyword SEISMIC OUTPUT.')
('SCNINP','Unrecognized data for keyword SEISMIC SOTEC.')
('SCNINP','Unrecognized data for keyword SOTEC INPUT.')
('SCNINP','Unrecognized data for keyword SOTEC LHSMAP.')
('SCNINP','Unrecognized data for keyword SOTEC STOP.')
('SCNINP','Unrecognized data for keyword START.')
('SCNINP','Unrecognized data for keyword STOP.')
('SCNINP','Unrecognized data for keyword VOLCANO FLAG.')
('SCNINP','Unrecognized data for keyword VOLCANO INPUT.')
('SCNINP','Unrecognized data for keyword VOLCANO LHSMAP.')
('SCNINP','Unrecognized data for keyword VOLCANO STOP.')
('SCNINP','Unrecognized data for keyword WASTE.')
('SCNINP','Unrecognized data for keyword WATER AREA.')

('SCNINP','Unrecognized data for keyword WATER GRAINF.')
('SCNINP','Unrecognized data for keyword WATER GRAINM.')
('SCNINP','Unrecognized data for keyword WATER INFILT.')
('SCNINP','Unrecognized data for keyword WATER INLET.')
('SCNINP','Unrecognized data for keyword WATER INPUT.')
('SCNINP','Unrecognized data for keyword WATER LEGLEN.')
('SCNINP','Unrecognized data for keyword WATER LHSMAP.')
('SCNINP','Unrecognized data for keyword WATER MATRIX.')
('SCNINP','Unrecognized data for keyword WATER OUTLET.')
('SCNINP','Unrecognized data for keyword WATER SATLEN.')
('SCNINP','Unrecognized data for keyword WATER SATURA.')
('SCNINP','Unrecognized data for keyword WATER STOP.')
('SCNINP','Unrecognized data for keyword WATER UNSATU.')
('SCNINP','Unrecognized data for keyword WATER WEIGHT.')
('SCUMRD','CUMREL file does not exist '// srufil)
('SCUMRD','File has more areas than MAXARE.')
('SCUMRD','File has more modules than MAXMDL.')
('SCUMRD','File has more nuclides than MAXNUC.')
('SCUMRD','File has more pathways than MAXFRM.')
('SCUMRD','File has more vectors than MAXVEC.')
('SEIEXC','SEISMO status code ='//code)
('SOTEXC','LHS map file does not exist.')
('SOTEXC','SOTEC input file does not exist.')
('SOTEXC','SOTEC status code ='//code)
('TEMEXC','TEMPER input file does not exist.')
('VOLEXC','LHS map file does not exist.')
('VOLEXC','VOLCANO status code ='//code)

APPENDIX C

REQUIREMENTS DOCUMENT FOR TPA COMPUTER CODE

C REQUIREMENTS DOCUMENT FOR TPA COMPUTER CODE

C.1 INTRODUCTION

Phase 1 of IPA was completed by the NRC staff in April, 1990. The objective of Phase 1 of IPA was to demonstrate NRC's capability to conduct a PA for a HLW repository (NRC, 1990). In the Phase 1 effort, a simple total-system code was developed that processed information regarding scenario consequences and generated a CCDF of releases at the accessible environment boundary of a hypothetical HLW repository. Estimates of cumulative release over 10,000 years at the accessible environment boundary were formed by executing a series of CMs. The CMs were executed prior to, and separately from, an executive program and the estimates of cumulative release were stored on magnetic media for latter processing. The Phase 1 executive program accessed these estimates and calculated a CCDF. The Phase 1 IPA report recommended development of an internal mode Exec to control data transfer and sequential execution of the CMs. This report documents the requirements of such an Exec.

The IPA Phase 2 Exec of the TPA computer code will provide a stronger, more direct link between the executive program and the CMs. The TPA computer code will consist of a number of stand-alone CMs, whose execution will be controlled by an executive program in a manner such that the interfaces between the analyst and the CM will be minimized. In general, the TPA computer code will provide for automated calculational procedures that implement a theoretical and methodological approach to TPA.

C.2 PURPOSE OF THE TPA CODE AND THE Exec PROGRAM

The purpose of the IPA TPA Phase 2 computer code is to provide, using the appropriate data, calculated estimates of system performance (CCDFs, doses to man, etc.) for a HLW repository. In calculating these estimates, the system code will handle the uncertainty in future states of nature and in the variability in model parameters.

The Exec program will direct the computational sequence and the flow of information for estimating system performance. It will link and integrate the various distinct CMs involved in the estimations. The program will perform computations for sets of parametric input vectors and scenarios through the various CMs with minimal operator intervention; this will minimize the manual manipulation of data I/O files.

C.3 REQUIREMENTS AND CONSIDERATIONS FOR THE TPA CODE

C.3.1 Traceability/Reproducibility of Results

The TPA code will provide the means to insure traceability and reproducibility of calculations and estimations of system performance. The TPA code will be designed such that the majority of data transfer and file management will be controlled and performed automatically according to commands given to the executive program. The executive program will create a special log file containing a listing of modules run and the names of files opened, which can be used to recreate an execution of the TPA code. In addition, standard output files from the TPA code and the individual CM will contain information by which one can determine the origin of a file: the date and time of creation, the name and version of the module(s) creating the file, an echo of the input parameters used, and, in the case of the TPA code, a listing of the scenarios analyzed.

The TPA code will also attempt to ensure reproducibility of calculations and results across the various hardware environments (e.g., VAX, CRAY, PC, etc.). This will be accomplished by keeping hardware-specific calls to a minimum.

C.3.2 Modularity

The IPA TPA code will attempt to integrate many stand-alone modeling programs (consequence calculation modules) into an easily executed process. Although each of the CMs will be compiled and linked separately and will retain the ability to run outside the IPA system, within the TPA code, execution of any particular CM will be controlled by the executive program. The executive program will direct the execution of specific modules in a specified sequence for a particular scenario.

The CMs will have four types of parametric input values: (i) global; (ii) local; (iii) sampled; and (iv) transferred. Global parameters are deterministic quantities common to the input of more than one module. These include, for example, number of scenarios, number of parametric input vectors, and simulation time. These global parameters will be supplied by the analyst in the TPA input file and will be passed to each CM when that module is executed.

Local parameters are module-specific deterministic quantities; that is, they are designed specifically for the running of a particular CM. These quantities will be supplied by the analyst in the standard input file of the module. These parameters may be overwritten by global sampled values within the TPA code.

Sampled input parameters are parameters that are sampled by the LHS or other such programs over the distributions supplied by the analyst. These quantities may be specific to a single CM or common to more than one module. The analyst will also need to create a file to provide the mapping between the LHS output and the CM input.

Transferred parameters are those parameters generated by CMs and used by another CM that is run later in the scenario sequence. The transferal of data between CMs will be accomplished by using output files formatted specifically for the receiving CM. Interface subroutines within this trailing module are used to import this data.

C.3.3 Considerations for Developing CMs

CMs should be developed as stand-alone modeling programs that rely solely on local and sampled (if need be) parametric input. Although the developer of a new CM may foresee the need for certain data from another CM, the new CM should not require a link to this second module to run. This approach will afford the opportunity to develop and test the CMs independently of each other and the executive program.

Stochastic parameters that will be sampled by the LHS program will be treated as follows. For a given scenario, all of the stochastic parameters from all of the CMs required for that scenario will be identified. The input file for the LHS will be prepared that will include user supplied information on distributions and correlations, if any. Note that this will allow accounting for parameter correlations between parameters of not only any one of the modules, but also among parameters of different CMs. Execution of the LHS program will create the required number of sample vectors. Each of the sample vectors will then be accessed in turn by various CMs to obtain the sampled parameter values needed by it.

The simplest way to design a new CM is to assume that sampled parameters will be available to it one vector at a time. If it becomes desirable to test a new CM with LHS sampling, the following design features will facilitate incorporation of the CM into the TPA code: (i) data assignment statements for the sampled variables are located in a single section or subroutine of the code (e.g., like the NEFTRAN GETRV subroutine); and (ii) access to the LHS output file for the necessary data is designed so that only one vector at a time is read (rather than reading data for all vectors at one time and storing them in an array). Interface subroutines will be written into each CM by the Task 1 team to obtain the needed global, sampled, and transferred data from the appropriate sources and to insert them into the modules.

C.3.4 Output and Display

Written output from the TPA computer code will be in tabular form. Additionally, the output data will be written to plot files, both in ASCII and binary form, to be available for manipulation by graphics packages. The use of a commercially available graphics software system as a default graphics module in the TPA code is being explored. The various CMs will maintain their capability to produce standard output files to be used for intermediate results and diagnostic purposes.

C.3.5 Programming Language

The programming language for the IPA software system is FORTRAN. A FORTRAN preprocessor, called 'preFOR,' is available for use in developing the CM. preFOR is a program that inserts blocks of code at locations specified by the developer; this can help to reduce repetitive coding and to minimize errors during code development.

C.3.6 Target Hardware

The target computer for the initial version of the TPA code is the VAX/VMS at the CNWRA. Later versions will accommodate the IBM-PC/DOS, Silicon Graphics, SUN, and CRAY computers.

C.4 REQUIREMENTS AND CONSIDERATIONS FOR THE IPA Exec PROGRAM

C.4.1 Consequence Module Linkage

The IPA Exec is a computer program that will direct the flow of information and the computational sequences for estimating system performance. It will invoke the execution of the individual CMs via a system call, which transfers control of the central processing unit (CPU) from the Exec program to the CM. The CM would then run in stand-alone mode, with the generation and disposition of any data files under the control of the CM. When execution of the modules is finished, control is returned to the executive program. The FORTRAN statement to accomplish this is machine-dependent, but most machines have similar capability. This CPU dependence will be handled by the FORTRAN preprocessor preFOR without generating a separate version of the code for each CPU.

C.4.2 Input

The main input to the Exec will be via a prepared format-free ASCII file compatible with the RDFREE I/O library. The RDFREE library accepts free form FORTRAN input using keywords to identify the data that follows. Input includes the title for the run, the name of the analyst, a list of scenarios to be analyzed, the CM to execute for the individual scenarios, the names of the input files for the various CMs invoked, and the output file names for the CM and the TPA code.

C.4.3 Invoking Consequence Modules

The TPA input file will direct the executive program as to which CMs to execute and the order in which they should proceed for any particular scenario. Additionally, in lieu of invoking a CM to generate needed data, previously created output files may be used instead.

C.4.3.1 Sampled Parameters

As previously mentioned, sampled parameters may be shared by one or more CMs depending on the scenario, and these values are generated by LHS or another such sampling routine. The values are retrieved by a subroutine that will be added to each CM by the Task 1 team. This subroutine will interpret a parameter map file to obtain the desired values from the LHS output file. Each CM in the system will require this mapping structure.

C.4.3.2 Global Constants

Global constants are a fixed set of parameters that are undeviating for a given scenario. They are provided to all subprocesses in an ASCII file, and will be read by a subroutine provided with each CM. The complete list of global constants will depend upon the various CMs that will be included in the TPA code. In the initial version of the code, a few global parameters will be included; others will be added as more CMs are incorporated.

C.4.3.3 Multiple Scenarios

The input data defining all scenarios for a given execution of the IPA system code will be provided in the Exec input file. Presently, parameters considered unique to a scenario are the names of the CMs to be invoked, the input file names for the individual CMs, and the name of the LHS parameter map file. For a particular scenario, the CMs will be executed in the order in which they appear in the TPA input file.

C.4.3.4 Inter-Module Data Transmission

To avoid elaborate coupling modules that would translate variably formatted output files from one CM to obtain input values for the next module, an auxiliary subroutine will be provided with each module to create a fixed format data output file with a minimum of header information and only the numeric data needed for the following module(s). This data can then be read into the trailing module(s) or the Exec program using appropriate FORTRAN statements.

C.4.4 Output Data Files

Each CM will open and write to a unique output data file(s) when it is invoked for the first vector of a scenario. Output generated from additional vectors will be appended to the end of this file(s).

Following execution of all CMs for each individual scenario, the Exec program will access the required output files and generate CCDFs and other system performance measures from the data.

C.4.5 Plot Files

Modules will be available to produce fixed format ASCII and binary output files for input to commercial graphics packages.

C.4.6 IPA Process Log

The Exec program will provide a log file that will collect information concerning the occurrence of certain events (e.g., opening of data input files, start of CM execution, etc.) during a run of the TPA code.

C.4.7 Parameter Units

The International System (SI) system of units will be used for input and output throughout the IPA system.

APPENDIX D

STRUCTURE OF RDFREE COMMANDS

D STRUCTURE OF RDFREE COMMANDS

D.1 GENERAL RULES FOR CONSTRUCTING RDFREE INPUT

The following rules will be helpful in generating a RDFREE compatible input file.

1. Each logical record should end with the terminating character (|).
2. Blank lines are allowed.
3. A comment may begin in any column, but always ends at the end of line (column 80).
4. An end of logical record terminating character (|) is not recognized in a comment. That is, a logical record will not be terminated unless the terminating character appears to the left of a comment if both are used on the same line.
5. The maximum number of characters for a keyword or a data item is 64.
6. TITLE is an assumed keyword, and its associated data string has a unique string variable.
7. Although the keyword input is not case sensitive, the master keyword list is, and should be provided in uppercase.
8. Only five separator characters are allowed (space, equal, comma, tab, and colon) in addition to carriage return (end of line).
9. The total number of string data values associated with all keywords of a given logical record is limited by the parameter **MAXSVL**.
10. The total number of numeric data values associated with all keywords of a given logical record is limited by the parameter **MAXDVL**.
11. The * provides a repeat function, where the data value immediately following the * is repeated, as indicated by the integer which immediately precedes the * .

APPENDIX E

FILE IDENTIFICATION TABLES

TPA FILE IDENTIFICATION TABLE

MODULE	FILE	SOURCE/DESTINATION	DESCRIPTION
TPA	tpa.inp	Input from analyst	Main TPA input file
	tpa.log	Output to analyst	Log of TPA system activities
	tpa.out	Output to analyst	Standard output from TPA Exec
	TPA_LHS.LGD	Output to LHS	Global data for LHS
	TPA_FLO.FGD	Output to FLOWMOD	Global data for FLOWMOD
	TPA_SOT.SGD	Output to SOTEC	Global data for SOTEC
	TPA_NEF.NGD	Output to NEFTRAN	Global data for NEFTRAN
	TPA_C14.CGD	Output to C14	Global data for C14
	TPA_DIT.DGD	Output to DITTY	Global data for DITTY
	TPA_SEI.SGD	Output to SEISMO	Global data for SEISMO
	TPA_DR1.DGD	Output to DRILLO1	Global data for DRILLO1
	TPA_VOL.VGD	Output to VOLCANO	Global data for VOLCANO
	TPA_DR2.DGD	Output to DRILLO2	Global data for DRILLO2
	TPA_AIR.AGD	Output to AIRCOM	Global data for AIRCOM
	flotpa.dat	Input from FLOWMOD	GWTT and infiltration
	neftpa.dat	Input from NEFTRAN	Total release at 10,000 yrs by vector, area, and nuclide
	nucrel.fmt	Output to analyst	Release values by nuclide
	cumrel.fmt	Output to analyst	Individual release values
	epasum.fmt	Output to analyst	EPA sums
	pathre.dat	Output to analyst	Release values by path
	scnrel.fmt	Output to analyst	Release values by scenario
	ccdf.fmt	Output to TECPLOT	Total CCDF for all scenarios
	xxxx_ccdf.dat	Output to TECPLOT	CCDF for scenario xxxx

MODULE	FILE	SOURCE/DESTINATION	DESCRIPTION
	xxxx_TLY.DAT	Output to sensitivity analysis	Tally of which modules were scheduled to run and which were completed
	dittpa.dat	Input from DITTY	Dose values by nuclide
	xxxx.cum	Output from TPA Exec	Cumulative release values for scenario xxxx

MODULE	FILE	SOURCE/DESTINATION	DESCRIPTION
LHS	lhs.inp	Input from analyst	Standard input file for LHS containing all the sampled variables for all of the modules
	TPA_LHS.LGD	Input from TPA	Global data values
	lhs_const.dat	Output to sensitivity analysis	Identification of constant data in the LHS output file
	lhs.out	Output to all consequence modules	List directed output containing all sampled variables for all modules and all vectors for the current scenario

MODULE	FILE	SOURCE/DESTINATION	DESCRIPTION
TEMPER	cant2.in	Input from analyst	Standard input file for temperature program
	tzone.dat	Output to SOTEC	Temperature values
	cant2av.out	Output to analyst	Average temperature
	cant2.out	Output to analyst	Standard output file
	cant2.lu6	Output to analyst	Standard output file

MODULE	FILE	SOURCE/DESTINATION	DESCRIPTION
FLOWMOD	flowoooo.inp	Input from analyst	Standard input file for base case
	flowcooo.inp	Input from analyst	Standard input file for pluvial case
	TPA_FLO.FGD	Input from TPA	Global data file
	lhs.out	Input from LHS	See above
	flomapoooo.dat	Input from analyst	FLOWMOD map file for LHS sampled values for nonpluvial cases
	flomapcooo.dat	Input from analyst	FLOWMOD map file for LHS sampled values for pluvial cases
	flux.tbl	Input from analyst	Matrix flux ratios
	infil.tbl	Input from analyst	Infiltration per zone
	flosot.dat	Output to SOTEC	Time and flux in cubic meters per year
	flotpa.dat	Output to DRILLO2 and sensitivity analysis	GWTT and infiltration
	flonef.dat	Output to NEFTRAN	Standard NEFTRAN input file

MODULE	FILE	SOURCE/DESTINATION	DESCRIPTION
SOTEC	sotec.inp	Input from analyst	Main input file for SOTEC in RDFREE format
	TPA_SOT.SGD	Input from TPA	Global data values
	lhs.out	Input from LHS	See above
	sotmap.dat	Input from analyst	Map file for LHS data
	flosot.dat	Input from FLOWMOD	Flux data
	sotec.nuc	Input from analyst	Nuclide data file[1]
	mechan.dat	Input from analyst	Mechanical parameters
	tzone.dat	Input from TEMPER	Temperature data file
	sotc14.dat	Output to C14	Airborne release data for C14
	seisot.dat	Input from SEISMO	Failure time by zone
	confail.out	Output to sensitivity analysis	Container failure times
	sotsei.dat	Output to SEISMO	Base case container thickness data
	inv1000.out	Output to analyst	Initital and 1,000 yr inventory by nuclide
	sotnef.dat	Output to NEFTRAN	Water release rates
	dr1sot.dat	Input from DRILLO1	Number of cans hit per zone
	maxrel.dat	Output to SA	Maximum fractional release per nuclide
	maxc14.dat	Output to SA	Maximum fractional release for C14

[1] The format of the nuclide data file was created by R. Codell, NRC.

MODULE	FILE	SOURCE/DESTINATION	DESCRIPTION
NEFTRAN	flonef.dat	Input from FLOWMOD	Main input file
	TPA_NEF.NGD	Input from TPA	Global data file
	lhs.out	Input from LHS	See above
	nefmap.dat	Not used	N/A
	neftpa.dat	Output to TPA	Release values are appended to this file for all vectors of a scenario
	sotnef.dat	Input from SOTEC	Source term file
	nefdr2.dat	Output to DRILLO2	Concentrations by year and zone
	nefrep.out	Output to analyst	Standard output
	watrel.in	Output to DITTY	Concentrations by year and nuclide

MODULE	FILE	SOURCE/DESTINATION	DESCRIPTION
C14	c14h.in	Input from analyst	Standard input file for C14
	TPA_C14.CGD	Input from TPA	Global data values
	lhs.out	Input from LHS	See above
	c14map.dat	Input from analyst	C14 map file for LHS output sampled data
	fdtemp2a.out	Input from analyst	Temperature field calculated by FDTEMP2D
	sotc14.dat	Input from SOTEC	C14 rates by year
	c14h.vel	Scratch file generated internally and based on sampled perm-eabilities (AKR)	Velocity data file
	c14air.dat	Output to AIRCOM	C14 release rates by year
	C14H.REP	Output to analyst	Report on unit 6

MODULE	FILE	SOURCE/DESTINATION	DESCRIPTION
DITTY	TPA_DIT.DGD	Input from TPA	Global data values
	airrel.in	Input from AIRCOM	Air release versus time
	watrel.in	Input from NEFTRAN	Liquid release versus time
	dittpa.dat	Output to TPA for both the air case and water case. File is read immediately after each vector.	Dose data for CCDF
	dittec.plt	Output to TPA	Plot file
	jointfre.in	Input from analyst	Wind data
	poppop.in	Input from analyst	Population data
	rmdlib.dat	Input from analyst	Radionuclide master library
	ftrans.dat	Input from analyst	Food transfer factor library
	bioac1.dat	Input from analyst	Bioaccumulation library
	dsfct30.dat	Input from analyst	Committed dose equivalents from chronic exposure for 70 yr
	grdf.dat	Input from analyst	External dose factors
	ditapoc.in	Input from analyst	Air input for population chronic case
	ditwpoc.in	Input from analyst	Water input for population chronic case
	ditfpoc.in	Input from analyst	Food (beef export) input for population chronic case
	dittya.out	Output to analyst	Standard output file for air runs
	dittyw.out	Output to analyst	Standard output file for liquid runs
	dittyqa.out	Output to analyst	Summary report

MODULE	FILE	SOURCE/DESTINATION	DESCRIPTION
SEISMO	seismo.in	Input from analyst	Standard input file
	sotsei.dat	Input from SOTEC base case run	Container thickness data
	TPA_SEI.SGD	Input form TPA	Global data file
	seismo.out	Output to analyst	Standard output file
	seisot.dat	Output to SOTEC	Container failure times by zone
	lhs.out	Input from LHS	Sampled data

MODULE	FILE	SOURCE/DESTINATION	DESCRIPTION
VOLCANO	volcano.in	Input from analyst	Standard input file
	TPA_VOL.VGD	Input from TPA	Global data file
	lhs.out	Input from LHS	Sampled data
	test15.nuc	Input from analyst	Nuclide data
	volmap.dat	Input from analyst	LHS map file
	volcano.out	Output to analyst	Standard output file
	volsot.dat	Output to SOTEC	Number of containers failed by zone
	volair.dat	Output to AIRCOM	Air release data
	volcano.hst	Output to analyst	Cumulative history of all events in a run

MODULE	FILE	SOURCE/DESTINATION	DESCRIPTION
DRILLO1	dr1map.dat	Input from analyst	LHS map file
	drill1.in	Input from analyst	Standard input file
	flowmod.inp	Input from analyst	FLOWMOD standard input file used also by DRILLO1
	lhs.out	Input from LHS	Sampled data file
	TPA_DR1.DGD	Input from TPA	Global data file
	dr1dr2.dat	Output to DRILLO2	Standard input file for DRILLO
	dr1sot.dat	Output to SOTEC	Container failure times by zone
	drill1.out	Output to analyst	Standard output file

MODULE	FILE	SOURCE/DESTINATION	DESCRIPTION
DRILLO2	dr2nuc.dat	Input from analyst	Nuclide data
	dr1dr2.dat	Input from DRILLO1	Standard input
	flotpa.dat	Input from FLOWMOD	GWTT and infiltration data
	nefdr2.dat	Input from NEFTRAN	Concentrations by nuclide and year
	sotnef.dat	Input from SOTEC	Release rates from EBS
	TPA_DR2.DGD	Input from TPA	Global data file
	dr2air.dat	Output to AIRCOM	Air release data
	drill2.out	Output to analyst	Standard output file

MODULE	FILE	SOURCE/DESTINATION	DESCRIPTION
AIRCOM	aircom.inp	Input from analyst	Nuclide respirable fractions
	c14air.dat	Input from C14	Air release data
	volair.dat	Input from VOLCANO	Air release data
	dr2air.dat	Input from DRILLO2	Air release data
	TPA_AIR.AGD	Input from TPA	Global data file
	airrel.in	Output to DITTY for air case	Combined air release data