

# **DRAFT**

## **RIP REPOSITORY PERFORMANCE ASSESSMENT AND STRATEGY EVALUATION MODEL: USER'S GUIDE Version 1.00a**

R. Kossik

Golder Associates Inc.

Prepared for the U.S. DOE Office of Civilian Radioactive Waste Management  
Work Performed Under Argonne National Laboratory Contract Number 02352401

March 2, 1992

903-1371.203

**TABLE OF CONTENTS**

<b>1.</b>	<b>INTRODUCTION .....</b>	<b>1</b>
<b>2.</b>	<b>BASIC RIP MODEL PHILOSOPHY AND STRUCTURE .....</b>	<b>2</b>
2.1	Overview of RIP Background and Methodology .....	2
2.2.1	Terminology .....	3
2.1.2	Total System Performance Assessment Model .....	4
2.1.3	Strategy Evaluation Model .....	6
2.2	Parameter Definition and the Flexibility of the User Interface .....	10
2.3	Software Structure and Hardware Requirements .....	13
2.4	Software Installation and Implementation .....	15
2.5	Model Applicability .....	16
2.6	Error Reporting .....	16
<b>3.</b>	<b>INTRODUCTION TO THE RIP USER INTERFACE .....</b>	<b>17</b>
3.1	Starting RIP and Manipulating Input Data Files .....	17
3.2	Basics of the User Interface .....	18
3.2.1	Menu Structure .....	18
3.2.2	Using Data Input Windows .....	23
3.2.3	Specifying Parameters .....	25
3.2.4	The On-Screen HELP Utility .....	26
3.2.5	Printing and Saving Input or Output Screens .....	27
3.2.6	Units .....	27
3.3	Entering Data and Applying RIP .....	28
3.4	Editing the [GENERAL] Input Window .....	28
<b>4.</b>	<b>DEFINING PARAMETERS .....</b>	<b>31</b>
4.1	Entering the Parameters Module .....	31
4.2	Browsing and Editing the Parameter Database .....	32
4.3	Editing Parameters .....	33
4.3.1	Defining a Constant Parameter .....	34
4.3.2	Defining a Function Parameter .....	35
4.3.3	Defining a Stochastic Parameter .....	37
4.3.4	Defining an Event Parameter .....	39
4.3.5	Reserved Parameter Names .....	40
4.4	Displaying Stochastic Parameters and Functions .....	41
4.5	Application of the Parameters Module .....	44
<b>5.</b>	<b>DATA ENTRY AND APPLICATION OF THE PERFORMANCE ASSESSMENT MODEL .....</b>	<b>45</b>
5.1	Input for the Waste Package Behavior and Radionuclide Release Component Model .....	46
5.1.1	Describing Near-Field Conditions .....	46
5.1.1.1	Water Contact Mode .....	49
5.1.1.2	Temperature Conditions .....	52
5.1.1.3	Other Factors .....	60

TABLE OF CONTENTS (cont.)

	5.1.1.4 Show Groups .....	61
5.1.2	Describing Waste Package Behavior and Radionuclide Release .....	62
	5.1.2.1 Waste Package Characteristics .....	63
	5.1.2.2 Mass Transfer Parameters .....	64
	5.1.2.3 Parameters Describing Bound Exposure (Alteration/Dissolution of the Waste Matrix) .....	67
	5.1.2.4 Describing Waste Package Failure .....	69
	5.1.2.5 Describing Radionuclide and Elemental Properties .....	78
	5.1.2.6 Linking the Waste Package Release to the Far-Field Transport Pathways .....	81
	5.1.3 Making a Waste Package Transfer File .....	82
5.2	Input for the Radionuclide Transport Pathways Component Model .....	88
	5.2.1 Pathway Definition .....	89
	5.2.2 Pathway Network Definition .....	89
	5.2.3 Flow Mode Definition .....	90
	5.2.4 Definition of Pathway Tolerance Values .....	93
5.3	Input for the Disruptive Events Component Model .....	94
	5.3.1 Event Definition .....	96
	5.3.2 Definition of Event Descriptors .....	96
	5.3.3 Definition of Event Consequences .....	97
5.4	Making a Total System Transfer File .....	103
5.5	Implementing the Back End Program .....	110
5.6	Input Documentation .....	111
6.	DATA ENTRY AND APPLICATION OF THE STRATEGY EVALUATION MODEL .....	112
7.	RESULTS POST-PROCESSING AND DISPLAY .....	113
8.	REFERENCES .....	114

LIST OF FIGURES

2-1	RIP Performance Assessment Model Components
2-2	RIP Performance Assessment Model Simulation Logic
2-3	Presentation of Probabilistic Results
2-4	Overview of RIP Strategy Evaluation
2-5	Suite f Performance Results for Strategy A
2-6	Probability of Post-Strategy Performance Results
2-7	RIP Front End/Back End Structure
3-1	RIP Menu Structure
5-1	Example of Discretization of Repository into Waste Package Groups
5-2a	Temperature Uncertainty in Time/Temperature Pairs

TABLE OF CONTENTS (cont.)

LIST OF FIGURES (cont.)

5-2b Time Uncertainty in Time/Temperature Pairs

LIST OF APPENDICES

APPENDIX A Stochastic Parameter Distributions

APPENDIX B Specifying Stochastic Fractions

APPENDIX C Normalization of Results

APPENDIX D Radionuclide Table

APPENDIX E Error Reporting

APPENDIX F Source Code Listing

## 1. INTRODUCTION

This document is a user's guide for the integrated repository performance assessment and strategy evaluation software RIP (Repository Integration Program) developed by Golder Associates Inc. (GAI) for the DOE Office of Civilian Radioactive Waste Management (OCRWM). The underlying theory and capabilities of the software are described in detail in a companion document (GAI, 1992). This companion document, the RIP Theory and Capabilities Manual, *should be read prior to using this user's guide* and is *required reading* to ensure appropriate application of the RIP software.

The purpose of this document is to provide the specific information required to specify input and use the RIP program. This document discusses hardware requirements, the basic structure of the software, data entry and application of the model, and post-processing of results. In combination with the RIP Theory and Capabilities Manual, this document provides the necessary documentation and instructions to apply the RIP software.

Chapter 2 describes the basic RIP model philosophy and structure, including a summary of RIP theory and capabilities, a discussion of the philosophy, structure and flexibility of the user interface, and details of hardware requirements and software installation. Chapter 3 then provides an overview of the RIP user interface, describing the menu organization and the basics of data entry and manipulation. Chapter 4 describes the RIP Parameters module, the software component which enables model parameters to be defined by the user stochastically and/or as functions of other parameters. This module is a cornerstone of the RIP methodology and an understanding of its structure and capabilities, and the flexibility that it imparts to the model, is required to fully apply RIP. Chapter 5 then discusses in detail the actual data entry and application of the performance assessment model. Chapter 6 discusses data entry and application of the strategy model. Finally, Chapter 7 describes post-processing of RIP results. References are included in Chapter 8.

**NOTE:** RIP is a complex piece of software. Proper application of RIP requires an understanding of the underlying principles and assumptions. Reading the associated documentation prior to using the software can not be overemphasized. Although RIP includes an excellent HELP utility, this should be used as a reminder to the user, not as a tutor. Before attempting to use the software, it is recommended that the user first read the Theory and Capabilities Manual. This is a large, detailed document and will not necessarily be completely digested in one reading. However, reading through the document once will provide the user with a basic understanding of RIP's capabilities and limitations. Having done so, the user should read this User's Guide. It would be most effective to follow along with RIP loaded on a computer in front of the reader as each part of the software is explained. Although learning to use RIP in the manner outlined above requires a substantial amount of time and effort up-front, this will be more than compensated for in the long-term, and will help ensure appropriate application of the software and meaningful interpretation of results.

## 2. BASIC RIP MODEL PHILOSOPHY AND STRUCTURE

### 2.1 Overview of RIP Background and Methodology

The U.S. Department of Energy's Office of Civilian Radioactive Waste Management (OCRWM) contracted Golder Associates Inc. (GAI) through Argonne National Laboratory to independently develop a repository site-suitability methodology, and, based on that methodology, to recommend a strategic plan of action for evaluating the suitability of the Yucca Mountain, Nevada site for the development of a high level radioactive waste repository. The approach selected by GAI was based on two cornerstones:

- Development and continual updating of a preliminary *total system performance assessment model*; and
- Integration of the performance assessment model with various proposed site characterization activities using a *probabilistic decision analysis* approach in order to evaluate alternative strategies and to guide and prioritize the site characterization process.

Traditionally, performance assessment has been considered as an essential method to be implemented at the later stages in a repository siting program in order to demonstrate that the candidate site is in compliance with the licensing criteria. However, performance assessment can also be used as a tool to aid in the development and implementation of the entire siting procedure. In fact, GAI and others (e.g., GAI, 1977; NRC, 1983; GAI, 1986) have long maintained that the only effective way to make valid site characterization decisions (what should be studied, when, and in what detail) is by using preliminary integrated performance assessments. This view has recently been reiterated by both the National Research Council (1990) and the Nuclear Waste Technical Review Board (1990).

The methodology that was developed by GAI is intended to not only provide an initial evaluation of the current YMP program, but to also provide a management tool to help assure continuing program refinement and an early resolution of site suitability. That is, the integrated performance assessment model is intended to be used in an ongoing review process to continually reevaluate which data needs are most critical, and to reallocate program resources appropriately.

A related and equally important use for the model has become apparent to GAI in the process of eliciting information regarding models and input data from scientists and engineers investigating the Yucca Mountain site. Namely, the performance assessment model (and the process of eliciting model input) can act as a mechanism for integrating the large amount of available information generated by different groups into a meaningful whole (in a sense, allowing one to keep the "big picture" and the ultimate aims of the project clearly in focus). Such an integration would not only be useful for project managers (and is ultimately required in order to carry out a meaningful assessment of site suitability), but is also useful to individual project scientists to help them formulate and present the results of their research in a manner such that *the results can be readily applied to predicting the performance and determining*

*the suitability of the site.*

The focus of the GAI effort was the development of a computer program designed to carry out preliminary system performance assessments and to quantitatively evaluate alternative strategies for site characterization. Named RIP (for Repository Integration Program), this program is intended to integrate existing knowledge and component-level performance models that have already been developed within the YMP or elsewhere.

### 2.2.1 Terminology

The experienced reader will note that the term "model" is used somewhat loosely in this report. Because modelling and simulation terminology is generally not used uniformly throughout the engineering and scientific community, it is important to clearly define at the outset our set of terminology. In particular, it is important to differentiate between what we refer to as the *computational model*, the *conceptual model*, and the *simulation model*.

As used here, the computational model is the tool which provides the mathematical and computational framework for modeling a system of processes. It is more accurately referred to as the *computational algorithm* or *simulator*. It does not specifically describe an actual physical system. Rather, it simply consists of a set of flexible tools (i.e., a group of algorithms) for doing so. The computational model is embodied within software (i.e., a computer code). The user of the software must provide the proper input data in order to simulate an actual physical system.

A *conceptual model* is a simplified representation of the physical and chemical properties and processes controlling behavior of a physical system. That is, a conceptual model is essentially a body of ideas, based on available data, which summarizes the current understanding of a system. The conceptual model not only includes a description and quantification of the controlling processes and parameters, but also quantifies the uncertainties involved. (In fact, due to large uncertainties involved in geological systems, a number of alternative conceptual models are likely to be developed.)

The simulation model is the actual application of the computational model to given conceptual model. That is, the simulation model couples the tools within the computational model (the actual computer code) with a particular conceptual model (represented by a distinct set of input data for the computer code).

Summarizing in general terms, the *computational model* is the mathematical framework embodied in the actual computer code. A *conceptual model* is simplified representation of the physical system to be modeled. A *simulation model* is the application of the computer code using a specific data set in order to simulate the behavior of the system described by the conceptual model.

In this document, the term "model" used by itself refers only to the *computational model*. Hence, when we speak of the "radionuclide transport component model" or the "waste package component model", we are referring to the computational algorithms describing radionuclide transport pathways or waste package behavior, respectively (a component model

being a subset or part of the total computational model). The complete terms "*conceptual model*" and "*simulation model*", on the other hand, are always used explicitly when referring to these two concepts.

### 2.1.2 Total System Performance Assessment Model

The major portion of the RIP software is the *total system performance assessment (PA) model*. The PA model is essentially a radionuclide transport model which consists of a series of inter-connected, fully-coupled component models. The component models consist, in general, of simple functions relating various system parameters which control processes affecting radionuclide transport. The two primary component models address: 1) waste package behavior and radionuclide release, and 2) transport pathways to the accessible environment. An additional component model describes ancillary processes (disruptive events), such as human intrusion, tectonics and volcanism, which may affect system parameters. Figure 2-1 is a schematic of the structure of the performance assessment model.

RIP is *not* a detailed deterministic model based completely on low-level physically-based parameters. (For example, it does not explicitly model radionuclide transport through the unsaturated and saturated zone by solving a three-dimensional advection-diffusion equation.) RIP is a *probabilistic model* that is intended to directly represent the uncertainties in processes and events and their controlling parameters. RIP is intended to describe, rather than explain system behavior. Hence, RIP is based on higher-level parameters which represent the controlling processes, without necessarily directly simulating the details. This is not to say that detailed models of the controlling processes are ignored. It is intended that such detailed models be used to generate the appropriate input parameters for RIP (e.g., in the form of response surfaces or approximate analytical expressions).

It is also important to note that RIP is a *preliminary* performance assessment model. It is intended to be used as a management tool to help guide site characterization, as well as a tool for project scientists who wish to carry out sensitivity analyses and/or test alternative hypotheses with respect to system performance. In its present form, it is not intended to be a licensing tool.

The RIP performance assessment software is similar to a spreadsheet: while it contains a large amount of built-in logic and calculational capabilities, the problem that is solved is entirely defined by the user. RIP has been developed such that it is relatively free of assumptions regarding the details of waste package behavior and radionuclide transport processes. That is, the model consists of a basic computational framework representing the controlling processes, but is intentionally flexible such that the user can represent the processes with as much detail as desired.

Due to the inherent uncertainties resulting from our lack of knowledge, many of the parameters controlling processes related to the performance of a repository can only be represented by probability distribution functions (pdf's), as opposed to single values. RIP is designed such that uncertainties in both the model parameters *and* the component models themselves can be explicitly represented.



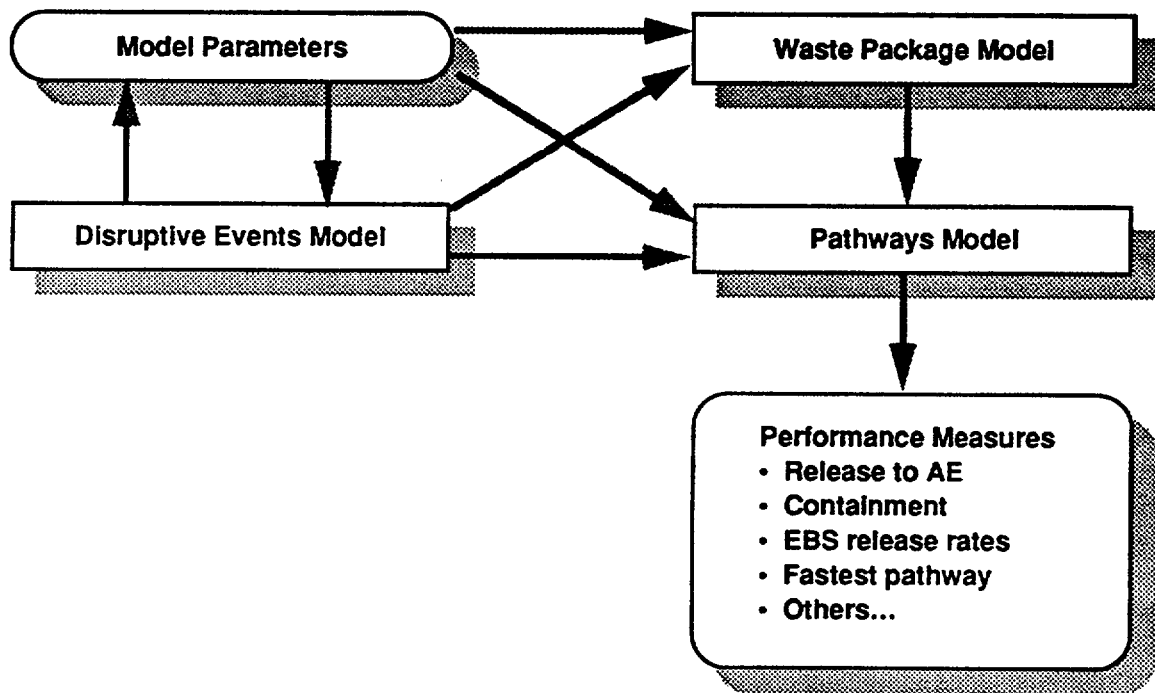


FIGURE 2-1  
RIP PERFORMANCE ASSESSMENT MODEL COMPONENTS  
ARGONNE/MODEL DEVELOPMENT

The RIP model uses a *simulation approach*, utilizing the Monte Carlo method to sample the probability distributions for the uncertain parameters (describing both processes and events) and simulate a large number of system realizations in order to determine probability distributions of site performance (e.g., cumulative release, transport time). That is, RIP steps through time and creates a time history of disruptive events and other system parameters for each system realization, simulates the behavior of the system under those conditions, and then combines the results of all the realizations in an appropriate manner to determine probability distributions of site performance. Because model parameters are described stochastically, each realization produces a different time history of events and processes. The integrated model is designed such that it can handle all combinations of model parameters and time histories which might be realized. RIP utilizes an importance sampling scheme to increase the efficiency of the Monte Carlo sampling process. Figure 2-2 summarizes the performance assessment model simulation logic.

In general terms, the output for the performance assessment model consists of performance measures for the repository system. These performance measures are probabilistic in nature. That is, output is not a single value, but a distribution which specifies the probability of exceedence for any particular value for the performance measure (e.g., cumulative release to the accessible environment). Figure 2-3 illustrates three ways in which such results can be presented.

As will be discussed subsequently, the RIP user can, if desired, simulate only the waste package behavior and radionuclide release portion of the performance model. The output would consist only of measures of waste package performance (e.g., cumulative waste package release, maximum annual waste package release). Likewise, the user can, if desired, only simulate the radionuclide transport pathways portion of the performance model (by assuming a constant waste package release rate). In either case, disruptive events can be included or ignored.

### 2.1.3 Strategy Evaluation Model

The performance assessment model can be run independently to produce probabilistic measures of performance for a site (e.g., cumulative release to accessible environment). In addition, however, the PA model can be used as a component of a probabilistic decision analysis strategy assessment model. The strategy model is essentially a shell which contains the performance assessment model. That is, the PA model is embedded within a decision analysis model which allows the user to evaluate alternative site characterization strategies.

Figure 2-4 provides a schematic of the RIP strategy evaluation model. The RIP user defines a characterization *strategy*, which consists of a set of characterization *activities*. A characterization activity is a specific scientific study or group of studies proposed for the site (e.g., surface-based testing). For each activity, the user specifies the estimated cost, the estimated duration (including required precedent activities), and the degree to which the uncertainty in one or more system parameters may be reduced by the activity.

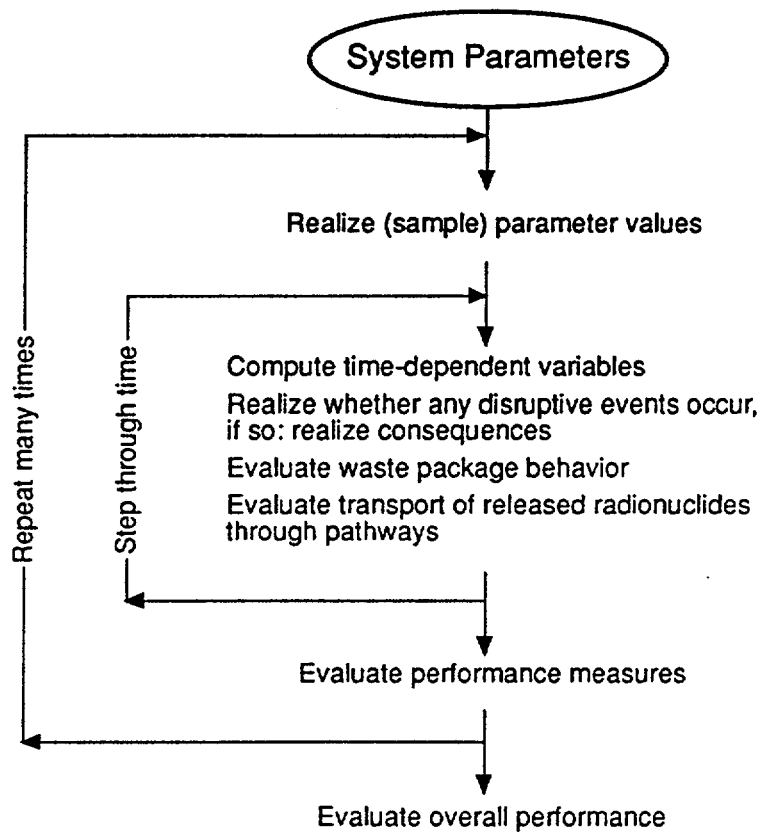
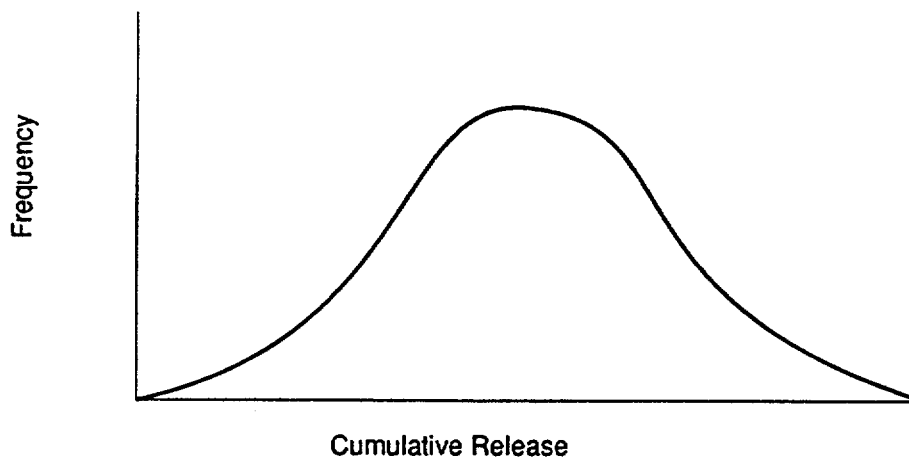
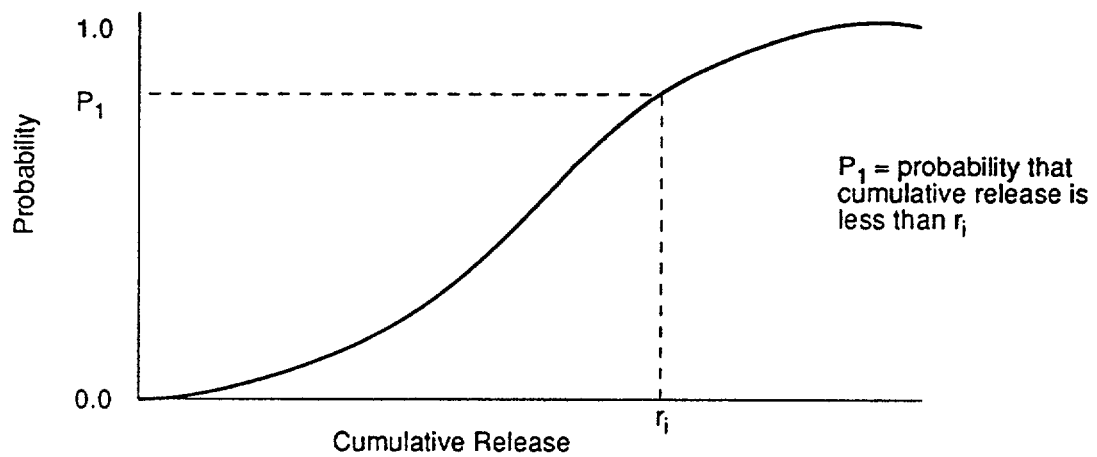


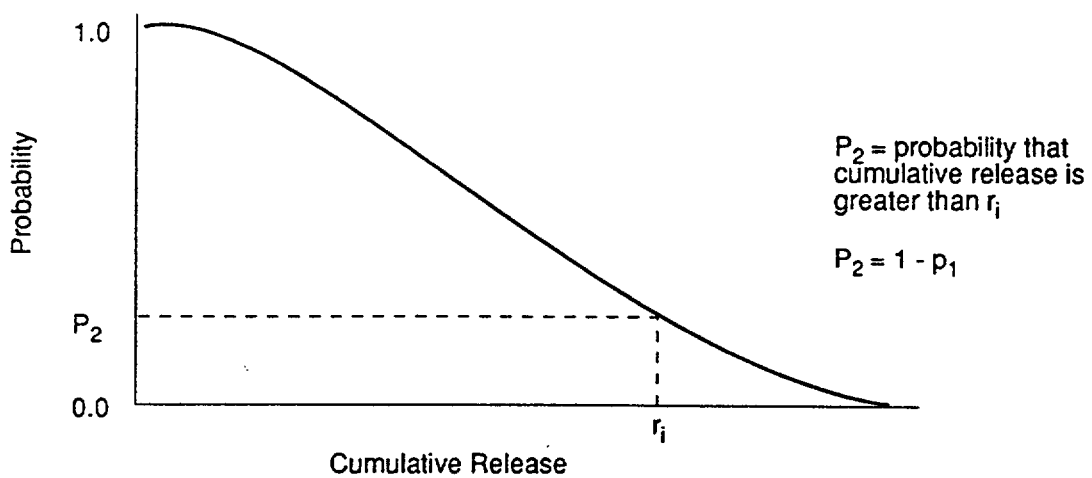
FIGURE 2-2  
RIP SIMULATION LOGIC  
ARGONNE/MODEL DEVELOPMENT



a) Probability density function



b) Cumulative distribution function



c) Complementary cumulative distribution function

FIGURE 2-3  
PRESENTATION  
PROBABILISTIC RESULTS  
ARGONNE/MODEL DEVELOPMENT

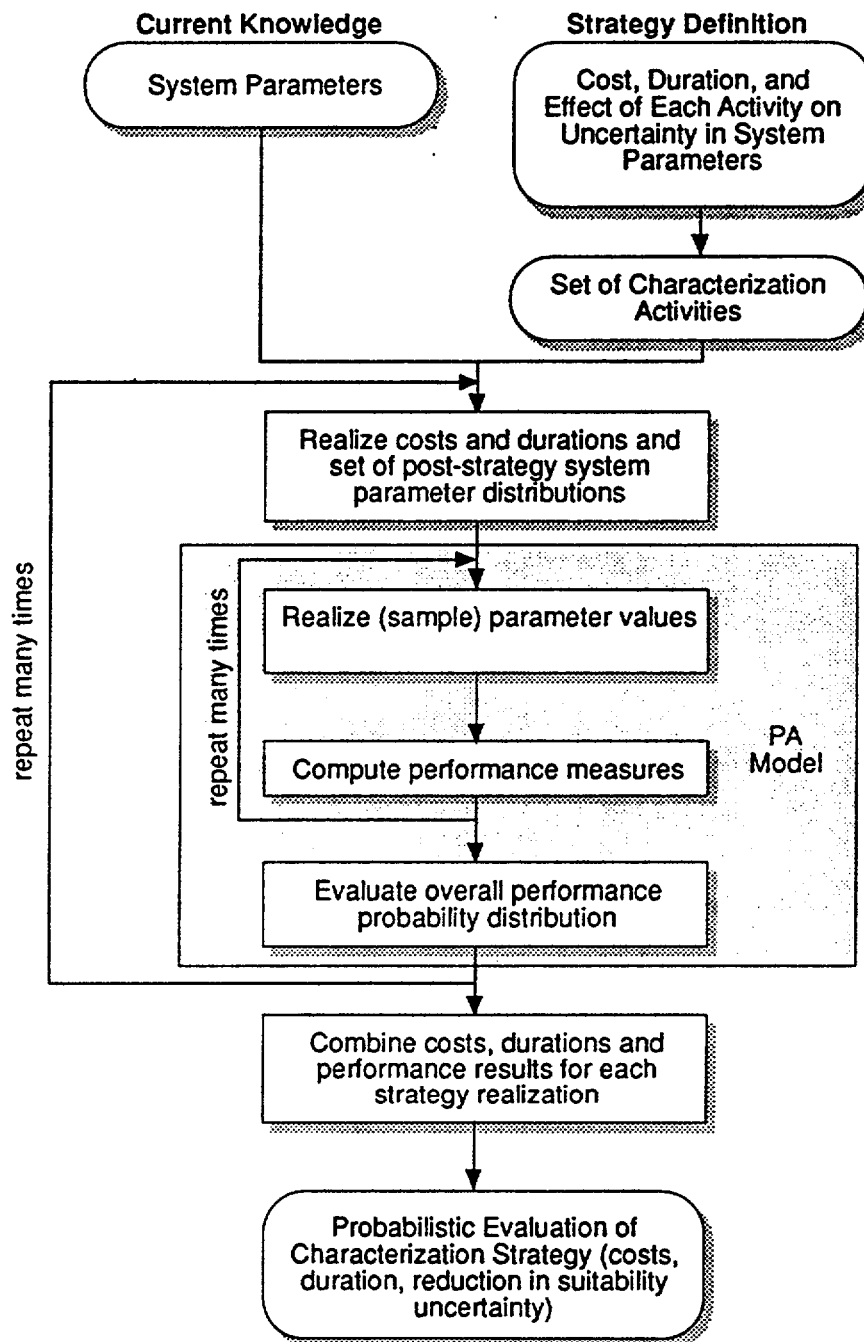


FIGURE 2-4  
OVERVIEW OF RIP  
STRATEGY EVALUATION  
ARGONNE/MODEL DEVELOPMENT

Given a current ("prior") data set of system parameters, RIP uses this information regarding activities to realize a large number of sets of post-strategy system parameter distributions ("posteriors"). That is, the current data set (in particular, the uncertainty in specified system parameters) is modified to reflect the effects of carrying out the activities within a characterization strategy. Because we can not predict precisely how a parameter's distribution will change due to an activity, but can only estimate how the *magnitude of the uncertainty* will be affected (i.e., the distribution shape may narrow, but we can't predict whether it will shift left or right), the results of a particular strategy can only be described probabilistically. In other words, a particular strategy defines a probability distribution of *sets of system parameter distributions*. The RIP strategy model samples this distribution of sets many times.

Based on the parameter distributions within a given realized set, RIP produces a probabilistic evaluation of repository performance measures. Each realized set of parameter distributions produces unique probabilistic performance results (e.g., in the form of a CCDF of cumulative release). Figure 2-5 illustrates this schematically. By realizing many different sets of parameter distributions for a particular strategy, producing a probabilistic performance result for each, and combining this suite of results in an appropriate manner, it is possible to determine probabilistically how that particular strategy will effect the evaluation of site suitability. Figure 2-6 presents an example of how the results shown in Figure 2-5 can be combined to evaluate the effect of the strategy.

Both the performance assessment model and the strategy evaluation model are discussed in detail in the RIP Theory and Capabilities document (GAI, 1992).

## 2.2 Parameter Definition and the Flexibility of the User Interface

The cornerstone of the RIP methodology, which it is critical to understand in order to take full advantage of the software, is the concept that model input parameters can be defined by the user with a great deal of flexibility. In particular, *a given system parameter can be specified as a constant, a stochastic (i.e., represented as a probability distribution), or as a function of other parameters* (which themselves can be constants, stochastics, or functions).

Representing system parameters as *stochastics* allows the RIP user to specify the degree of *uncertainty* in a particular parameter. For example, if the current level of knowledge regarding a particular system parameter (such as an elemental solubility) is such that it is only possible to specify its value within certain limits (e.g., greater than  $10^{-4}$  g/m<sup>3</sup> but less than  $10^{-1}$  g/m<sup>3</sup>), it would be most appropriate to specify this parameter as a probability distribution (e.g., a log-uniform distribution ranging between  $10^{-4}$  and  $10^{-1}$ ). RIP allows the user to choose from a wide variety of probability distributions (e.g., normal, log-normal, triangular, beta, gamma, discrete, etc.) for a stochastic parameter. For each Monte Carlo realization of the repository system, all of the stochastic parameters are sampled from their specified distributions.

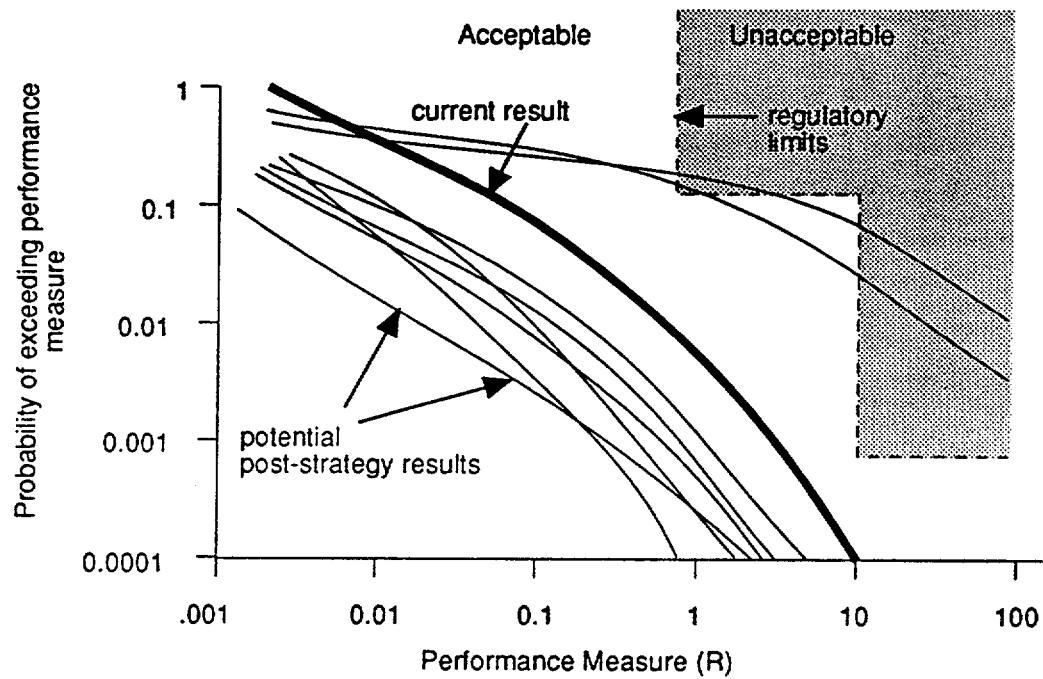


FIGURE 2-5  
 SUITE OF PERFORMANCE  
 RESULTS FOR STRATEGY A  
 ARGONNE/MODEL DEVELOPMENT

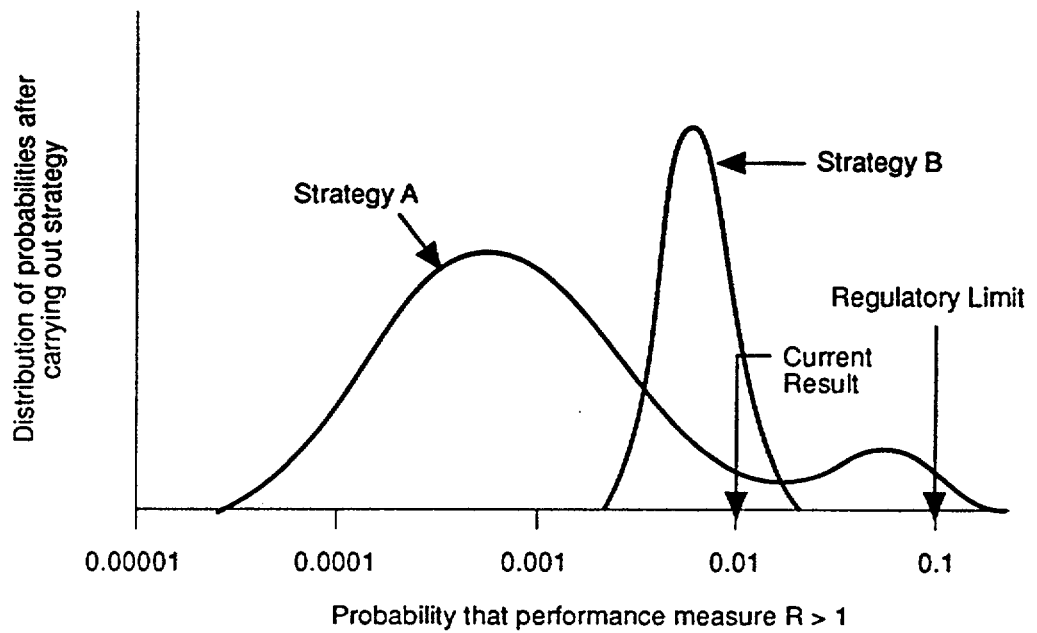


FIGURE 5  
**PROBABILITY OF POST-STRATEGY  
 PERFORMANCE RESULTS**  
 ARGONNE/MODEL DEVELOPMENT



The ability to represent system parameters as *functions* of other parameters imparts to the user the ability to readily add detail to any given system parameter or process represented by RIP. The user can even create and specify processes and parameters which are not explicitly incorporated into RIP. In a sense, the RIP program is similar to a spreadsheet: while it contains a large amount of built-in logic and calculational capabilities, the problem that is solved is entirely defined by the user. That is, RIP has been developed such that it is relatively free of assumptions regarding the details of waste package behavior and radionuclide transport processes. Similarly to a spreadsheet user, the RIP user can define a very simple model, or a very complex one. Hence, RIP consists of a basic computational framework representing the controlling processes, but is intentionally flexible such that the user can represent the processes with as much detail as desired. This allows the conceptual model being represented by RIP to be continuously and easily modified as more information becomes available.

The user interface of the software is designed such that the user literally types in the desired function. In addition to standard mathematical operators (e.g., SIN, COS, MAX, ERF, LOG), the user can define functions using relational operators (e.g., >, <, =) and IF, THEN logic. Such flexibility allows the user to easily modify and add detail to the conceptual and computational model *without having to make changes to the software* (i.e., without modifying and compiling the source code).

As an example, suppose that instead of simply representing an elemental solubility as a stochastic parameter (as illustrated above), the user wished to incorporate a solubility model which explicitly accounted for the effects of temporally or spatially variable chemistry (e.g., pH) conditions. To accomplish this, the user could define a new parameter, called *pH*. This parameter could be defined as a constant, a stochastic, or a function (of other parameters or of time). It could also be assigned a random spatial variability throughout the repository. The elemental solubility could then be described as a function of this parameters. For example,

$$SOL_n = (A \times pH) + B \quad (2.1)$$

where  $SOL_n$  is the solubility of element  $n$  and  $A$  and  $B$  are constants or additional user-defined parameters.

Use of the *Parameters Module*, the software component which implements the concepts described above, is discussed in detail in Chapter 4.

## 2.3 Software Structure and Hardware Requirements

The RIP software consists of two independent units, referred to here as the *front end* and the *back end*. The front end comprises the user interface, while the back end contains the actual computational algorithms. Technically, the front end and the back end can be thought of as two separate programs. The basic structure of the software is illustrated in Figure 2-7. The user defines the problem to be simulated via the window-driven user interface of the front

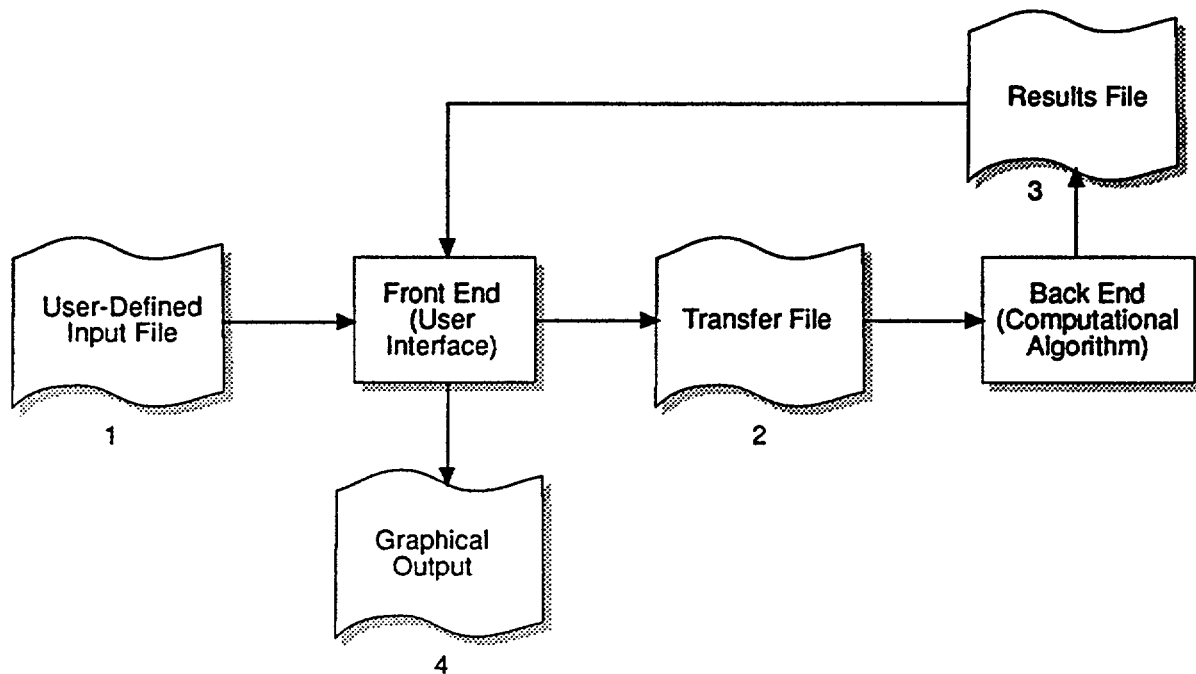


FIGURE 27  
**RIP FRONT END/BACK  
STRUCTURE**  
ARGONNE/MODEL DEVELOPMENT

end. The front end then communicates this information to the back end via a *transfer file*. The transfer file is read by the back end, which then carries out the actual computations and produces a *results file*. The results file is then read by the front end which processes the results, as directed by the user, into a variety of graphical and text formats.

The front end and the back end operate independently and can be implemented on different machines, with the transfer files and the result file being transferred between the machines electronically (e.g., via a network), or manually (e.g., on a magnetic disk).

The front end is written in Turbo Pascal (Version 6.0) and is intended to be run on a personal computer (PC). The PC should have at least 640K RAM (although the actual amount of memory required is a function of the specific problem to be simulated, and more than 640K may be required for larger data sets).

The back end is written in ANSI standard C. As a result, the back end is very portable and can be run on a variety of computers. Although the back end will run on a PC, because of the memory barriers of DOS, the size and complexity of applications run under DOS will be seriously limited. Use of a UNIX based machine is highly recommended. At the very least, a PC with an 80286 processor, a math co-processor, and 640K RAM is required to run the RIP back end. In any case, because of the computationally intensive nature of the program (i.e., use of the Monte Carlo approach), running the back end on a PC may be unacceptably slow.

## 2.4 Software Installation and Implementation

The distribution disk for RIP includes a file called READ.ME. This is an ASCII text file which contains a list of all the files on the distribution disk and complete instructions for installing both the back end and front end on a PC or any other machines. Note that in addition to program files, the distribution disk contains the input file for a simple sample problem.

Starting and using the front end of RIP is discussed in detail in the following sections. Because the front end comprises the user interface, supporting data input and post processing of results, this User's Guide is, for the most part, a description of the front end. As will be described subsequently, the back end of RIP is invoked using a single command line. As discussed above, the back end automatically reads in the transfer file created by the front end and creates a results file. No other user action is required.

It should be noted that the time scales required to run the front end and the back end will typically be quite different. The front end is an interactive program in which the user specifies data (and manipulates results). While definition of a model can involve weeks or months of data review and analysis, and days to enter and verify the input data, creation of the transfer file is relatively rapid (typically minutes on a powerful PC). The back end, however, is the computational engine for RIP. For large complex simulations with complex input and large numbers of realizations, execution of the back end and the creation of a results file may take many hours even when run on powerful machines.

## 2.5 Model Applicability

Before describing RIP in detail, it is useful to briefly mention the applicability of this simulation tool. As has been pointed out above, RIP was specifically developed for application to the proposed high-level radioactive waste repository at Yucca Mountain, Nevada. *Its use, however, is not necessarily limited to this particular site.* This is because, for the most part, RIP is designed in a very flexible manner. In simple terms, *RIP is a simulation tool designed to probabilistically model the release of radionuclides from buried waste packages and the subsequent transport of those radionuclides through the environment.*

Hence, although a thorough understanding of the theory and capabilities of RIP is required in order to apply it to any site, for the most part its flexibility should allow application to a variety of sites. Of course, for some applications, it may prove more efficient and accurate to modify the RIP software to better represent some situations.

The user will note that RIP is currently configured such that output is presented in terms of U.S. regulatory limits (e.g. cumulative radionuclide releases are normalized to U.S. EPA limits). This may not be appropriate for application to other sites. The user can choose, however, not to normalize results so that they are output directly in terms of radionuclide mass.

## 2.6 Error Reporting

Although RIP has undergone a variety of verification testing aimed at ensuring that the major algorithms of the software are properly implemented, due to the complexity of the software, small errors are likely to remain. In addition, as RIP is applied, users may desire features or modifications which will improve the software. Appendix E of this report contains the forms for reporting software problems and/or suggesting modifications.

### 3. INTRODUCTION TO THE RIP USER INTERFACE

In this chapter, an introduction to the user interface (i.e., the front end) is presented. The organizational structure of the input windows is discussed, and basic instructions for browsing through the windows, entering data, and saving input files (and screens), are described. The on-screen help utility is also described.

Details of the specific input windows are discussed in subsequent chapters.

Although images of the input screens which are discussed are presented in this document as figures, it may be useful to install and run RIP at this point in order to follow along as various features of the user interface are discussed.

#### 3.1 Starting RIP and Manipulating Input Data Files

After properly installing the software as described in Section 2.4, the front end program can be started by typing "RIP". The following screen will appear:

```
[ ] Repository Integration Program (Rev. 1.00 )
  General      PA Model      Strategy Model      Results
Available Memory 261520      file date:

Repository Integration Program
(c) Golder Associates Inc., 1991, 1992.

THIS SOFTWARE IS UNDER DEVELOPMENT, AND HAS NOT BEEN COMPLETED,
DEBUGGED, VERIFIED, OR VALIDATED FOR PRACTICAL APPLICATIONS.
THE SOFTWARE IS PROVIDED FOR INFORMATION PURPOSES ONLY, AND
NO WARRANTY IS EXPRESSED OR IMPLIED REGARDING ITS ACCURACY,
FUNCTIONALITY, OR APPLICABILITY.

Any permission granted for the use of this program includes
the explicit condition that all liabilities and responsibilities
for any results obtained from its use are exclusively
those of the individual user or organization.
No responsibility is assumed or implied for user support.

Press any key to continue:
```

RIP will then prompt the user for a file name. This refers to the name of a file to which all of the input information will be saved. If the user specifies a new file name (i.e., the name of a file which does not currently exist in the directory), RIP will inform the user that that file does not exist, giving him/her the choice to specify a different file name or create a new file. All subsequent changes made during that session will be saved to the specified file. If the user specifies the filename without an extension, the extension ".rp" is automatically added.

If the user specifies a file which already exists, RIP will read this file. The user interface can then be used to edit the information contained in the file. The user can read an existing file automatically when starting RIP by directly specifying the filename in the command. For

example, "RIP test" or "RIP test.rp" will automatically load the input file TEST.RP.

After specifying the filename, RIP will prompt for any desired changes to the data file description:

```

[ ]-----Repository Integration Program (Rev. 1.00 )-----
      General      PA Model      Strategy Model      Results
-----
Available Memory 237200 ----- file date:
Change the data file description?
User's Guide Tutorial

```

This short description or title is subsequently displayed at the top of the user interface screen, along with the file date (of the last update), and the amount of random access memory (RAM) currently available on the machine.

While in the user interface, the user can save the current input file by hitting <Alt>-<S>. Note that the user is reminded of this and several additional key combinations (which will be discussed subsequently) at the bottom of every input screen (These prompts, however, are not reproduced on the screens printed in this manual). RIP will then prompt the user for the file name and file description, allowing new files to be created. As a safeguard, RIP will warn a user if he/she has made some changes which have not yet been saved, and tries to exit the program. *Regardless of this safeguard, it is good practice to periodically save large input files while editing.*

## 3.2 Basics of the User Interface

### 3.2.1 Menu Structure

After specifying the name of the data file, the main window will appear on the screen:

```

[ ]-----Repository Integration Program (Rev. 1.00 )-----
      General      PA Model      Strategy Model      Results
-----
Available Memory 228304 ----- file date: 01-14-92 08:21:31
User's Guide Tutorial

```

The RIP user interface is structured into a *hierarchy of menus* which provide access to a variety of *interactive data input windows*. The overall menu structure is illustrated graphically in Figure 3-1 and in tabular form in Table 3-1.

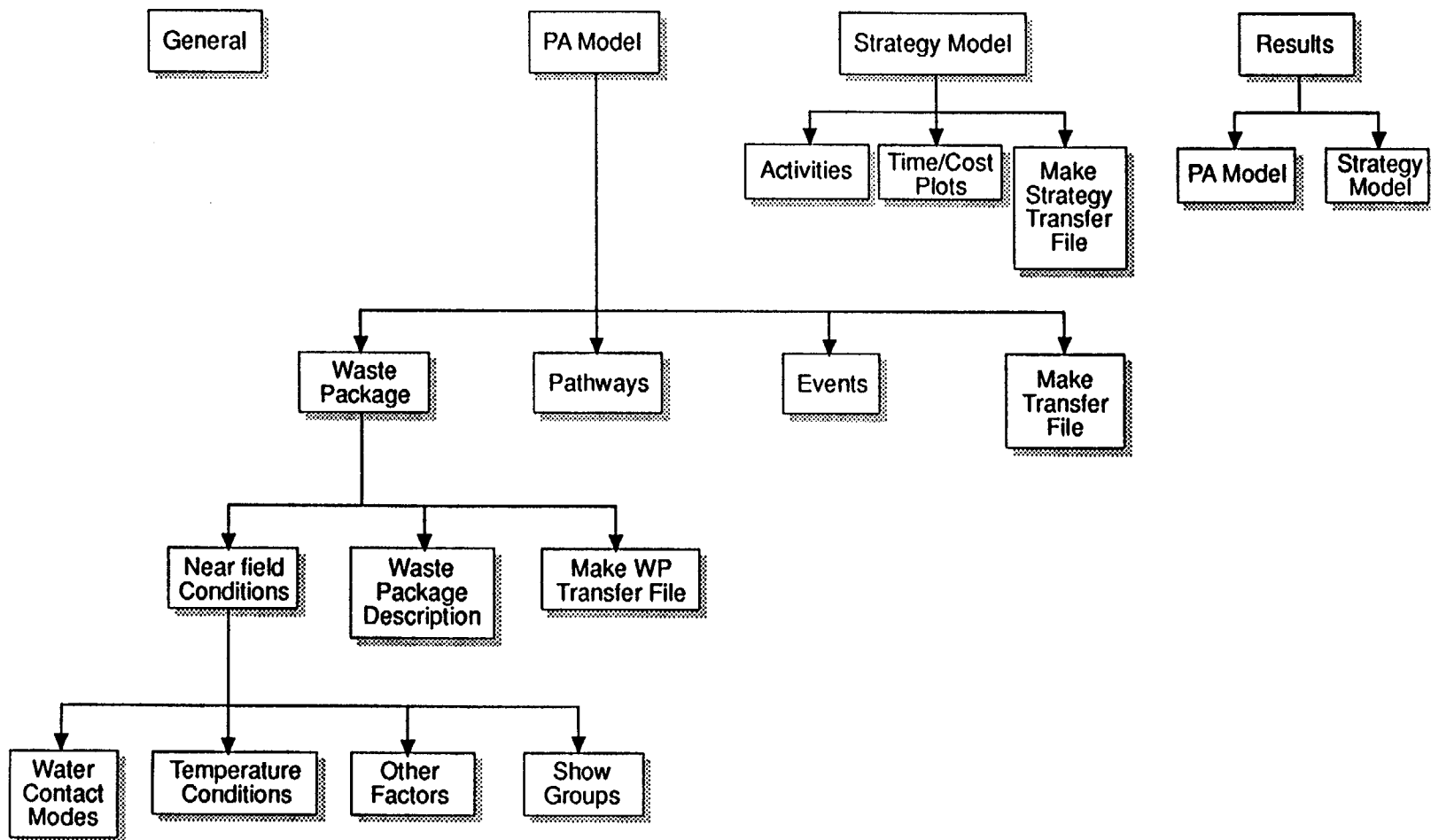
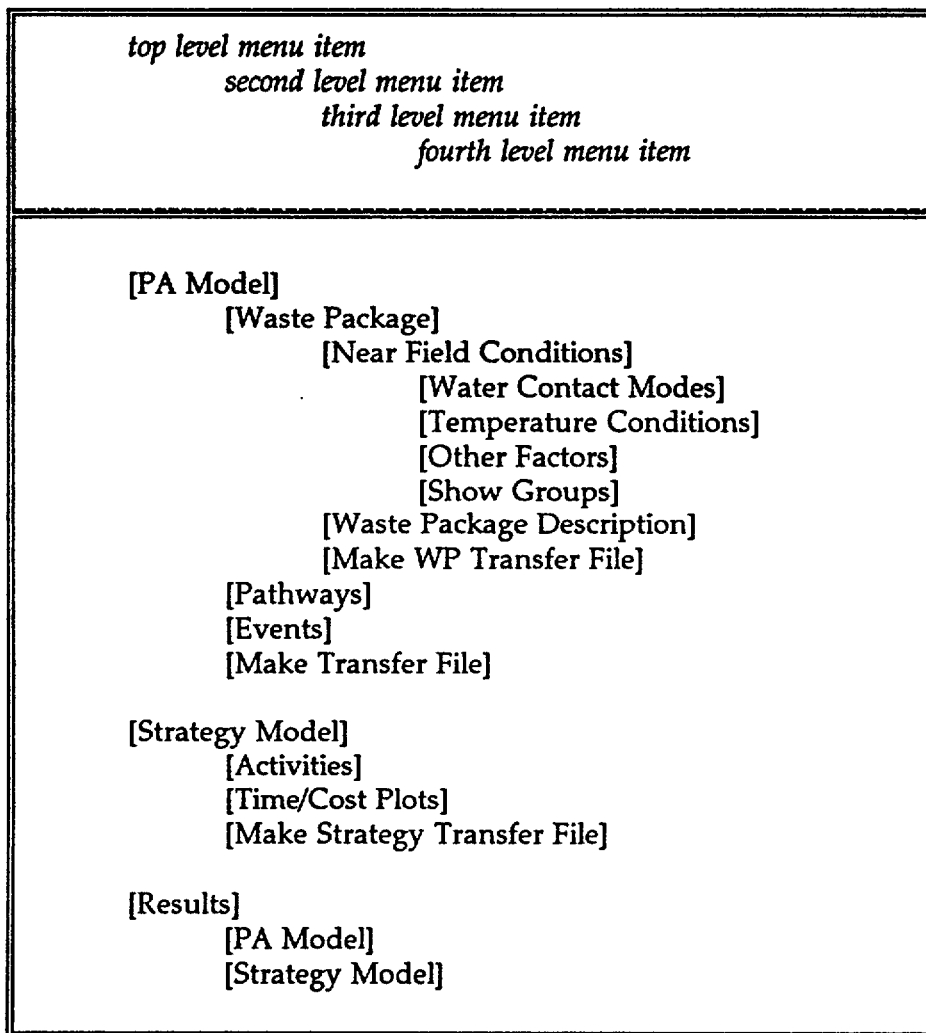


FIGURE 3-1  
RIP MENU STRUCTURE  
ARGONNE/MODEL DEVELOPMENT

TABLE 3-1  
RIP Menu Structure



Within a menu, the user can move between choices using the right and left arrow keys (or, alternatively, <TAB> and <SHIFT>-<TAB>). The user selects a particular choice by moving to the choice and hitting <ENTER>. Upon selection, either a lower level menu or an input window will appear. *The user can move backward through the structure (i.e., from lower level windows or menus to higher levels) by using the <ESC> key.* Hitting <ESC> from the main (highest level) menu exits the program.

If available, the user can also move between menu items with a mouse. Menu items are selected by clicking on the item. Note that using a mouse, the user can move backward through the menu structure by clicking on the "[■]" symbol in the upper left hand corner of each window. Clicking on this symbol is equivalent to hitting the <ESC> key. (Note that this symbol is represented by "[ ]" in the windows reproduced within this report.)



As the following menu items are briefly summarized below, it is recommended that the user browse the user interface to become familiar with the menu and window structure, and the techniques for moving around within the interface.

As illustrated above, the top level menu consists of four choices:

- [General] provides access to an input window for specifying several general inputs pertaining to printer setup, echoing of input files and global parameters regarding the Monte Carlo sampling technique to be used.
- [PA Model] provides access to input windows describing the integrated performance assessment model;
- [Strategy Model] provides access to input windows describing the strategy evaluation model;
- [Results] provides access to the post-processor.

These menu options are discussed in somewhat greater detail below.

#### [General]

Selection of [General] provides access to a single data input window. The [General] data input window is discussed in detail in Section 3.4

#### [PA Model]

Selection of [PA Model] provides access to a second-level menu consisting of four choices pertaining to the total system performance assessment model:

- [Waste Package] provides access to input windows pertaining to the waste package behavior and radionuclide release component model;
- [Pathways] provides access to input windows pertaining to the radionuclide transport pathways component model;
- [Events] provides access to input windows pertaining to the disruptive events component model;
- [Make Transfer File] provides access to a window which is used to specify general simulation information and the form of the output, and to actually run the front end program for the performance assessment model and create a transfer file for the back end.

[Pathways], [Events] and [Make Transfer File] provide direct access to data windows. [Waste Package] provides access to a third-level menu which further subdivides waste package input:

- [Near Field Conditions] provides access to windows for specifying the environmental conditions at the repository;
- [Waste Package Description] provides access to windows for describing the waste package failure and radionuclide release mechanisms;
- [Make Transfer File] provides access to a window which is used to specify general simulation information and the form of the output, and to actually run the front end program for the waste package behavior and radionuclide release component of the performance assessment model and create a transfer file for the back end.

[Waste Package Description] and [Make Transfer File] provide direct access to data windows. [Near Field Conditions] provides access to a fourth level menu which further subdivides the near field conditions input: [Water Contact Modes], [Temperature Conditions], [Other Factors] and [Show Groups].

The details of the performance assessment model input windows are discussed in Chapter 5.

#### [Strategy Model]

Selection of [Strategy Model] provides access to a second-level menu consisting of three choices pertaining to the strategy evaluation model:

- [Activities] provides access to input windows for defining activities within a characterization strategy;
- [Time/Cost Plots] provides access to windows for plotting probability distributions of cost and duration for the specified strategy;
- [Make Strategy Transfer File] provides access to a window which is used to specify general simulation information and the form of the output, and to actually run the front end program for the strategy evaluation model and create a transfer file for the back end.

The details of the strategy evaluation model input windows are discussed in Chapter 6.

#### [Results]

Selection of [Results] provides access to a second level menu consisting of two choices pertaining to the post-processing of results:

- [PA Model] provides access to windows for evaluation and processing of results generated by the performance assessment model;

- [Strategy Model] provides access to windows for evaluation and processing of results generated by the strategy evaluation model.

The details of the results input windows are discussed in Chapter 7.

### 3.2.2 Using Data Input Windows

As described above, the menu structure provides access to the data input windows. For example, the following screen shows the data input window displayed by selecting the menu items

[PA Model]  
 [Waste Package]  
 [Waste Package Description]:

Repository Integration Program (Rev. 1.00)			
General	<u>PA Model</u>	Strategy Model	Results
User's Guide Tutorial			
[ ]			
<u>Waste Package</u>	Pathways	Events	Make Transfer File
[ ]			
Near Field Conditions	<u>Waste Package Description</u>	Make WP File	
[ ]			
Waste Package Database Editor			
ID : SPENT	Package 1 of 2		F1=Prev
DESC: Spent Fuel			F2=Next
			F3=Add
			F4=Del
Number of Packages	: 0		
Mass Waste per Package (kg):	0	as MTHM :	0
Waste Burnup (MWd/MTHM):	0		
Is mass transfer/bound exposure independent of failure mode? : Yes			
Mass Transfer			
Exposure			
Primary Container Failure Definition		Plotting Options	
Secondary Container Failure Definition		» Rewetting time : 0	
Inventory		» Cont. fail time : 0	
		Plot axes : automatic	

Note that the windows overlap such that the selected path to the current window is clearly visible on the screen.

Many windows for RIP (such as the waste package window illustrated above) are used to specify the details pertaining to a particular unit or structure within the model (e.g., a waste package type, a transport pathway, a disruptive event, a radionuclide). The RIP user interface is designed such that any number of different types of a particular unit or structure can be specified. For example, the user may want to define 3 different waste package types, 12 different pathways, or 3 different disruptive events. To accommodate this, those windows pertaining to these kinds of units or structures can have multiple records or *pages*. Within a window, pages can be added, deleted, or browsed using specified function keys (i.e., F1, F2, F3, F4).

For example, in the screen shown above, the unit of interest is the waste package type. This particular screen indicates in the upper right hand corner of the window ("Package 1 of 2")

that two waste package types have been defined (each waste package type being represented on a different page of the window). This particular screen shows page 1 of the window. As indicated in the upper right hand corner, the function keys <F1> and <F2> can be used to browse the pages. Hitting <F2> will bring up page 2 of the screen, which defines waste package type 2:

[ ] Repository Integration Program (Rev. 1.00 )			
General		PA Model	Strategy Model
User's Guide Tutorial			
[ ]			
Waste Package		Pathways	Events
Make Transfer File			
[ ]			
Near Field Conditions		Waste Package Description	Make WP File
[ ] Waste Package Database Editor			
ID : HLDW		Package 2 of 2	
DESC: High Level Defense Waste Glass		F1=Prev F2=Next F3=Add F4=Del	
Number of Packages : 0			
Mass Waste per Package (kg): 0		as MTHM : 0	
Waste Burnup (MWD/MTHM): 0			
Is mass transfer/bound exposure independent of failure mode? : Yes			
Mass Transfer			
Exposure			
Primary Container Failure Definition		Plotting Options	
Secondary Container Failure Definition		» Rewetting time : 0	
Inventory		» Cont. fail time : 0	
		Plot axes : automatic	

Pages can be added or deleted using the <F3> and <F4> keys, respectively.

As will be described in detail in subsequent sections, function keys can also be used for other purposes (e.g., showing lists, calling up graphics). The available function keys and their uses are always shown on the window.

Within an input window, the user can move between input fields using the arrow keys (or <TAB> and <SHIFT>-<TAB>). A mouse can also be used to move around a particular window. Note that the mouse only moves the cursor within an active window, and can not select items in underlying windows.

Moving through an input window places the cursor in one of four types of input fields: a *descriptive field*, a *data field*, an *option field*, or an *access field*.

A descriptive field is used to provide a description or an identification for a particular process, event or item defined by the window. An *alphanumeric name* or *short description* is entered. In a data field, either a *number* or a *parameter name* is entered. In an option field, the <PgDn> or <PgUp> keys are used to toggle through a limited number of input options. When the cursor is moved to an option field, the field will blink on the screen. An access field either provides access to a pop-up window or causes a particular action to be carried out (e.g., creation of a file).

Within a particular field, the left and right arrow keys, and the <Home> and the <End> keys can be used for movement. The <Ins>, <Delete> and <Backspace> keys can also be

used.

In the input window illustrated above, the first two data fields ("ID" and "DESC") are descriptive fields. The next several (through "Waste Burnup") are data fields. The next field, which answers a question regarding mass transfer/bound exposure, is an option field. In this particular case, there are simply two options, Yes or No. The input fields starting with "Mass Transfer" and ending with "Inventory" are access fields which call up lower level input windows. For example, selecting "Mass Transfer" and hitting <ENTER> brings up the following screen:

Repository Integration Program (Rev. 1.00)			
General	PA Model	Strategy Model	Results
User's Guide Tutorial			
[ ]			
Waste Package	Pathways	Events	Make Transfer File
[ ]			
Near Field Conditions	Waste Package Description	Make WP File	
[ ]			
Waste Package Database Editor			
ID [ ]	MASS TRANSFER		
DE	Repository Infiltration Rate (m/yr): INFIL	(0.005)	rev
	Water Volume Contacting Matrix (m <sup>3</sup> m <sup>3</sup> ): VOL	(1)	ext
Nu	Effective Catchment Area (m <sup>2</sup> ): CATCH	(50)	dd
Ma	Geometric Factor for Diffusion (m): OMEGA	(1)	el
Wa			
Is mass transfer/bound exposure independent of failure mode? : Yes			
Mass Transfer			
Exposure			
Primary Container Failure Definition		Plotting Options	
Secondary Container Failure Definition		» Rewetting time : 0	
Inventory		» Cont. fail time : 0	
		Plot axes : automatic	

Use of access fields is self-explanatory. Moving to the desired field and hitting <ENTER> provides access to the lower level input window or carries out the specified action (access fields are selected by double clicking when using a mouse). Use of descriptive fields and option fields is also straightforward. After typing in characters for a descriptive field, or selecting the desired option in an option field using the <PgDn> or <PgUp> toggle, the <ENTER> key (or an arrow key) enters the data and moves the cursor down to the next input field.

Data fields represent the major form of data input. There are actually two types of data fields: *constant data fields* and *parameter data fields*. Constant data fields (which are green on a color display) only accept numbers. After entering the desired number, the <ENTER> key (or the arrow keys) move the cursor down to the next input field. Parameter data fields (which are yellow on a color display) accept both numbers and parameter names, and are discussed in greater detail below.

### 3.2.3 Specifying Parameters

Recalling the discussion in Section 2.2, the cornerstone of the RIP methodology is the fact that the software allows most input to be defined by the user as constants, stochastics, or functions of other parameters. This is carried out by specifying an alphanumeric parameter name (as opposed to a number) in a parameter data field.

Parameters are defined and edited through the "parameters module" of RIP, which is discussed in detail in Chapter 4. The parameters module contains a data base of parameters defined by the user. The user may define as many parameters as deemed necessary. Defining a parameter consists of specifying a name, a type (e.g., constant, stochastic, function), and a form (e.g., a value, a specified distribution, or a specified function or dependency).

The parameters module is entered automatically by typing a parameter name in a parameter data field within any input window and hitting <ENTER>. If the parameter name specified in the parameter data field exists in the parameters module data base, the parameters module editing window is displayed. As an example, consider the "mass transfer" window shown above. In this case, INFIL happens to be the name of a previously defined parameter. Moving the cursor to this parameter data field and hitting <ENTER> provides access to the parameters module and allows that parameter to be edited:

Repository Integration Program (Rev. 1.00)	
General	PA Model
Strategy Model	
User's Guide Tutorial	
Results	
Waste Package Pathways Events Make Transfer File	
Parameter Editing	
Identifier : INFIL Description: repository level infiltration rate Parameter Type : Stochastic ( log- Uniform : from 0.0005 to 0.05 ) Edit Parameter Definition	
F7=Display Dist. F8=Expected Value	
Mass Transfer	
Exposure	
Primary Container Failure Definition	Plotting Options
Secondary Container Failure Definition	» Rewetting time : 0
Inventory	» Cont. fail time : 0
	Plot axes : automatic

Note that <ESC> exits the parameters editing window and returns the user to the original input window. For all parameter data fields, RIP displays the expected value of the parameter to the right of the field in parentheses (if the parameter is constant, "Constant" is displayed).

The details of parameter editing are provided in Chapter 4. As will be discussed in detail in that chapter, in addition to entering the parameters module as described above, the parameters module can be entered directly from anywhere within the user interface using the hot-key <Alt>-<P>.

### 3.2.4 The On-Screen HELP Utility

RIP provides an on-screen HELP utility which provides explanations, warnings, and reminders to the user regarding various input windows or fields. The HELP utility is entered automatically by hitting <Alt>-<H>. Note that the user is reminded of this at the bottom of every screen (although this is not shown on the screens reproduced for this report).

(\*\*\*\*\*discuss details of HELP\*\*\*\*\*)

### 3.2.5 Printing and Saving Input or Output Screens

For documentation purposes, the user may wish to print or save a screen capture of the input windows within the user interface. As will be discussed in subsequent sections, RIP provides several more powerful ways for documenting input (e.g., creating an echo file of the entire data set, discussed in Section 3.4; creating a listing of the parameter data base, discussed in Section 4.2) this utility may be useful for some users. Note that the windows reproduced in this document were created using this utility.

Printing and saving of screens is invoked automatically using the following key combinations:

- <Ctrl>-<P> sends the current screen to the printer. The user must specify the type of printer under the [General] window, discussed in Section 3.4.
- <Ctrl>-<F> sends the current screen to an ASCII file named SCREENS.DAT. Screens are added sequentially to the file, and are not overwritten.

Only *text screens* can be printed and saved in this manner. There exist, however, several mechanisms to print and save *graphics screens*, which may contain graphical representations of the input or output:

- RIP provides a mechanism (discussed in later sections) by which the user can create a plot file in the HPGL or HPGL2 format. This file can then be imported into a compatible program (e.g., AUTOCAD, WordPerfect), or, in some cases, copied directly to an HP plotter (i.e., an HPGL2 file can be copied directly to an HP Laserjet III).
- When running the front end on a PC with DOS 5.0, it is possible to send graphics screens directly to the printer by typing <Shift>-<PrintScreen>. In order for this utility to be active, the GRAPHICS command should be added to the AUTOEXEC.BAT file. The type of printer must be specified as an argument for the GRAPHICS command (e.g., GRAPHICS LASERJETII). In addition, several options for the command are available. The user should refer to the DOS reference manual for more details.
- The user can use a screen capture utility (e.g., WordPerfect GRAB or HiJaak) to capture a screen for import into a compatible graphics program.

### 3.2.6 Units

RIP is a dimensional model. That is, it requires that data be input using a specific set of data units. The actual units which are required for any given input field are specified in parentheses next to the field.

### 3.3 Entering Data and Applying RIP

For the most part, data is entered into RIP and a transfer file can be created by sequentially stepping through the menus and input windows of the user interface. The user begins by entering data in the input windows under [PA Model]. When editing an existing data input set, the user can move through the various input windows in any manner desired. Because of the interdependency of several of the input windows describing the PA model, however, when first defining a problem, the user must move through the windows in a specified order (as will be described in detail in Section 5). If just running the performance assessment model, the user can then create a transfer file from a window under [PA Model]. The back end program, which reads the transfer file and creates a results file, must then be executed. Post processing of results generated by the back end is carried out through the windows under [Results] in the user interface.

When evaluating alternative characterization strategies, in addition to the PA Model input, additional input must be entered in the [Strategy Model] input windows, where a transfer file can be created. Again, the transfer file must be read by the back end program, which creates the results file. Post processing of results is again carried out via the windows under [Results] in the user interface.

Note that while entering data in any part of RIP, the parameters module can be entered in several ways (as will be described in detail below) to define any parameters specified in the input windows.

We have briefly discussed above three of the four top level menu items in the user interface: [PA Model], [Strategy Model], and [Results]. The fourth top level menu item, [General], provides access to a window containing three input fields. Two of these fields pertain to user utilities. One allows the user to specify the type of printer to which the user interface is attached (allowing input screens to be captured and printed). A second allows the user to create an ASCII echo file of the entire input data set, which is a valuable documentation utility. The third input field allows global specification of a parameter which controls the efficiency of the Monte Carlo sampling scheme implemented in RIP.

The RIP user interface and the various input windows are discussed in detail below. The [General] input window is described in Section 3.4. The parameters module is then discussed in Chapter 4. Chapters 5 and 6 describe the performance assessment model and the strategy evaluation model, respectively. Finally Chapter 7 describes the post-processing of results.

### 3.4 Editing the [GENERAL] Input Window

The input window accessed by selecting [GENERAL] from the top level menu consists of three input fields:



```

[ ] Repository Integration Program (Rev. 01 )
  General  PA Model  Strategy Model  Results
  [ ] General Data  Users's Guide Tutorial  file date: 02-26-92 09:27:11
Printer Type      : Laserjet II
Re-set importance-sampling to: Unchanged
Use Latin-Hypercube sampling?: Yes
Repeat sampling sequences?   : Yes
Create Echo File
Available memory 169728

```

**Printer Type** - This is an option field (it blinks). It is required in order to print the current screen (using <CTRL>-<P>). The current version of RIP provides the user with three choices: Epson LX, HP Laserjet, and HP Laserjet II. Press <PgUp> or <PgDn> to select the appropriate printer.

**Reset importance sampling to** - This is also an option field. As discussed in detail in Appendix C of the RIP Theory and Capabilities Manual, RIP utilizes an *importance sampling scheme* to increase the efficiency of the Monte Carlo sampling process. The user can choose the degree to which importance sampling is to be used (i.e., the amount of sampling bias) on any given parameter. As will be discussed in detail in Chapter 4, the level of biasing for each parameter can be specified separately within the parameters module. *This particular input field in the [General] input window allows the user to set the level of biasing for all parameters globally.* This field offers the user five choices: Unchanged, No Bias, Little Bias, Moderate Bias, and Lots of Bias. Unchanged is the default value, and has no effect on the sample biasing (i.e., it leaves the specified sample biasing unchanged). The other four options globally change the bias for all parameters to the specified level. No Bias results in straight Monte Carlo sampling with no importance sampling. The other three choices represent increasing levels of importance sampling. Biasing is discussed in detail in Section 4.3.3.

**Use Latin-Hypercube sampling** - This is an option field which specifies whether or not Latin-Hypercube sampling will be used within the Monte Carlo sampling scheme. It is answered Yes or No. Latin-Hypercube sampling is discussed in detail in Appendix C of the RIP Theory and Capabilities Manual. Note that the use of Latin-Hypercube sampling can be used whether or not importance sampling is being implemented.

**Repeat sampling sequences** - This is an option field which specifies whether a repeatable random number sequence is to be used. Answering No creates a new random number sequence for sampling. Answering Yes results in the use of a repeatable random number sequence (allowing simulations to be repeated exactly).

**Create Echo File** - This is an access field which allows the user to create an echo of the current input file. This is an ASCII file which summarizes all input, providing documentation for the run. A pop-up window prompts the user for the name of the file to be created. Specifying a name and hitting <ENTER> immediately creates the new echo file.

Note that the user can enter the [GENERAL] input window directly from the top level menu

or indirectly from anywhere within the user interface by pressing <Alt>-<G>.

#### 4. DEFINING PARAMETERS

As described in Sections 2.2 and 3.2.3, the *parameters module* is the cornerstone of the RIP methodology, and consists of a library of computer procedures designed to assist in numerical simulations of complex, stochastic systems. The fundamental concept is that the symbols used to express the parameters of an equation can refer to a variety of alternative types of parameters. A parameter can be represented as:

- 1) a constant;
- 2) a stochastic (uncertain) variable; or
- 3) a function of other parameters.

The RIP user could, for example, define a parameter Y using the following equation:

$$Y = A * X + B$$

where:

- |   |  |
|---|--|
| A | is a stochastic parameter, with a log-normal probability distribution; |
| X | is a function of some additional parameters (e.g., $X = D * Z$ ); and  |
| B | is a constant.   |

The parameter Y would also be stochastic, since it is a function of one or more stochastic parameters. Using the parameters module, the above equations, the probability distributions of the stochastic parameters, and the values of the constants can be directly entered by the user. The user can then compute and display the probability distribution of Y, along with its subsidiary parameters A and X.

The parameters module allows the user to develop a data base of parameter definitions. In addition, having defined a parameter, and prior to actually running the repository system simulation using RIP, the user can evaluate and display the value (for a deterministic parameter) or the probability distribution (for a stochastic parameter). Entry and editing of data is straightforward, with the user simply entering the symbolic name for each parameter, and typing in the equation (for a function), the type and form of the probability distribution (for a stochastic parameter), or the value (for a constant).

The nature of the parameter types, the facilities to enter and edit parameter data, and the means of generating graphical representations of stochastic parameters are presented in the following sections.

##### 4.1 Entering the Parameters Module

The parameters module can be entered in two ways:

- It is entered automatically by typing a parameter name in a parameter data field in any input window and hitting <ENTER>. If the parameter name

specified exists in the parameter module data base, the parameters module editing window is displayed. If the parameter name does not exist, RIP will ask whether the user wishes to add it to the database. If the answer is Yes, the parameters editing window is then displayed. If the answer is No, the parameter is not added to the parameters database (the user will need, however, to add the parameter to the database prior to running a simulation).

- The parameters module can also be entered directly from anywhere within the user interface by hitting <Alt>-<P>.

## 4.2 Browsing and Editing the Parameter Database

The parameter module editing window is illustrated below:

[ ] Repository Integration Program (Rev. 1.00 )			
General	PA Model	Strategy Model	Results
[ ] Parameter Editing			
Identifier : AGE1		Parameter No. : 1 of 73	
Description: aging rate for general corrosion			
Parameter Type : Function			
Edit Parameter Definition			
F1=Prev Param	F2=Next Param	F3=Add Param	F4=Delete Param
F5=Find Param	F6=Print Summary	F7=Display Dist.	F8=Expected Value
F9=Bayesian update (Current update level=1	)		F10=Pop Update

The parameter data base is ordered alphabetically by name. If the parameter editing window is entered using <Alt>-<P>, it will display the first parameter in sequence. If the editing window is entered indirectly by typing a parameter name within a parameter field and hitting <Enter>, it will display the specified parameter. The user can browse through the data base and perform editing operations using the function keys:

F1	Move to previous parameter in alphabetic order
F2	Move to next parameter in alphabetic order
F3	Insert a new parameter
F4	Delete the current parameter
F5	Find a specific parameter by name
F6	Print a summary of the database
F7	Display the distribution of a stochastic parameter or function
F8	Display the expected value of a stochastic parameter
F9	Carry out a simulated Bayesian update
F10	Move downward ("pop") through stack of Bayesian updates

Note that if the parameters module is entered indirectly (by specifying a parameter name and hitting <ENTER>), only the <F7> and <F8> are active. This is because when the parameters module is entered in this manner, only the parameter which has been specified can be edited. The parameters module must be entered directly using <Alt>-<P> to have

full access to the entire parameters database.

Functions keys <F1> and <F2> allow you to move backwards and forwards through the data base in alphabetical order.

The <F3> key allows a new parameter record to be created. The user is prompted for the name of the new parameter to be created. The user can fill in values for the new parameter as described below in Section 4.3.

The <F4> key is used to delete parameters from the data base. When you press the <F4> key, a window will pop up with the query

"Confirm deletion of parameter NAME: (Y):"

Press "Y" or <Enter> to confirm the deletion. Press "N" if you want to change your mind.

The <F5> key is used to find a parameter within the data base. When you press the <F5> key, a window will pop up to prompt you for the ID of the parameter you wish to find and edit. Type all or part of the ID, press <Enter>, and the data for the selected parameter will be displayed ready for editing. An error message will appear if no matching parameter can be found in the data base.

The <F6> key is used to print a summary of the parameters data base. When you press the <F6> key, a window will pop up with the query

"Do you want detailed parameter output? (N):"

Press "Y" or <Enter> for a detailed listing of all parameters. Press "N" for a short listing (one line for each parameter). The detailed listing goes to a file named PARAM.DET. The short listing goes to a file named PARAM.GEN. You can send these files to your printer after leaving the parameters program.

The <F7> key is used to display the distribution of a stochastic parameter or function. This is described in more detail in section 4.4 below.

The <F8> key displays the expected value of the parameter.

The <F9> and <F10> keys are only active for a certain type of parameter (a stochastic parameter), and are discussed in more detail in Section 4.3.3.

### 4.3 Editing Parameters

The arrow keys (or the <TAB> and <SHIFT>-<TAB> keys) can be used to move between fields which can be edited in the parameter window. Within a field, the left and right arrow keys, and the <Home> and the <End> keys can be used for movement. The <Ins>,

==	equal to
<>	not equal to
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
!!	or
&&	and
!	not

Expressions using relational operators are evaluated as to whether they are true or false. If the expression is true, it takes on the value of 1, if it is false, it takes on the value of 0. An expression which consists of a value (without operators) is true (=1) for all non-zero values. Hence 5&&3 is equal to 1 (it is a true statement). 5&&0 is equal to 0 (it is false since 0 is equivalent to a false statement). Note that !0 is equal to 1 (a true statement); !(any non-zero value) is equal to 0 (a false statement).

The precedence of the operators is as follows (from highest to lowest):

parentheses	()
unary	-, !
multiplicative	*, /
additive	+, -
relational	>, <, >=, <=
equality	==, <>
logical AND	&&
logical OR	!!

The following built in trigonometric functions expect an argument in radians:

SIN(A)  
COS(A)  
TAN(A)  
COT(A)

The following inverse trigonometric functions return a value in radians:

ASIN(X)  
ACOS(X)  
ATAN(X)

These built in functions are also defined:

ERRF(X)	Gauss error function
LOG(X)	logarithm base 10
LN(X)	natural logarithm
MIN(X,Y)	returns the min of X and Y

MAX(X,Y)	returns the max of X and Y
BESS(V,X)	Bessel function of X of order V
BETA(X,Y)	Beta function

In addition, the user can use IF, THEN logic and relational operators within an equation. The form of the function is as follows:

IF(X,Y,Z)

where X, Y, Z are parameters, expressions, or functions. The IF function is interpreted as follows: if X is not equal to 0 (i.e., if X is a true expression), then the function is equal to Y, otherwise it is equal to Z. Specifying the first argument as an expression using a relational operators, and nesting the function permits a large degree of flexibility. For example,

IF( X==1,Y,(IF(X==2,Z,ZZ) ) ) says the following:

if X = 1, the function = Y;  
 else, if X = 2, the function = Z;  
 else, the function = ZZ

There is no limit to the degree of function nesting.

When you have completed editing the function definition, press the <Esc> key to close this window and return to the main parameter editing window.

#### 4.3.3 Defining a Stochastic Parameter

When the user selects [Edit Parameter Definition] for a stochastic parameter, a "Probability Distribution Definition" window will appear. An example of this window appears below:

[ ] Repository Integration Program (Rev. 1.00 )			
General	PA Model	Strategy Model	Results
[ ] Parameter Editing			
Identifier : DISS		Parameter No. : 8 of 73	
Description: matrix dissolution rate for spent fuel			
Parameter Type : Stochastic			
Edit Parameter Definition			
F1=Prev Pa	[ ] UNIFORM DISTRIBUTION		F1-Change type
F5=Find Pa	Probability Distribution Definition		F2-Log/Linear
F9=Bayesia	Sampling Bias : No Bias		log
Lower bound: 0.1		(Constant )	
Upper bound: 10		(Constant )	

Function keys can be used to modify the distribution or to display more information.

F1 Change distribution type  
 F2 Switch between linear and log distribution

Pressing the <F1> key will switch between the following distributions:

Normal  
Uniform  
Triangular  
Poisson  
Beta  
Gamma  
Binomial  
Determinate  
Weibull  
Cumulative  
Discrete

Appendix A discusses the form and required inputs for each of these distributions in detail.

Pressing the <F2> key will toggle between linear and logarithmic distributions.

As is described in Appendix A, the fields that appear in the "Probability Distribution Definition" window will change with the distribution type. For example, the window for a normal distribution includes fields for the mean and standard deviation of the distribution, while a triangular distribution is defined by fields for lower bound, upper bound and most likely value. In any of these fields, the user can enter either a numeric constant or the name of another parameter. If one of the fields that defines a probability distribution contains the name of another parameter, that parameter must be edited separately.

Every stochastic parameter input window has a "*Sampling Bias*" input field. This refers to the amount of importance sampling (or biasing) which is to be used when sampling the distribution. As discussed in Section 3.3.1, this can be set globally for all stochastic parameters from the [General] input window. Otherwise, it can be set for each individual parameter here. The user has four choices to toggle through: No Bias, Little Bias, Moderate Bias, and Lots of Bias. No Bias is the default. No Bias results in straight Monte Carlo sampling with no importance sampling. The other three choices represent increasing levels of importance sampling. Biasing is discussed in detail in Appendix C of the RIP Theory and Capabilities Manual. Using the importance sampling terminology defined in that appendix, the biasing is specified in RIP as follows:

No Bias	n=1.00
Little Bias	n=1.33
Moderate Bias	n=1.67
Lots of Bias	n=2.00

When editing of the stochastic parameter is completed, press the <Esc> key to close this window and return to the main parameter editing window.

The <F9> and <F10> keys in the main parameter window allow the user to create (and view) a Bayesian update of a stochastic parameter (these keys are only active for stochastic



parameters). This feature is added simply to illustrate the Bayesian updating scheme incorporated into the strategy model. Bayesian updating is discussed in detail in Chapter 6 of the RIP Theory and Capabilities Manual and Chapter 6 of this document.

The following pop-up window is displayed upon hitting the <F9> key:

Repository Integration Program (Rev. 1.00)			
General	PA Model	Strategy Model	Results
[ ] Parameter Editing			
Identifier : DISS		Parameter No. : 8 of 73	
Description: matrix dissolution rate for spent fuel			
Parameter Type : Stochastic			
Edit Parameter [ ]			
BAYESIAN UPDATE			
Mult. factor on original S.D. : 1			
Bounds of Distribution			
Min : 0.10046207 Max : 9.95400524			
True value to update about : 2.39327377			
Execute Update			
F1=Prev Param		e Param	
F5=Find Param		ted Value	
F9=Bayesian up		Update	

The "Multiplication factor on original S.D" is the precision of the "new information" relative to that represented by the original distribution (the "old information" or prior knowledge). Specifying 0.1 implies that the new information is ten times more precise than the old information (i.e., the standard deviation of the new information is a tenth that of the old); specifying 2 implies that the new information is only half as precise as the old information. The user must also specify the "true value to update about". This must be between the minimum and the maximum specified in the fields immediately above. "Execute Update" updates the parameter. The updated distribution can then be viewed (using the <F7> key). Note that the updated distribution is *always in the form of a cumulative distribution*, regardless of the form of the original distribution.

After updating a parameter the display field named "current update level" on the parameter editing screen will change (from 1 to 2). If another update is carried out, this value becomes 3. This simply identifies the distribution which is "at the top of the stack" and which can currently be accessed and viewed. Additional updates update the previously updated distribution. Hence update level 3 is an update of update level 2, which is an update of update level 1 (the original distribution). The user can "pop" down the stack (all the way back to update level 1) by hitting <F10>.

Note that this feature is intended only to illustrate the updating algorithm used in the strategy model. It does not affect the shape of the distribution used in performance simulations. The original distribution (update level 1) is always used for performance simulations (i.e., RIP automatically "pops" all the way back to level 1 prior to the simulation).

#### 4.3.4 Defining an Event Parameter

When the user selects [Edit Parameter Definition] for an event parameter, a window labeled "Edit Event" will appear:

[ ] Repository Integration Program (Rev. 1.00 )			
General		PA Model	Strategy Model
[ ]		Parameter Editing	
Identifier : EVENT1		Parameter No. : 10 of 73	
Description:			
Parameter Type : Event			
Edit Parameter Definition			
F1=Prev	[ ] EDIT EVENT		m
F5=Find	Annual Rate of Occurrence : 0.0001	(Constant )	lue
F9=Baye	Can Event Reoccur : Yes		

Enter a numeric constant or the name of a parameter in the "Annual Rate of Occurrence" field. This is the Poisson rate. Enter "Yes" or "No" (using the <PgUp> and <PgDn> keys) in the field labeled "Can Event Reoccur". RIP assumes events behave as Poisson processes. If the event cannot reoccur, RIP will recognize only the first occurrence of the event.

When used in a function, the realized value of an event parameter is a function of time. It is equal to the number of event occurrences for the current timestep (e.g., for a given timestep, it is equal to zero if the event has not occurred that timestep).

When you have finished editing the event parameter, press the <Esc> key to close the window and return to the parameter editing window.

#### 4.3.5 Reserved Parameter Names

It is important to reiterate that in all of the parameter types described above (with the exception of Constant), the user can define the parameters such that they are dependent on other user-defined parameters.

Function type parameters can be described directly as functions of other parameters. In the examples shown below, the parameters are defined as functions of three other user-defined parameters (A, B, and C):

$$\text{IF}(A \geq 10, B, C)$$

$$(A + B) * \text{LOG}(C)$$

Stochastic type parameters can be defined such that the input fields describing the distribution (e.g., the mean and standard deviation for a normal distribution) can themselves be specified as parameters (i.e., constants, stochastics, or functions). Likewise, event type parameters can be defined such that the rate of occurrence can be specified as a parameter.

In addition to defining parameters such that they are dependent on other user-defined parameters, however, the user can also define parameters which are described in terms of several *internally-defined, reserved parameter names*. In particular, the user can refer to the following parameter names:

TIME	the simulation time
TEMP	the time-varying temperature at the edge of a waste package
CONTAC	the water contact mode at a particular waste package
" — "	the names of any user-defined environmental factors defined under [Near Field Conditions]

Use of TIME is straightforward. If the user wishes to describe a process with a parameter which changes with time (TIME=0 being the start of the simulation, which is the time of closure), that parameter can simply be described as a function of time (in years). The other names listed above refer to *environmental factors*, which describe *near field conditions* in the immediate vicinity of a waste package (e.g., temperature, moisture, or chemical conditions). These are described in detail in Section 5.1.1.

Not all input variables are allowed to be specified as functions of time. In some cases, this is due to computational considerations associated with the algorithms implemented within RIP; in others it is simply not physically reasonable to describe the input in such a manner. Inputs which are constrained in this manner are clearly noted in the following chapter. As will be shown in Section 5.1.1, TEMP is defined by the user as a function of time. Therefore, any inputs which are constrained to being temporally invariant can not be described as a function of TIME or TEMP. The user interface automatically checks for these constraints prior to creation of the transfer file.

RIP will not allow the user to define parameters having the names listed above. These names are *reserved*. Likewise, operator names included in the user-interface equation parser (e.g., IF, COS, LOG) are also reserved.

#### 4.4 Displaying Stochastic Parameters and Functions

When defining stochastic parameters, particularly complex ones, it is useful to view the form of the distribution when in the process of defining it. RIP gives the user this capability.

There are two ways to display the distribution of a stochastic parameter: 1) by sampling the specified distribution; and 2) by showing the analytical function which describes the distribution. The first approach samples the distribution a specified number of times and uses this information to graph the distribution. The second approach simply plots the specified analytical solution rather than sampling the distribution. The first distribution approaches the second as the number of samples grows large. Stochastic parameters can be displayed in either manner. Stochastic functions can only be displayed by sampling.

To display the distribution of a stochastic parameter or a stochastic function, press <F7> while in the "Parameter Editing" window. For stochastic parameters (as opposed to stochastic functions), RIP will first prompt the user to indicate whether he/she wants to generate Monte Carlo results for display. The user should answer No to display the analytical function directly. If the user answers Yes (generate Monte Carlo results), RIP will prompt for the *number of realizations* to sample for plotting the distribution.

For stochastic functions, it is not possible to display the exact solution, and Monte Carlo sampling is automatically implied. In this case, the first question discussed above is not applicable and is not displayed. In either case (exact solution or Monte Carlo sampling), the following input screen will then appear:

Repository Integration Program (Rev. 1.00)				
General	PA Model	Strategy Model	Results	
[ ]- F4=Targets/Plot				
PARAMETER DISPLAY WINDOW				
Plot type	: CCDF	Plot Range	Data Range	
X-axis	: Linear 0.1	0.35	0.1	0.35
Y-Axis	: Linear 0	1	0	1
Grid	: No			
Title	: Result for stochastic parameter CON1			
X-axis Title	: Parameter CON1:			
Percentiles of the Parameter's Distribution				
0.001	: 0.1002	Statistics: Mean : 0.2		
0.01	: 0.102	S.D. : 0.057735027		
0.05	: 0.11	Skewness : 0		
0.10	: 0.12	Kurtosis : 1.8		
0.25	: 0.15			
0.50	: 0.2			
0.75	: 0.25			
0.90	: 0.28			
0.95	: 0.29			
0.99	: 0.298			
0.999	: 0.2998			
		Targets		
		Cum. Prob.: 0.5	Value : 0.2	
		Value : 0.2	Cum. Prob.: 0.5	
		Prob. Dens.: 5.		
F1=Screen Plot F2=HardCopy Plot F3=Print Results-Array				

This window has two parts: the top half allows the user to specify the characteristics of the graphical display. The bottom half displays the percentiles of the parameter's distribution and the distribution's statistics. The function key <F4> toggles between the two halves.

The bottom half of the screen displays the cumulative distribution for the parameter of interest. Eleven probability values, ranging from 0.001 to 0.999 are displayed. While in the lower screen, the user can request (under "Targets") the parameter value corresponding to a specific probability or the probability corresponding to a specific parameter value.

The top half of the screen allows the user to modify the axes and specify the form of the graphical output.

The arrow (or <TAB> and <SHIFT>-<TAB>) keys can be used to move between fields. The following fields may be edited:

Plot type	toggle between CDF (cumulative distribution function), PDF (probability density function) and CCDF (complementary cumulative distribution function) using <PgDn> or <PgUp>
X -axis plot range	define X limits of plot
Y axis plot range	define Y limits of plot
X Axis and Y axis	toggle to select linear or log scale using <PgDn> or <PgUp>
Grid	toggle to turn grid on or off using <PgDn> or <PgUp>

Title	enter title for plot
X-axis Title	enter label for X axis
Y-axis Title	enter label for Y axis

Within this window, the following function keys are active:

F1	Screen Plot
F2	Hardcopy Plot
F3	Print Results Array

Press <F1> for a screen display of the CDF, PDF or CCDF in the format that you have defined. As discussed in Section 2.4, several methods can be used to capture or print this screen directly.

<F2> produces a hardcopy plot file of the distribution. The user is prompted for an output format and for a file name. The current version of RIP only accommodates HPGL and HPGL2 output formats. After leaving the parameters program, the user can send the file created here to a printer or plot device or load it into a suitable software program.

<F3> produces an ASCII file listing of all of the realized values. The user is prompted for a filename. An example listing (based on just 20 realizations) is shown below.

```
The mean of the results is    10.103
The standard deviation is    1.136
The skewness is             0.064
The kurtosis is             2.446
```

```
There were    20 realizations.
There are    20 in the results list.
The total weight of the results is    20.000
```

The list of results is as follows :

1 Result value =	7.873547	Weight =	0.050000	Cum =	0.025000
2 Result value =	8.617570	Weight =	0.050000	Cum =	0.075000
3 Result value =	8.813642	Weight =	0.050000	Cum =	0.125000
4 Result value =	8.840748	Weight =	0.050000	Cum =	0.175000
5 Result value =	8.891684	Weight =	0.050000	Cum =	0.225000
6 Result value =	9.462067	Weight =	0.050000	Cum =	0.275000
7 Result value =	9.504783	Weight =	0.050000	Cum =	0.325000
8 Result value =	9.711614	Weight =	0.050000	Cum =	0.375000
9 Result value =	10.082269	Weight =	0.050000	Cum =	0.425000
10 Result value =	10.121875	Weight =	0.050000	Cum =	0.475000
11 Result value =	10.283372	Weight =	0.050000	Cum =	0.525000
12 Result value =	10.302446	Weight =	0.050000	Cum =	0.575000
13 Result value =	10.420690	Weight =	0.050000	Cum =	0.625000
14 Result value =	10.567050	Weight =	0.050000	Cum =	0.675000
15 Result value =	10.571251	Weight =	0.050000	Cum =	0.725000
16 Result value =	11.063910	Weight =	0.050000	Cum =	0.775000
17 Result value =	11.218164	Weight =	0.050000	Cum =	0.825000
18 Result value =	11.484658	Weight =	0.050000	Cum =	0.875000
19 Result value =	11.781811	Weight =	0.050000	Cum =	0.925000
20 Result value =	12.442783	Weight =	0.050000	Cum =	0.975000

Press <Esc> to erase the plot and return to the original window.

#### 4.5 Application of the Parameters Module

After experimenting with the parameters module, the user will come to realize that it can be quite useful and powerful as a stand alone tool (i.e., external to other portions of the RIP software) for examining the behavior of stochastic functions. That is, the user can build relatively complex parameters which are described as complex functions of other parameters (some of which may be stochastic), and propagate the uncertainty through a specified mathematical model. Hence, the user can define a stochastic model of a system and then examine its behavior.

For example, parameter A may be defined as a complex function of the stochastic parameters B, C, D, E and F (incorporating IF, THEN logic and complex mathematical functionalities). The parameters module allows the user (by selecting <F7>) to propagate the uncertainty represented in the probability distributions defining B, C, D and F through the mathematical model specified by the user in order to determine the probability distribution of A.

This ability can be very useful to the user for examining intermediate results not explicitly output by the main RIP program.

## 5. DATA ENTRY AND APPLICATION OF THE PERFORMANCE ASSESSMENT MODEL

In this chapter, we describe the details of data entry for the performance assessment model. As discussed in Section 3.2.1, the data input windows for the performance assessment model are accessed by selecting [PA Model] from the top level menu. Input for the PA model is subdivided into four categories: [Waste Package], [Pathways], [Events], and [Make Transfer File].

Before attempting to input data into RIP, it is necessary for the user to first formalize the conceptual model for the system. This entails describing the characteristics and expected behavior of the waste package, the near field environmental conditions expected at the repository, and the transport pathways through which radionuclides can migrate in the subsurface environment. Building and documenting such a conceptual model requires both knowledge pertaining to the particular system to be simulated, and a clear understanding of the capabilities and limitations of the RIP software. Before entering data into RIP, the user should organize and sketch out the conceptual model of the site. Figure 4-1 in the RIP Theory and Capabilities document shows an example of a simplified conceptual model of the Yucca Mountain site.

Having developed an appropriate conceptual model of the system to be simulated, the data representing that conceptual model can be entered into RIP. Before describing the details of the input windows, however, it is necessary to describe the order in which data is entered by the performance assessment model. When editing an existing data input set, the user can move through the menu and input windows in any manner desired. When creating an input data set from scratch, data is generally entered by simply stepping sequentially through the menus and input windows. However, because of the interdependency of several of the input windows, there are several exceptions to this rule. The necessary steps for entering data into the performance model when starting from scratch are as follows:

- 1) Specify input under [Waste Package] as follows:
  - a) Under [Near Field Conditions], specify input for [Water Contact Modes] and [Other Factors]. Do not specify input for [Temperature Conditions] at this time.
  - b) Under [Waste Package Description], specify all input except that which describes the waste package inventories (accessed by the Inventory access field within the main waste package input window).
  - c) Return to [Near Field Conditions], and specify input for [Temperature Conditions].
- 2) Specify all input under [Pathways].
- 3) Return to [Waste Package]; under [Waste Package Description] specify input which

describes waste package inventories (accessed by the **Inventory** access field within the main waste package input window).

4) Specify all input under **[Events]**.

5) If running the performance assessment model alone (without the strategy shell), specify all data under **[Make Transfer File]**, and create a transfer file for the back end.

Note that while entering data, the parameters module can be repeatedly accessed as described in the previous section in order to define any parameters specified in the input windows. The **[General]** input window can also be accessed in order to print an input screen, save an echo file, globally set the level of importance sampling, or ~~turn on or off~~ Latin-Hypercube sampling, as discussed in Section 3.4. ~~turn on or off~~

The various input windows under **[PA Model]** are described sequentially below.

## 5.1 Input for the Waste Package Behavior and Radionuclide Release Component Model

Selecting **[Waste Package]** from the menu under **[PA Model]** provides access to input windows for describing the waste package behavior and radionuclide release component model. Input for this component model is divided into three categories: **[Near-Field Conditions]**, **[Waste Package Description]**, and **[Make WP Transfer File]**. Each of these categories is discussed in detail below.

### 5.1.1 Describing Near-Field Conditions

Selecting **[Near-Field Conditions]** from the menu under **[Waste Package]** provides access to input windows for describing near-field conditions in the vicinity of the waste packages. Other waste package input parameters can subsequently be described in terms of these conditions. There are three categories of near-field conditions, each of which is represented by a menu item: **[Water Contact Mode]**, **[Temperature Conditions]**, and **[Other Factors]**. A fourth menu item, **[Show Groups]**, is used to summarize the near-field conditions. Because the concepts involved in defining near-field conditions are somewhat complex and not self-explanatory, the discussion presented in this section is longer than those describing other aspects of the model.

Before discussing the details of specifying near-field conditions, it is necessary to understand how RIP enables the user to represent *variability* between waste packages. *Variability* should not be confused with *uncertainty*. Uncertainty in a model parameter implies a lack of knowledge regarding the actual value of the parameter. This is represented in RIP by describing the uncertain parameter as being stochastic (i.e., represented by a distribution). The parameter's distribution is sampled each realization. That single parameter value is then used throughout the realization. Variability in a model parameter, on the other hand, implies that for a given realization, a distribution of parameter values exist. Note that the variability



that is represented here is a *random variability*. There are no spatial trends implied. The variability in a particular parameter occurs randomly throughout the repository.

Waste package-to-waste package variability is incorporated in the model by internally discretizing the waste package population into a discrete number of *waste package groups*. Groups are defined as a function of both near-field conditions and waste package type (e.g., spent fuel, defense waste). That is, a particular waste package group includes waste packages of a particular type which are subject to a specific set of near-field conditions. As will be discussed in detail in Section 5.1.2, the user specifies the number of waste packages of each type which are emplaced in the repository. We describe here how waste package groups represent variable near-field conditions.

Near-field conditions are defined in terms of specified variable *environmental factors* (water contact mode, temperature, pH, redox conditions, stress, etc.). The user must determine which environmental factors he/she wishes to include in the model. Two environmental factors, water contact mode (i.e., moisture conditions) and temperature, are explicitly incorporated in the computational algorithm. As will be shown, the structure of RIP is such that other user-defined environmental factors can readily be added.

For each environmental factor which the user wishes to represent as being variable, it is necessary to define a distribution describing its variability. Based on additional user input (to be described below), RIP then converts the distribution describing the parameter's variability into an *equivalent discrete distribution*. (As will be discussed below, water contact mode is an exception to this rule: its variability is directly specified in a discrete manner.)

As a simple example, consider a case in which we assume that the environmental conditions can be defined in terms of only two controlling variable environmental factors: the mode of water contact and the temperature. To further simplify the example, assume that the variability distributions of these controlling parameters are already discretized. Since these parameters are described in terms of discrete distributions, they will automatically define a discrete number of *waste package groups*. For example, if we define water contact mode in terms of three discrete categories (e.g., "wet-drip", "moist-continuous" and "dry"), and temperature conditions in terms of three discrete categories (e.g., "hot", "warm", "cool"), the waste package population would be divided into  $3 \times 3 = 9$  waste package groups. If more than one waste package type is included in a simulation (e.g., spent fuel and defense waste), the population would also be distinguished by waste package type, resulting effectively in  $3 \times 3 \times 2 = 18$  groups. This is illustrated schematically in Figure 5-1.

A certain fraction of the entire waste package population will be located in each of these groups. For example, if 30% of the waste packages are "wet-drip", 20% of the waste packages are "warm", and 80% of the waste packages are of the type "spent fuel", then the "wet-drip", "warm", "spent fuel" waste package group would contain  $(0.3 \times 0.2 \times 0.8 = 0.048)$  4.8% of the waste packages. (When computing the number of waste packages in each group, RIP rounds numbers to the nearest integer. Any deviation in the total number of waste packages is then corrected by adding or subtracting an appropriate number of waste packages to the largest group.)

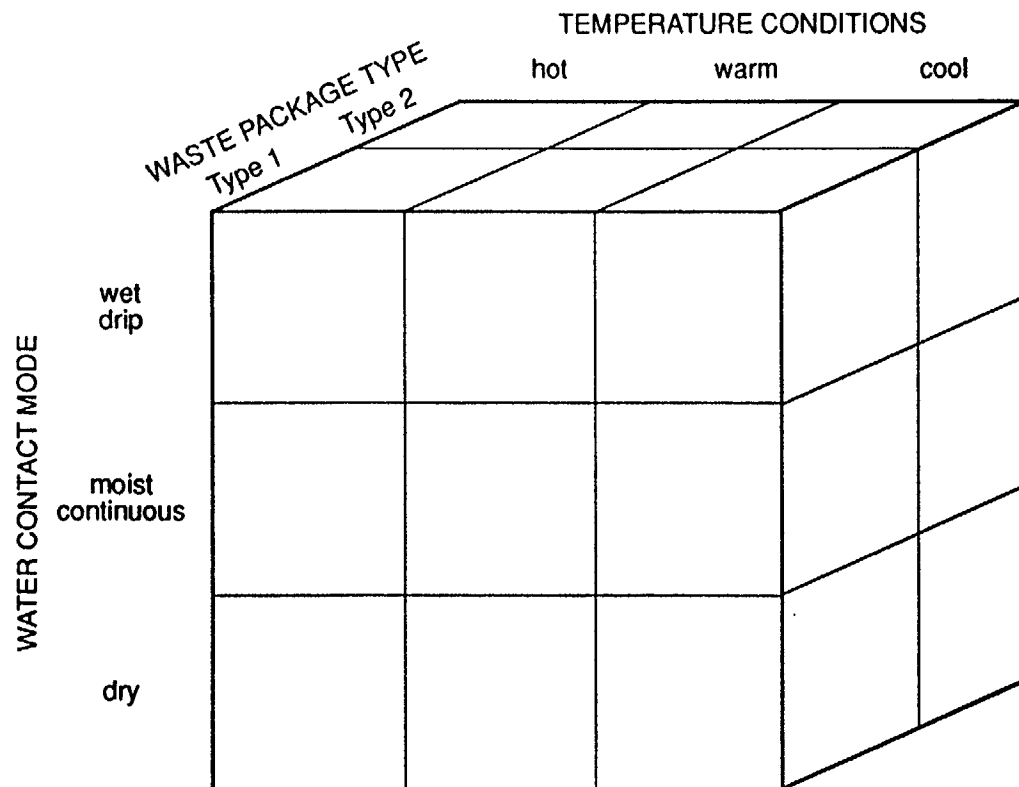


FIGURE 5-1  
 EXAMPLE OF DISCRETIZATION OF REPOSITORY  
 INTO WASTE PACKAGE GROUPS  
 ARGONNE/MODEL DEVELOPMENT

Each environmental factor has a name by which it can be referenced (e.g., CONTAC, TEMP, PH). System parameters can be made dependent on environmental factors by referencing these names in a function. The value of these parameters would therefore vary from group to group. Within the computational algorithm, at every timestep within a realization RIP computes the release from each group, and then sums the results over all the groups to compute the total release from the repository.

Note that this representation of variability requires that the different environmental factors defined by the user be independent of each other. (e.g., the distribution of water contact modes is independent of the distribution of temperature). This and other aspects of RIP's representation of variability are discussed further in the RIP Theory and Capabilities Manual.

The actual manner in which environmental factors are specified is presented below.

#### 5.1.1.1 Water Contact Mode

Selecting [Water Contact Mode] from the menu under [Near-Field Conditions] provides access to the following input window:

[ ] Repository Integration Program (Rev. 1.00 )			
General	PA Model	Strategy Model	Results
[ ] User's Guide Tutorial			
Waste Package	Pathways	Events	Make Transfer File
[ ]			
Near Field Conditions	Waste Package Description	Make WP File	
[ ]			
Water Contact Mode	Temperature Conditions	Other Factors	Show Groups
[ ] Water Contact Mode			
Environmental factor ID : CONTAC	Value : 1	Mode 1 of 2	F1=Prev
Water Contact Mode Description :			F2=Next
moist-continuous water contact			F3=Add
Fraction of Waste Packages Remaining : 1			F4=Del
Fraction remaining WPs in this mode : 0.4	(Constant )		

Due to hydrogeologic variability, waste packages in an unsaturated environment will be subject to different moisture conditions throughout the repository. This variability is represented in RIP by specifying a discrete distribution of *water contact modes* (e.g., wet-drip, moist-continuous) in the immediate vicinity of a waste package. The water contact mode is referenced by the user by assigning an integer value to the internal environmental factor CONTAC, each integer value indicating (i.e., acting as an identifier for) a different water contact mode. Other processes and parameters can then be described as a function of water contact mode (e.g., mass transfer parameters and matrix alteration/dissolution processes). For example, if the user defined water contact modes A and B, with the integer identifiers 1 and 2, respectively, and then wanted a particular system parameter W to vary depending on the water contact mode, W could be defined as a function type parameter (see Section 4.3.2) using IF,THEN logic:

IF(CONTAC==1,50,100)

In this case, the parameter W would take on the value of 50 for contact mode A (CONTAC = 1), and the value of 100 for contact mode B.

For each value of CONTAC (i.e., for each water contact mode), the user must specify the fraction of waste packages subjected to those conditions. Hence, water contact mode is an exception to the rule described above which stated that the variability of the environmental factor is specified as a continuous distribution which is then converted to an equivalent discrete distribution. CONTAC is directly specified in a discrete manner.

In the example illustrated in Figure 5-1, "wet drip", "moist continuous" and "dry" refer to different values of CONTAC (1, 2 and 3).

Note that water contact mode variability is assumed to be due to local random hydrogeologic conditions at the scale of the individual waste packages - it is not meant to represent large-scale spatial trends. In addition, note that the water contact mode represents the condition *after* the thermal period. As will be described below, waste packages may be defined such that they remain dry for some period of time due to a thermal pulse resulting from radioactive decay. The water contact modes, however, represent the expected conditions during the *post-thermal* period.

The function keys <F1> through <F4> can be used to add, delete or browse through the modes.

The specific inputs for this window are as follows:

**Value:** This is the integer identifier for a particular water contact mode. It can be specified by the user as any two digit integer value. Its default value is the mode number specified in the upper right hand corner of the input screen (i.e., for "Mode 2 of 4", Value is originally 2).

**Water Contact Mode Description:** This is simply a short description of the water contact mode, limited to 60 characters.

**Fraction of remaining WPs in this mode:** This is the *fraction of the remaining balance* of waste packages which are in this water contact mode. The "Fraction of Waste Packages Remaining" is specified in line immediately above this field. For example, suppose the user wished to define several contact modes. If the first was specified as having 40% of the waste packages, the window for the second water contact mode would look like this:

Repository Integration Program (Rev. 1.00)				
General	PA Model	Strategy Model	Results	
User's Guide Tutorial				
Waste Package	Pathways	Events	Make Transfer File	
Near Field Conditions	Waste Package Description		Make WP File	
Water Contact Mode	Temperature Conditions	Other Factors	Show Groups	
Water Contact Mode				
Environmental factor ID : CONTAC		Value : 2	Mode 2 of 2	F1=Prev F2=Next F3=Add F4=Del
Water Contact Mode Description :				
Fraction of Waste Packages Remaining : 0.6				
Fraction remaining WPs in this mode : 1		(Constant )		

RIP would indicate that 60% of the waste packages remain to be defined. Specifying 0.5 in the "fraction remaining" field would mean that 50% of the remaining packages (or 30% of the total) are allocated to water contact mode 2. The remaining 30% of the waste packages would have to be allocated to additional modes. Specifying 1 in this field would mean that all of the remaining waste packages (60% of the total) are in water contact mode 2.

For the last water contact mode defined, the value automatically takes on the value of 1. For other modes, however, this field can be specified as a parameter (a constant, a stochastic, or a function), with some constraints: It must *always* be greater than or equal to 0 and less than or equal to 1. Furthermore, if described as a function, the parameter *can not be defined as a function of time*.

Entering the *fraction of remaining balance* (as opposed to the *actual fraction*) may appear somewhat awkward. However, this convention allows the fractions to be represented stochastically, while still ensuring that they total to one. This convention is used in several other places within the RIP user interface and its proper application is discussed further in Appendix B.

Note that variable water contact modes will typically only be applicable for unsaturated repositories. When simulating saturated repositories, a single mode of water contact is likely to be appropriate. A single water contact mode (implying no variability in water contact mode) is the default for RIP.

Note that because the fraction of waste packages in each water contact mode can not vary with time, they can not directly respond to temporal changes in environmental conditions such as repository groundwater flux. Nevertheless, if this is deemed an important effect to consider, it can be accommodated in an approximate manner. As an illustrative example, suppose the user defines two water contact modes: mode A and mode B. Further suppose that under current repository flux conditions, the waste packages are split evenly between these two modes (50% in each). As the flux increases, however, more waste packages shift to mode B. This can be accommodated in an approximate manner by defining one or more *transition* modes. For illustrative purposes, we will denote mode AB as a transition mode. We might then say that modes A, AB, and B have 20%, 30%, and 50% of the waste packages, respectively. We can then define mode AB such that it is identical to mode A under low flux

conditions (say  $FLUX < 10$ ), and identical to mode B under high flux conditions ( $FLUX > 10$ ). For example, if parameter  $W$  depends on the water contact mode (taking on the value of 50 in mode A and 100 in mode B), the user might specify  $W$  as follows:

$$W = IF(CONTAC==1,50, W1)$$

where  $W1$  and  $W2$  are parameters defined as follows:

$$W1 = IF(CONTAC==2,100, W2)$$

$$W2 = IF(CONTAC==3!!FLUX <= 10,50,100)$$

In this example, contact modes A, B and AB have been assigned the integer identifiers 1, 2, and 3, respectively.  $W$  would take on the value 50 if in contact mode A, 100 if in contact mode B, 50 if in contact mode AB and  $FLUX$  is less than or equal to 10, and 100 if in contact mode AB and  $FLUX$  is greater than 10. The transition could be made more continuous by defining additional transition modes (i.e., modes that switch behavior at intermediate flux values).

#### 5.1.1.2 Temperature Conditions

Selecting [Temperature Conditions] from the menu under [Near Field Conditions] provides access to input windows for describing temperature conditions.

The thermal behavior of the repository is represented in RIP by a built-in system parameter describing the temperature history at the edge of a waste package. This parameter is defined as a function of *waste package type*. Therefore, as discussed previously, temperature conditions can not be defined prior to defining the waste package types (which will be described below in Section 5.1.2). The temperature history is computed internally by RIP as follows:

$$TEMPAV_i(t) = T_m(t) + T_i(t)$$

where:

$TEMPAV_i(t)$  is the average temperature at the edge of waste package type  $i$  at time  $t$ ;

$T_m(t)$  is the mean temperature in the rock throughout the entire repository at time  $t$ ;  
and

$T_i(t)$  is the incremental temperature above the repository mean temperature at the edge of waste package type  $i$  at time  $t$ .

The user defines time histories for  $T_m$  and  $T_i$  as a series of discrete points, and specifies the uncertainties in these time histories.

Because some of the controlling factors are likely to be variable, the actual temperature history at waste packages may vary significantly throughout the repository. This is represented in RIP by quantifying the *variability* about the average temperature  $TEMPAV_i(t)$  and creating a

*discrete distribution* of temperature variability. The procedure, although simple, is not self-explanatory, and is detailed below.

Specification of temperature variability is carried out in two steps. First the user specifies the form of a variability distribution which is used to modify the average temperature  $TEMPAV$ . We refer to the variability parameter as  $TEMPV$ . In the current version of RIP, the distribution of this parameter can be either triangular or uniform, and must be symmetric about the value 1 (e.g., a uniform distribution between 0.85 and 1.15). Having specified the distribution, the user then specifies how this distribution is to be discretized by defining several (up to six) cumulative probability levels (e.g., 0.33, 0.66, 1.0). RIP then discretizes the distribution by using the expected value of the ranges defined by the specified probability levels as the discrete values. For example, specifying a uniform distribution with a variability of 15% (0.85 to 1.15), and discretization at the 0.33, 0.66, and 1.0 cumulative levels, defines the following discrete distribution:

<u>TEMPV</u>	<u>PROBABILITY</u>
0.9	0.33
1.0	0.33
1.1	0.34

The distribution has been discretized into three discrete values corresponding to the three ranges defined by the specified probability levels: 0 to 0.33, 0.33 to 0.66, and 0.66 to 1.0. The discrete values for the distribution are computed as the expected value of each range (e.g., the 0.333 cumulative probability level corresponds to a value of 0.95; the expected value of a uniform distribution between 0.85 and 0.95 is 0.9). Note that if a triangular distribution had been specified, the values would have been different, since the expected value in each range will be shifted toward 1.0.

The three discrete values described above would define three temperature categories which define variability across the repository: 33% of the packages would have a temperature of  $0.9 \times TEMPV_i(t)$ , 33% of the packages would have a temperature of  $1.0 \times TEMPV_i(t)$ , and 34% of the packages would have a temperature of  $1.1 \times TEMPV_i(t)$ .

A single value of  $TEMPV$  of 1.0 (implying no variability in temperature) is the default for RIP.

The actual temperature at a particular waste package is referenced by the internal environmental factor  $TEMP$  (which is completely analogous to the environmental factor  $CONTAC$  discussed previously). Other model parameters can subsequently be defined as a function of temperature, by referencing the name  $TEMP$ . At any given time,  $TEMP$ , as shown above, is computed internally by RIP as follows:

$$TEMP = TEMPV \times TEMPV_i(t)$$

In the example illustrated in Figure 5-1, "hot", "warm" and "cool" refer to different values of  $TEMPV$  (e.g., 1.1, 1.0, 0.9). Hence, in this simple example,  $TEMPV$  and  $CONTAC$  are the environmental factors which define the waste package groups. Since  $TEMPV$  varies between waste package groups,  $TEMP$  also varies between waste package groups.

Having described the concepts behind the representation of temperature conditions in RIP, the manner in which the various temperature parameters are entered can be described. Selecting [Temperature Conditions] from the menu under [Near Field Conditions] provides access to the following input window:

Repository Integration Program (Rev. 1.00)			
General	<u>PA Model</u>	Strategy Model	Results
User's Guide Tutorial			
[ ]			
<u>Waste Package</u>	Pathways	Events	Make Transfer File
[ ]			
<u>Near Field Conditions</u>	Waste Package Description	Make WP File	
[ ]			
Water Contact Mode	<u>Temperature Conditions</u>	Other Factors	Show Groups
[ ]			
<u>Temperature Conditions</u>			
Repository mean temperature			
Incremental temperature at edge of WP			
Define temperature variability			
Rewetting temperature : REWET (78 )			

The first two fields provide access to windows describing the mean temperature history in the rock ( $T_m$ ) and the incremental temperature above the mean temperature at the edge of each type of waste package ( $T_i$ ), respectively. The third field provides access to a window describing temperature variability throughout the repository. The fourth field is a parameter data field.

**Repository Mean Temperature:** This field provides access to the following input window for defining  $T_m$ :

Repository Integration Program (Rev. 1.00)			
General	<u>PA Model</u>	Strategy Model	Results
User's Guide Tutorial			
[ ]			
<u>Waste Package</u>	Pathways	Events	Make Transfer File
[ ]			
<u>Near Field Conditions</u>	Waste Package Description	Make WP File	
[ ]			
Water Contact Mode	<u>Temperature Conditions</u>	Other Factors	Show Groups
[ ]			
<u>Temperature Conditions</u>			
<u>Repository mean temperature</u>			
Incremental temperature at edge of WP			
Define temperature variability			
Rewetting temperature : REWET (78 )			
[ ]			
<u>Repository Mean Temperature</u>			
Time (Yrs)	Temperature		
100	150	Temperature Uncertainty	
1000	100	TEMPUN	(1 )
5000	75		
10000	60	Time Uncertainty	
100000	30	TIMEUN	(1 )
0	0		
F1=View Temperature			

**Time/Temperature:** These pairs must be entered as numbers (constants). RIP subsequently interpolates (linearly) in time between these pairs. The first and last points are interpolated horizontally. For example, if the first pair is



specified as (time = 1000, temperature = 200), a temperature of 200 will be assigned to all times previous to time = 1000. Time is specified in years and must be non-negative. Temperature can be specified in any desired units.

**Uncertainty in Temperature:** This is used to specify the uncertainty in the temperature time history defined above. The user can specify a parameter as a constant, a stochastic, or a function. The parameter can not, however, be specified as a function of time. Each realization, all of the inputs described above in the Temperature column are multiplied by the realized value of this parameter. *It is intended that the user specify this parameter as a stochastic centered about the value 1* (e.g., a triangular distribution centered at 1, with a minimum of 0.5 and a maximum of 2). This essentially shifts the temperature history up or down each realization, as illustrated schematically in Figure 5-2. The user should be especially careful not to specify nonsensical input here (e.g., distributions that could result in unrealistic temperature histories).

**Uncertainty in Time:** This is also used to specify the uncertainty in the temperature time history defined above. The user can specify a parameter as a constant, a stochastic, or a function. The parameter can not, however, be specified as a function of time. Each realization, all of the inputs described above in the Time column are multiplied by the realized value of this parameter. *It is intended that the user specify this parameter as a stochastic centered about the value 1* (e.g., a triangular distribution centered at 1, with a minimum of 0.5 and a maximum of 2). This essentially shifts the temperature history left or right each realization, as illustrated schematically in Figure 5-2. The user should be especially careful not to specify nonsensical input here (e.g., distributions that could result in unrealistic temperature histories).

The <F1> key plots (to the screen) the temperature history specified by the user. Note that for a PC running DOS 5.0, the plot can be sent directly to the printer by typing <Shift>-<PrintScreen>, as discussed in Section 2.4. <Esc> erases the plot and returns to the input screen.

**Incremental Temperature at Edge of WP:** This field provides access to the following window for defining  $T_i$ :

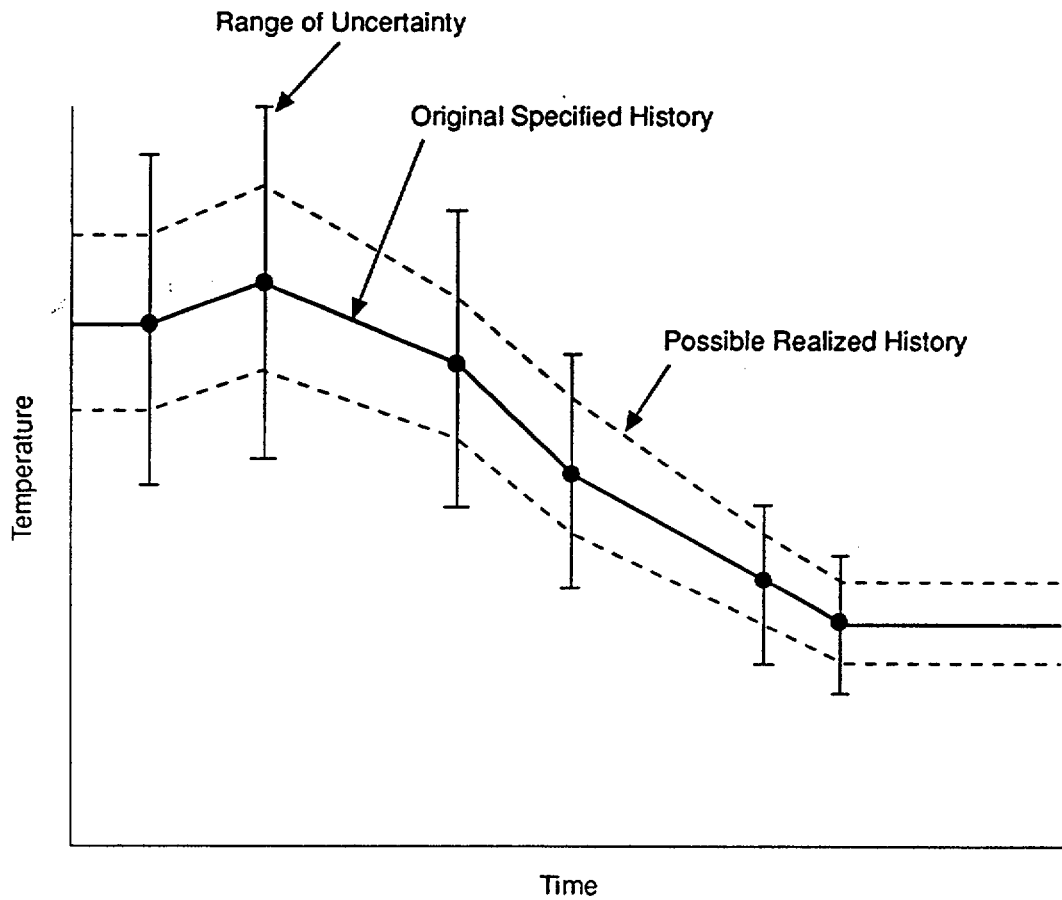


FIGURE 5  
TEMPERATURE UNCERTAINTY  
IN TIME/TEMPERATURE PAIRS  
ARGONNE/MODEL DEVELOPMENT

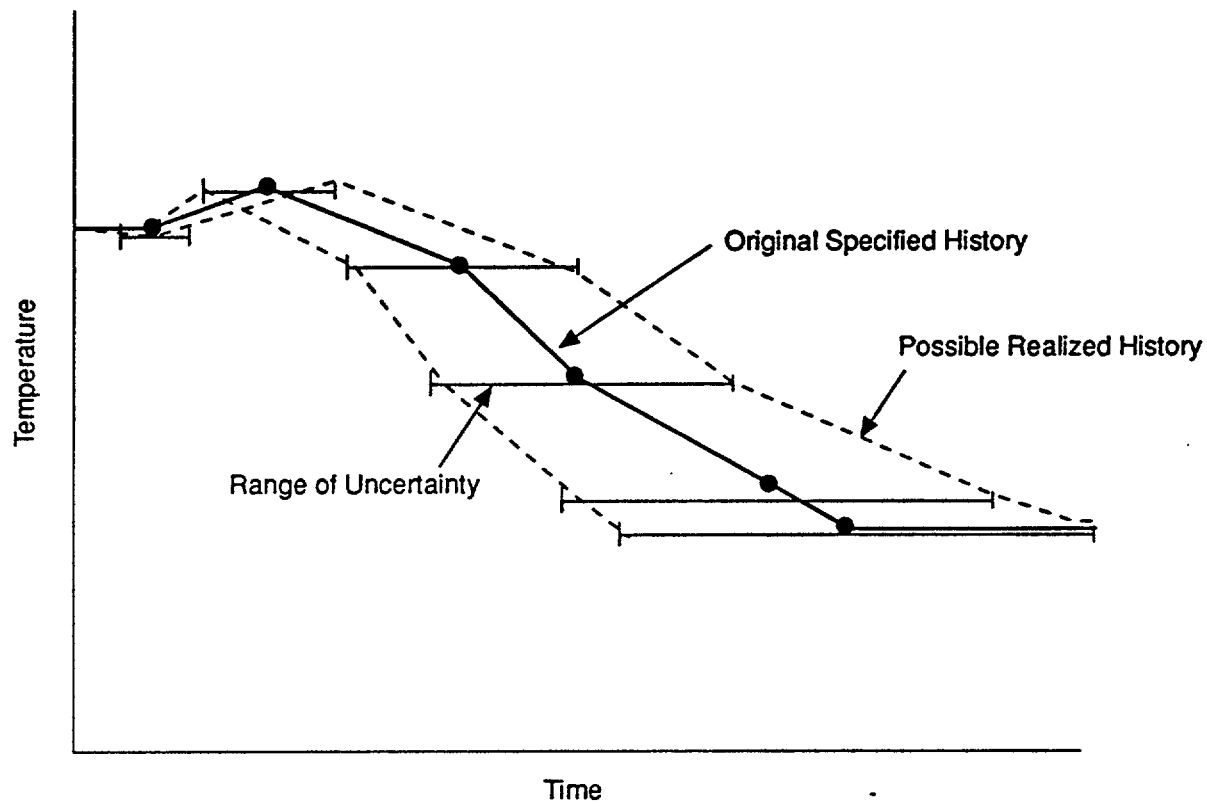


FIGURE 5-2b  
**TIME UNCERTAINTY IN  
 TIME/TEMPERATURE PAIRS**  
 ARGONNE/MODEL DEVELOPMENT

Repository Integration Program (Rev. 1.00)			
General	PA Model	Strategy Model	Results
User's Guide Tutorial			
Waste Package	Pathways	Events	Make Transfer File
Near Field Conditions		Waste Package Description	Make WP File
Water Contact Mode	Temperature Conditions	Other Factors	Show Groups
Temperature Conditions			
Repository mean temperature			
Incremental temperature at edge of WP			
Define temperature variability			
Rewetting temperature : REWET (78 )			
Incremental Temperature at edge of WP			
WP 1 of 2	WP ID : SPENT		
Temperature Uncertainty: WP1UN	(1	Time (Yrs)	Temperature
	)	100	50
		1000	40
		5000	25
		10000	10
		100000	0
		0	0

F1=Previous Record  
 F2=Next Record  
 F3=View Current Record  
 F4=View Current + Mean

The user must specify an incremental temperature history for each waste package type which has been defined. Conversely, temperature histories can only be specified for waste package types which have previously been defined. The <F1> and <F2> keys are used to move back and forth through the different waste package types which have been defined.

Recall that the average temperature at the edge of a waste package type  $i$  is computed by RIP as  $T_m(t) + T_i(t)$ .

**Time/Temperature:** These pairs must be entered as numbers (constants). RIP subsequently interpolates (linearly) in time between these pairs. The first and last points are interpolated horizontally. For example, if the first pair is specified as (time = 1000, temperature = 200), a temperature of 200 will be assigned to all times previous to time = 1000. Time is specified in years and must be non-negative. Temperature can be specified in any desired units.

**Uncertainty in Temperature:** This is used to specify the uncertainty in the incremental temperature time history defined above. The user can specify a parameter as a constant, a stochastic, or a function. The parameter can not, however, be specified as a function of time. Each realization, all of the inputs described above in the Temperature column are multiplied by the realized value of this parameter. It is intended that the user specify this parameter as a stochastic centered about the value 1 (e.g., a triangular distribution centered at 1, with a minimum of 0.5 and a maximum of 2). This essentially shifts the incremental temperature history up or down each realization. **WARNING:** the user should be especially careful not to specify nonsensical input here (e.g., distributions that could result in unrealistic temperature histories).

Note that the user only specifies an uncertainty in the temperature and can not

specify an uncertainty in time (with regard to the time/temperature pairs), as was done for the mean temperature in the rock.

The <F3> key provides a plot of the incremental temperature history just specified ( $T_i$ ). <F4> provides a plot of the incremental temperature history plus the mean temperature history ( $T_m + T_i$ ), showing the actual history of the average temperature at the waste package. Note that for a PC running DOS 5.0, screen plots can be sent directly to the printer by typing <Shift>-<PrintScreen>, as discussed in Section 2.4. <Esc> erases the plot and returns to the input screen.

**Define Temperature Variability:** This field provides access to the following window for defining temperature variability:

Repository Integration Program (Rev. 1.00)			
General	PA Model	Strategy Model	Results
User's Guide Tutorial			
[ ]			
Waste Package	Pathways	Events	Make Transfer File
[ ]			
Near Field Conditions	Waste Package Description		Make WP File
[ ]			
Water Contact Mode	Temperature Conditions	Other Factors	Show Groups
[ ]			
Temperature Conditions			
Repository mean temperature			
Incremental temperature at edge of WP			
Define temperature variability			
Rewetting temperature : REWET (78 )			
[ ] Define Temperature Variability			
Environmental Factor ID : TEMP			
Define type of variability : Uniform			
Define symmetric variability about 1 : 0.3			
Define Discretization Levels : 0.33 0.66 1			
0 0 0			

**Define type of variability:** This defines the type of distribution describing the temperature variability. The distribution can be either uniform or triangular, the user toggling between the two with the <PgUp> or <PgDn> keys.

**Define symmetric variability about 1:** This further defines the distribution representing the temperature variability. The uniform or triangular distribution is symmetric about 1. This particular field specifies the magnitude of the symmetric variability. For example, if a uniform distribution is specified, and a value of 0.3 is entered here, the uniform distribution defined would range from 0.7 to 1.3. This value must be specified as a number which is greater than or equal to 0 and less than or equal to 1. A value of 0 indicates no variability.

**Define Discretization Levels:** These are cumulative probability levels by which the continuous variability distribution defined directly above is to be converted to a discrete distribution. The manner in which this is done is described at the beginning of this section (5.1.1.2). These must be input as numbers which are greater than or equal to zero and less than or equal to 1. Because they are

*cumulative* probability levels, they must increase from first to last (left to right), and the last specified value must be 1 (any remaining 0's after a 1 are ignored).

**Rewetting Temperature:** This is the temperature at which the waste packages rewet. It is only appropriate for repositories in which water can be driven away during a thermal period. For saturated repositories where this is not possible, the rewetting temperature should simply be set to a large number (greater than the highest possible waste package temperature specified above). It may be described as a constant, a stochastic or as a function (e.g., as a function of water contact mode, or repository level flux). It can not, however, be a function of time.

### 5.1.1.3 Other Factors

Selecting [Other Factors] from the menu under [Near Field Conditions] provides access to input windows for defining other environmental factors (e.g., pH, stress, redox conditions) whose variability the user wishes to include in the model. The user first defines the name (ID) of the environmental factor (e.g., PH) and the average value about which the environmental factor's variability is to be centered (e.g., 7). Note that this average value need not be a constant, and may be described stochastically or as a function. A variability distribution is then defined in the same manner as described for temperature (e.g., 0.9, 1.0, 1.1). We refer to each of these discrete values in the distribution as the *variability factor* for each environmental factor category. In this example, we would refer to this factor as PH\_V (that is, the environmental factor name with "\_V" appended on to the end). PH\_V is analogous to TEMPV discussed in Section 5.1.1.2. This would define a number of discrete PH categories (e.g.,  $0.9 \times 7 = 6.3$ ,  $1.0 \times 7 = 7.0$ ,  $1.1 \times 7 = 7.7$ ).

The user-defined environmental factor ID (in this case PH) is completely analogous to the internal environmental factors CONTAC and TEMP discussed previously. That is, other model parameters could subsequently be defined as a function of the pH, which would be variable through the repository, by referencing the environmental factor name PH. In the example illustrated in Figure 5-1, adding the three pH categories (e.g., "acidic" corresponding to 6.3, "neutral" corresponding to 7.0, and "basic" corresponding to 7.7) would add a fourth dimension to the diagram, resulting in  $3 \times 3 \times 2 \times 3 = 54$  waste package groups. As shown above, the actual value of the environmental factor (in this case, the pH) at any time in any particular group is computed as the product of the average value of the factor and the variability factor. In this example,

$$PH = (\text{average value of pH}) \times PH\_V$$

Because PH\_V varies from group to group, PH (which can be referenced by the user) varies from group to group.

The following input window is accessed by selecting [Other Factors] from the menu under [Near Field Conditions]:

Repository Integration Program (Rev. 1.00)			
General	<u>PA Model</u>	Strategy Model	Results
User's Guide Tutorial			
[ ]			
<u>Waste Package</u>	Pathways	Events	Make Transfer File
[ ]			
<u>Near Field Conditions</u>	Waste Package Description	Make WP File	
[ ]			
Water Contact Mode	Temperature Conditions	<u>Other Factors</u>	Show Groups
[ ]			
Environmental Factors			
Environmental Factor ID : PH	Factor 1 of 1		F1=Prev
Parameter : MEANPH (7.2 )			F2=Next
Define type of variability : Uniform			F3=Add
Define symmetric variability about 1 : 0.1			F4=Del
Define Discretization Levels : 0.5 1	0	0	
	0	0	

**Environmental Factor ID:** This is the name by which the environmental factor can be referenced by the user. It is limited to six characters.

**Parameter:** This is the parameter which defines the average value of the environmental factor about which the variability is to be defined. It can be specified as a constant, a stochastic, or a function.

**Define type of variability:** This defines the type of distribution describing the environmental factor's variability. The distribution can be either uniform or triangular, the user toggling between the two with the <PgUp> or <PgDn> keys.

**Define symmetric variability about 1:** This further defines the distribution representing the environmental factor's variability. The uniform or triangular distribution is symmetric about 1. This particular field specifies the magnitude of the symmetric variability. For example, if a uniform distribution is specified, and a value of 0.3 is entered here, the uniform distribution defined would range from 0.7 to 1.3. This value must be specified as a number which is greater than or equal to 0 and less than or equal to 1.

**Define Discretization Levels:** These are cumulative probability levels by which the continuous variability distribution defined directly above is to be converted to a discrete distribution. The manner in which this is done is described at the beginning of Section 5.1.1.2. These must be input as numbers which are greater than or equal to zero and less than or equal to 1. Because they are *cumulative* probability levels, they must increase from first to last (left to right), and the last specified value must be 1 (any remaining 0's after a 1 are ignored).

The user can move through environmental factors which have been defined with the <F1> and <F2> keys. Environmental factors can be added or deleted with the <F3> and <F4> keys, respectively.

#### 5.1.1.4 Show Groups

Selecting [Show Groups] from the menu under [Near Field Conditions] does not provide

access to any input windows. Rather, it provides a summary display of all of the waste package groups which have been defined by the user. The display window appears as follows:

[ ] Repository Integration Program (Rev. 1.00 )

Group #	# WPs	WP TYPE	CONTAC	TEMPV	PH_V
1	1980	1	1	0.7990	0.9000
2	1980	1	1	0.7990	1.1000
3	1980	1	1	0.9970	0.9000
4	1980	1	1	0.9970	1.1000
5	2040	1	1	1.1980	0.9000
6	2040	1	1	1.1980	1.1000
7	1485	1	2	0.7990	0.9000
8	1485	1	2	0.7990	1.1000
9	1485	1	2	0.9970	0.9000
10	1485	1	2	0.9970	1.1000
11	1530	1	2	1.1980	0.9000
12	1530	1	2	1.1980	1.1000
13	1485	1	3	0.7990	0.9000
14	1485	1	3	0.7990	1.1000
15	1485	1	3	0.9970	0.9000
16	1485	1	3	0.9970	1.1000
17	1530	1	3	1.1980	0.9000
18	1530	1	3	1.1980	1.1000

In this particular example, the user has defined (using the previous input windows) 1 waste package type, 3 water contact modes, 3 temperature categories, and 2 categories of the user-defined environmental factor PH. This results in  $1 \times 3 \times 3 \times 2 = 18$  waste package groups.

For each group, the number of waste packages, the waste package type, the water contact mode (CONTAC), the temperature variability factor (TEMPV), and the variability factor for other user-defined environmental factors (in this case, PH\_V) are displayed. the screen shown above indicates that waste package group 1 has 1980 waste packages of type 1 in water contact mode 1, with a temperature history 0.799 times the average temperature, and a pH 0.9 times the average pH.

Note that if groups are defined stochastically, the values shown [Show Groups] are computed based on expected values.

[Show Groups] is a useful option for summarizing and verifying the waste package-to-waste package variability that the user has specified.

### 5.1.2 Describing Waste Package Behavior and Radionuclide Release

Selecting [Waste Package Description] from the menu under [Waste Package] provides access to input windows for specifying the details of waste package behavior and radionuclide release. The following input window is displayed:



Repository Integration Program (Rev. 1.00)			
General	PA Model	Strategy Model	Results
User's Guide Tutorial			
[ ]			
Waste Package	Pathways	Events	Make Transfer File
[ ]			
Near Field Conditions	Waste Package Description	Make WP File	
[ ]			
Waste Package Database Editor			
ID : SPENT	Package 1 of 1		F1=Prev
DESC: spent fuel waste packages			F2=Next
			F3=Add
			F4=Del
Number of Packages	: 30000		
Mass Waste per Package (kg):	2350	as MTHM :	2.08
Waste Burnup (MWd/MTHM):	60000		
Is mass transfer/bound exposure independent of failure mode? : Yes			
Mass Transfer			
Exposure			
Primary Container Failure Definition	Plotting Options		
Secondary Container Failure Definition	» Rewetting time : 0		
Inventory	» Cont. fail time : 0		
	Plot axes : automatic		

The user can add additional waste package types with the <F3> key. Waste package types can be deleted with the <F4> key. The <F1> and <F2> keys can be used to move between defined waste package types.

Specification of multiple waste package types allows the user to represent the different kinds of waste packages which will be emplaced in the repository (e.g., spent fuel, high level defense waste). This is necessary because different kinds of waste packages may behave quite differently (e.g., each type having different inventories, container failure rates, and/or mass exposure and mass transfer characteristics).

Note that different waste package types could be defined to represent waste packages which are similar but behave slightly differently. For example, a waste package type could be defined which had slightly cooler temperatures to represent those waste packages which are emplaced near the edge of the repository.

The specific input fields in the waste package window shown above are discussed in the following sections.

#### 5.1.2.1 Waste Package Characteristics

For each waste package type, the following input is required:

**ID:** This is a six character identifier for the waste package type.

**DESC:** This is a 60 character description for the waste package type.

**Number of Packages:** This is the number of waste packages of this particular type emplaced in the repository. This must be a non-negative integer.

**Mass Waste per Package (kg):** This is the total mass of the *waste matrix* in each waste package. For spent fuel packages, for example, it would essentially be input as the

mass of  $\text{UO}_2$  (which makes up the majority of the waste matrix). This must be a non-negative constant.

**Mass Waste as MTHM:** This is the total mass of waste as MTHM (metric tons of heavy metal). It must be a non-negative constant. This input is necessary in order to provide normalized releases per 40 CFR Part 191. If the user is not interested in this particular form of the output, he/she can choose not to compute it (as discussed in Sections 5.1.3 and 5.4), and this input need not be specified here. Appendix C discusses in detail how the normalization is carried out.

For spent fuel waste, this input is straightforward, as this is often the form in which the available data is reported. It can also be computed from the mass in kg by assuming that for all practical purposes the waste matrix consists of  $\text{UO}_2$ , and using the stoichiometric ratio of  $\text{UO}_2$  to U ( $270/238 = 1.13$ ).

For defense wastes, this should be input in terms of the MTHM of the fuel from which the reprocessed waste was generated.

**Waste Burnup (MWd/MTHM):** This is the waste burnup in terms of MegaWatt-days/MTHM. It must be a non-negative constant. This input is also necessary in order to provide normalized releases per 40 CFR Part 191. If the user is not interested in this particular form of the output, he/she can choose not to compute it (as discussed in Sections 5.1.3 and 5.4), and this input need not be specified here. Appendix C discusses in detail how the normalization is carried out.

When specifying a waste burnup for spent fuel, the user should make sure that the specified burnup is consistent with the radionuclide inventories specified elsewhere (since the inventories should vary with burnup). For defense wastes, this should be input in terms of the burnup of the fuel from which the reprocessed waste was generated.

**Is mass transfer/bound exposure independent of failure mode?:** This is an option field which can be selected as Yes or No. If answered as No, several of the input parameters describing mass transfer and bound exposure become dependent on the mode in which the waste package primary container failed (discussed in detail in Sections 3.2.3.3 and 3.2.4.1 of the RIP Theory and Capabilities Manual). If answered as Yes, all mass transfer and bound exposure parameters are independent of the mode of container failure. As will be detailed subsequently, this option field affects the appearance of several subsequent windows ("Mass Transfer", "Exposure", and "Primary Container Failure Definition").

#### 5.1.2.2 Mass Transfer Parameters

**Mass Transfer:** This field provides access to an input window for describing mass transfer out of and away from the waste package. The user should refer to Section 3.2.4 of the RIP Theory and Capability Manual to review the manner in which mass transfer is represented in RIP.

In summary, mass transfer is represented in RIP by the following equations:

$$k_p(n) = k_{p,a}(n) + k_{p,d}(n)$$

$$k_{p,a}(n) = \text{INFIL} \times \text{CATCH} \times C(n)$$

$$k_{p,d}(n) = D_{\text{eff}}(n) \times \omega \times C(n)$$

$$C(n) = \text{MIN}\left[\frac{M_p(n)}{V}, C_s(n)\right]$$

where:

$k_{tp}(n)$  = maximum possible mass transfer rate (from a single waste package) for radionuclide  $n$  [g/yr];

$k_{tp,a}(n)$  = maximum possible advective mass transfer rate (from a single package) for radionuclide  $n$  [g/yr];

$k_{tp,d}(n)$  = maximum possible diffusive mass transfer rate (from a single package) for radionuclide  $n$  [g/yr];

INFIL = repository level groundwater infiltration rate [m/yr];

CATCH = effective catchment area for a waste package [m<sup>2</sup>];

$D_{\text{eff}}(n)$  = effective diffusion coefficient for radionuclide  $n$  [m<sup>2</sup>/yr];

$\omega$  = geometric factor for diffusive mass transfer [m];

$C(n)$  = the effective concentration of radionuclide  $n$  at the waste package [g/m<sup>3</sup>];

$C_s(n)$  = saturation concentration of radionuclide  $n$  [g/m<sup>3</sup>];

$V$  = volume of water in contact with waste matrix in a failed waste package [m<sup>3</sup>]; and

$M_p(n)$  = average exposed mass of radionuclide  $n$  per failed waste package (computed internally by RIP) [g].

As mentioned above, the specific input required varies depending on whether mass

transfer is described as a function of container failure mode. If mass transfer is independent of failure mode, the following window is accessed by this field:

Repository Integration Program (Rev. 1.00)				
General		PA Model	Strategy Model	Results
User's Guide Tutorial				
Waste Package		Pathways	Events	Make Transfer File
Near Field Conditions		Waste Package Description		Make WP File
Waste Package Database Editor				
ID	MASS TRANSFER			rev
DE	Repository Infiltration Rate (m/yr): INFIL	(10)	)	ext
	Water Volume Contacting Matrix (m <sup>3</sup> m <sup>3</sup> ): VOL	(1)	)	dd
NU	Effective Catchment Area (m <sup>2</sup> ): CATCH	(50)	)	el
Ma	Geometric Factor for Diffusion (m): OMEGA	(1)	)	
Wa				
Is mass transfer/bound exposure independent of failure mode? : Yes				
Mass Transfer				
Exposure		Plotting Options		
Primary Container Failure Definition		» Rewetting time : 0		
Secondary Container Failure Definition		» Cont. fail time : 0		
Inventory		Plot axes : automatic		

**Repository Infiltration Rate:** This is the groundwater infiltration rate (in m/yr) at the repository level. It can be described as a constant, a stochastic, or a function. It should always be specified as non-negative. Typically, it will be described as a function of surface infiltration, which in turn may be a function of time-varying precipitation.

**Water Volume Contacting Matrix:** This is the volume of water (in m<sup>3</sup>) in contact with waste matrix in a failed container. It can be described as a constant, a stochastic, or a function. It should always be specified as non-negative. It is used to compute the effective concentration at a failed waste package for radionuclides which are not solubility controlled. It represents the volume of water which is at the effective concentration C(n). Specification of a very small value here is conservative in that it ensures that the effective concentration is equal to the saturation concentration.

**Effective Catchment Area:** This is the effective catchment area (in m<sup>2</sup>) for one waste package. It can be described as a constant, a stochastic, or a function. It should always be specified as non-negative. It is multiplied by the repository infiltration rate to compute the effective volume rate of flow of water which is at a concentration of C(n).

**Geometric Factor for Diffusion:** The geometric diffusion factor has units of length (m), and must represent *steady-state* diffusive transport (i.e., it cannot be a function of time). It can be described as a constant, a stochastic, or a function. It should always be specified as non-negative. Various forms for this parameter are discussed in the Theory and Capabilities Manual, Section 3.2.4.1.

If mass transfer/bound exposure is indicated to be described as a function of container failure mode, "Effective Catchment Area" and "Geometric Factor for Diffusion" are not included in this window, as they must be input separately for each container failure mode:

Repository Integration Program (Rev. 1.00)			
General	PA Model	Strategy Model	Results
User's Guide Tutorial			
Waste Package Pathways Events Make Transfer File			
Near Field Conditions Waste Package Description Make WP File			
Waste Package Database Editor			
ID	MASS TRANSFER		rev
DE	Repository Infiltration Rate (m/yr): INFIL	(10)	ext
	Water Volume Contacting Matrix (m³m³m): VOL	(1)	del
NU	Mass Waste per Package (kg): 2350 as MTHM : 2.0799999		
	Waste Burnup (MWd/MTHM): 60000		
Is mass transfer/bound exposure independent of failure mode? : No			
Mass Transfer Exposure			
Primary Container Failure Definition		Plotting Options	
Secondary Container Failure Definition		» Rewetting time : 0	
Inventory		» Cont. fail time : 0	
		Plot axes : automatic	

The user should note that RIP currently only represents steady-state mass transfer processes. It does not represent transient effects. The implications of this are discussed in detail in the RIP Theory and Capabilities Manual, Section 3.2.4.

### 5.1.2.3 Parameters Describing Bound Exposure (Alteration/Dissolution of the Waste Matrix)

**Exposure:** This field provides access to an input window for describing exposure of the radionuclide inventory bound within the waste matrix. The user should refer to Section 3.2.3 of the RIP Theory and Capability Manual to review the manner in which bound exposure is represented in RIP.

In summary, bound exposure is defined by the following equations:

$$k_{a/d} = k_{wat} + k_{air}$$

$$k_{wat} = R_{dis} \times S \times f_w$$

where:

$k_{a/d}$  = total fractional rate of matrix alteration/dissolution [1/yr];

$k_{wat}$  = fractional alteration/dissolution rate of matrix in water [1/yr];

$k_{air}$  = fractional alteration rate of matrix in air [1/yr];

$R_{dis}$  = aqueous matrix alteration/dissolution rate [ $\text{g}/\text{m}^2/\text{yr}$ ];

$S$  = effective surface area of waste matrix in a failed container per unit mass [ $\text{m}^2/\text{g}$ ];  
and

$f_w$  = fraction of waste matrix surface area in a failed container which is wet.

As mentioned above, the specific input required varies depending on whether bound exposure is described as a function of container failure mode. If bound exposure is independent of failure mode, the following window is accessed by this field:

Repository Integration Program (Rev. 1.00)				
General		PA Model	Strategy Model	Results
User's Guide Tutorial				
[ ]				
Waste Package		Pathways	Events	Make Transfer File
[ ]				
Near Field Conditions		Waste Package Description	Make WP File	
[ ]				
Waste Package Database Editor				
ID [ ]	EXPOSURE			
DE	Air Alteration Rate	(1/yr): AIR	(0 )	rev
	Matrix Dissolution Rate	(g/m <sup>2</sup> /yr): DISS	(5 )	ext
Nu	Surface Area of Matrix	(m <sup>2</sup> /g): 0.0001	(Constant )	el
Ma	Fraction of Waste Wetted	(0<x<1): 0.1	(Constant )	
Wa				
Is mass transfer/bound exposure independent of failure mode? : Yes				
Mass Transfer				
Exposure				
Primary Container Failure Definition		Plotting Options		
Secondary Container Failure Definition		» Rewetting time : 0		
Inventory		» Cont. fail time : 0		
		Plot axes : automatic		

**Air Alteration Rate:** This is the fractional air alteration (oxidation) rate (in  $\text{yr}^{-1}$ ) for the waste matrix. As discussed in the Theory and Capabilities Manual, RIP allows the user to assume that air alteration can have the effect of exposing bound radionuclides by making them accessible due to a change in structure or a loss of integrity of the waste matrix. Under these conditions, the structure of the matrix is assumed to be altered to the extent that exposure of bound radionuclides is no longer limited by the matrix dissolution. That is, the bound inventory could be exposed via air oxidation of the matrix without having to dissolve the matrix. If this is not a valid mechanism, of course, this input should be set to zero. It can be described as a constant, a stochastic, or a function. It should always be specified as non-negative.

**Matrix Dissolution Rate:** This is the aqueous alteration/dissolution rate for the waste matrix (in  $\text{g}/\text{m}^2/\text{yr}$ ). It can be described as a constant, a stochastic, or a function. It should always be specified as non-negative.

**Surface Area of Matrix:** This is the surface area of matrix *per unit mass* ( $\text{m}^2/\text{g}$ ). It is multiplied by the "matrix dissolution rate" and the "fraction of waste wetted" to compute the fractional alteration/dissolution rate. It can be

described as a constant, a stochastic, or a function. It should always be specified as non-negative.

**Fraction of Waste Wetted:** This is the fraction of the surface area of waste in a failed container which is actually wetted. It can be described as a constant, a stochastic, or a function. Because it represents a *fraction*, it must be greater than or equal to zero and less than or equal to one.

If mass transfer/bound exposure is indicated to be described as a function of container failure mode, "Fraction of Waste Wetted" is not included in this window, as it must be input separately for each container failure mode:

Repository Integration Program (Rev. 1.00)				
General		PA Model	Strategy Model	Results
User's Guide Tutorial				
[ ]				
Waste Package		Pathways	Events	Make Transfer File
[ ]				
Near Field Conditions		Waste Package Description	Make WP File	
[ ]				
Waste Package Database Editor				
ID	EXPOSURE			rev
DE	Air Alteration Rate	(1/yr): AIR	(0)	ext
	Matrix Dissolution Rate	(g/m <sup>2</sup> /yr): DISS	(5)	dd
Nu	Surface Area of Matrix	(m <sup>2</sup> /g): 0.0001	(Constant)	el
Ma				
Waste Burnup (MWd/MTHM): 60000				
Is mass transfer/bound exposure independent of failure mode? : No				
Mass Transfer				
Exposure				
Primary Container Failure Definition		Plotting Options		
Secondary Container Failure Definition		» Rewetting time : 0		
Inventory		» Cont. fail time : 0		
		Plot axes : automatic		

#### 5.1.2.4 Describing Waste Package Failure

**Plotting Options:** Several plotting options are presented on the lower right hand side of the waste package data base editor screen. These all pertain to creating screen and hardcopy plots of the failure distributions. As will be discussed, these plots can be created from within the "Primary Container Failure Definition" and "Secondary Container Failure Distribution" windows.

**Rewetting time:** As will be discussed below, primary container failure modes can be conditioned on the time of rewetting. That is, they may not start until the waste package is rewet. Since the rewetting time may vary throughout the repository, in order to plot the primary container failure distribution, the user must specify a rewetting time upon which the plot can be based. This must be a non-negative number.

**Cont. fail time:** As will be discussed below, secondary container failure modes can be conditioned on the time of primary container failure. That is, they may not start until the primary container has failed. Since the primary container

failure time varies from container to container, in order to plot a single secondary failure distribution, the user must specify a primary container failure time upon which the plot can be based. This must be a non-negative number.

**Plot axes:** For either a plot of the primary or secondary failure distributions, the plot axes can be manually or automatically specified. If manual is chosen, RIP will prompt the user for the maximum and minimum for both the x and the y axes.

**Primary Container Failure Definition:** This field provides access to an input window for describing the primary container failure distribution for the waste packages. The user should refer to Section 3.2.2 of the RIP Theory and Capability Manual to review the manner in which container failure is represented in RIP. As mentioned above, the specific input required varies depending on whether mass transfer/bound exposure is described as a function of container failure mode. If mass transfer/bound exposure is independent of failure mode, the following window is accessed by this field:

[ ] Repository Integration Program (Rev. 1.00 )			
General	PA Model	Strategy Model	Results
[ ] User's Guide Tutorial			
[ ] Waste Package Pathways Events Make Transfer File			
[ ] Near Field Conditions Waste Package Description Make WP File			
[ ] Primary Container Failure Definition			
Description : general corrosion			
Start when Rewet : No		Fail Mode 1 of 3	
If no, Start time : 0		(Constant ) for WP# 1	
Probability : 1		(Constant ) WP ID : SPENT	
Aging Rate : 1		(Constant )	
Distribution Type : Weibull			
Define Failure Mode		F1=PREV F2=NEXT F3=ADD F4=DEL F5=VIEW F6=HARD	
Primary Container Failure Definition		» Rewetting time : 0	
Secondary Container Failure Definition		» Cont. fail time : 0	
Inventory		Plot axes : automatic	

The user can define as many separate failure modes as desired, and these modes are then combined by RIP in an appropriate manner to produce the total failure distribution. The user can add additional failure modes with the <F3> key. Failure modes can be deleted with the <F4> key. The <F1> and <F2> keys can be used to move between defined modes. The <F5> and <F6> keys are used to view the total failure distribution and will be described in greater detail below.

**Description:** This is a 60 character description for the failure mode.

**Start when Rewet:** This is an option field with two choices: Yes and No. If the user specifies Yes, the failure mode does not start until the waste package is rewet (i.e., until the temperature at the edge of the waste package is less than or equal to the rewetting temperature). If the user specifies No, the failure mode will start at whatever time is specified in the following field.



**If no, start time:** This is the start time for the failure mode if the user specifies No above (it is ignored if the user specifies Yes above). This can be a constant, a stochastic, or a function. It must, however, be non-negative. Also, because it represents a time, it cannot be specified as a function of time. The default value is 0.

**Probability:** This represents the probability that a random waste package will be susceptible to this particular failure mode (i.e., the fraction of waste packages which are susceptible to this mode). It acts as a scaling factor for the failure distribution. This can be a constant, a stochastic, or a function. It must, however, be greater than or equal to zero and less than or equal to one. Furthermore, it can not be a function of time. The default value is one.

For most container failure modes, this parameter will have only one of two values: 1 or 0. In this sense, it acts as an "on/off" indicator for failure modes. This might be used, for example, if failure modes depended on variable environmental factors. For example, if the particular failure mode was only active under a certain water contact mode (identified by  $\text{CONTAC} = 1$ ), and inactive in the others, the parameter could be specified as follows:

$$\text{IF}(\text{CONTAC}=1,1,0)$$

That is, if  $\text{CONTAC} = 1$ , the probability = 1, else the probability = 0. As will be discussed below, the aging rate can also be used in this manner to control which failure modes are active.

In some cases, the user may wish to specify this parameter as a fraction (as opposed to 0 or 1). For example, if a certain percentage (say 0.01%) of the waste packages failed due to faulty emplacement and the user wished to represent this mode of failure by a separate distribution (as opposed to incorporating it into other failure mode distributions), he/she would simply assign a probability of 0.0001 to whatever distribution was defined.

**Aging Rate:** Whereas the probability described above has the effect of scaling the frequency axis (i.e., the y-axis) of the failure distribution, the aging rate has the effect of stretching and/or shrinking the time axis (i.e., the x-axis) of the failure distribution. This parameter can be a constant, a stochastic, or a function, and, as will be discussed below, it *can* vary with time. It must, however, be non-negative. The default value is 1. The aging rate is discussed in detail in Section 3.2.2.3 of the Theory and Capabilities Manual.

The aging rate allows failure distributions to respond directly to time-varying environmental conditions. For example, the aging rate could be specified to increase (or decrease) with changes in temperature, repository level infiltration, or some other user defined parameter or environmental factor (such as pH). It is easiest to illustrate the use and physical significance of this parameter by considering an example. Suppose that the aging rate for a certain mode was

described such that above a certain temperature (120°), the container aged (i.e., failed) at twice the rate as under the standard temperature under which the failure distribution for the mode was defined. To accomplish this, the aging rate could be describes in the parameters module as follows:

IF(TEMP >= 120, 2, 1)

If the container was subjected to this elevated temperature for, say the first 1000 years, the *effective* age of the container (with respect to that failure mode) would then be computed as 2000 years, twice the actual age. Hence, if the original failure distribution was defined such that 25% of the containers failed within 1000 years, and 60% of the containers failed within 2000 years, the model would compute that under the elevated temperature conditions, 60% of the containers would fail within 1000 years.

Unlike the probability, the aging rate can be described directly as a function of time or as a function of some other time-varying parameter (e.g., temperature, pH, infiltration rate). Also, because the aging rate can either shrink or stretch the time axis (whereas the probability can only shrink the probability axis), it can be greater than one. When it is greater than one, it acts to accelerate the failure rate (with respect to the original distribution); when it is less than one, it acts to decelerate the failure rate.

In addition to using the aging rate in the manner specified above, it can also be used as an "on/off" switch for the failure mode (in the exact same manner as the probability described above). This is because an aging rate of 0 freezes the time axis, effectively turning off the mode.

**Distribution Type:** This specifies the form of the failure distribution. It is an option field with the following choices: Weibull, exponential, uniform, and degenerate. The mathematical forms of these four distributions are discussed in detail in Appendix F of the Theory and Capabilities Manual.

**Define Failure Mode:** This field provides access to input windows for specifying the parameters which define the failure mode distribution specified above. Each distribution type has a different input window:

Repository Integration Program (Rev. 1.00)			
General	<u>PA Model</u>	Strategy Model	Results
User's Guide Tutorial			
[ ]			
<u>Waste Package</u> Pathways Events Make Transfer File			
[ ]			
Near Field Conditions		<u>Waste Package Description</u>	Make WP File
[ ]			
Description	Weibull Failure Distribution		F1=PREV F2=NEXT F3=ADD F4=DEL F5=VIEW F6=HARD
Start when			
If no, Sta	Weibull slope ( $\alpha$ ) : 2.5	(Constant )	
Probabilit	= Mean Lifetime (B-e) : 5000	(Constant )	
Aging Rate	F1=Change distribution F2=Display F3=Hardcopy		
Distributi	Define Failure Mode		
Primary Container Failure Definition		» Rewetting time : 0	
Secondary Container Failure Definition		» Cont. fail time : 0	
Inventory		Plot axes : automatic	

Repository Integration Program (Rev. 1.00)			
General	<u>PA Model</u>	Strategy Model	Results
User's Guide Tutorial			
[ ]			
<u>Waste Package</u> Pathways Events Make Transfer File			
[ ]			
Near Field Conditions		<u>Waste Package Description</u>	Make WP File
[ ]			
Description	Exponential Failure Distribution		F1=PREV F2=NEXT F3=ADD F4=DEL F5=VIEW F6=HARD
Start when			
If no, Sta	Lambda (1/Yrs) : 0.0001	(Constant )	
Probabilit	F1=Change distribution F2=Display F3=Hardcopy		
Aging Rate	Distribution Type : Exponential		
Distribution Type	Define Failure Mode		
Primary Container Failure Definition		» Rewetting time : 0	
Secondary Container Failure Definition		» Cont. fail time : 0	
Inventory		Plot axes : automatic	

Repository Integration Program (Rev. 0.01)			
General	<u>PA Model</u>	Strategy Model	Results
User's Guide Tutorial			
[ ]			
<u>Waste Package</u> Pathways Events Make Transfer File			
[ ]			
Near Field Conditions		<u>Waste Package Description</u>	Make WP File
[ ]			
Description	Uniform Failure Distribution		F1=PREV F2=NEXT F3=ADD F4=DEL F5=VIEW F6=HARD
Start when			
If no, Sta	Duration (Yrs) : 4000	(Constant )	
Probabilit	F1=Change distribution F2=Display F3=Hardcopy		
Aging Rate	Distribution Type : Uniform		
Distribution Type	Define Failure Mode		
Primary Container Failure Definition		» Rewetting time : 0	
Secondary Container Failure Definition		» Cont. fail time : 0	
Inventory		Plot axes : automatic	

The Weibull requires two parameters, the Weibull slope and the mean lifetime. The exponential and the uniform are both defined by single parameters. The degenerate (in which waste packages are failed instantaneously) requires no parameters (only the starting time). All parameters describing distributions can be specified as constants, stochastics, or functions. The parameters must be non-negative, and can not be described as a function of time.

Note that three function keys are active in the windows displayed above. <F1> allows the user to change the distribution defined in the previous field while in this window. Upon exiting out of the window, the "Distribution Type" displayed will be updated. <F2> provides a screen plot of the distribution. Note that for a PC running DOS 5.0, the plot can be sent directly to the printer by typing <Shift>-<PrintScreen>, as discussed in Section 2.4. <F3> provides a hardcopy plot of the distribution. In this case, the user is prompted for the output format (currently only HPGL or HPGL2 can be chosen) and a filename. As was discussed above under "Plotting Options", the user must specify the rewetting time assumed for the plot, and the plot axes. Note that aging rates are not incorporated into plots of failure distributions.

Hitting <Esc> returns the user to the Primary Container Failure input window. We have previously discussed function keys <F1> through <F4>. <F5> and <F6> are used to plot *combined* failure distributions. After hitting <F5> or <F6>, the user is prompted for the primary failure mode numbers which are to be included in the plot. RIP then combines the specified modes in an appropriate manner and provides a screen plot or hardcopy plot of the combined distribution. Note that for a PC running DOS 5.0, screen plots can be sent directly to the printer by typing <Shift>-<PrintScreen>, as discussed in Section 2.4. For hardcopy plots (<F6>), the user is prompted for the output format (HPGL or HPGL2) and a filename.

As mentioned above, the specific input required for each primary container failure mode varies depending on whether mass transfer/bound exposure is described as a function of container failure mode. If mass transfer/bound exposure is *dependent* on failure mode, the following window is accessed by the "Primary Container Failure Definition" field:

Repository Integration Program (Rev. 0.01)			
General	PA Model	Strategy Model	Results
User's Guide Tutorial			
[ ] Waste Package Pathways Events Make Transfer File			
[ ] Near Field Conditions Waste Package Description Make WP File			
[ ] Primary Container Failure Definition			
Description : general corrosion			
Start when Rewet : No		Fail Mode 1 of 3	
If no, Start time : 0		(Constant ) for WP# 1	
Probability : 1		(Constant ) WP ID : SPENT	
Aging Rate : 1		(Constant )	
Distribution Type : Uniform			
Define Failure Mode			
Alteration/Dissolution/Mass Transfer		F1=PREV F2=NEXT F3=ADD F4=DEL F5=VIEW F6=HARD	
Primary Container Failure Definition		» Rewetting time : 0	
Secondary Container Failure Definition		» Cont. fail time : 0	
Inventory		Plot axes : automatic	

This is identical to the input window described above, with the exception of an additional access field: "Alteration/Dissolution/Mass Transfer".

**Alteration/Dissolution/Mass Transfer:** This field provides access to the following input window:

Repository Integration Program (Rev. 0.01)			
General	PA Model	Strategy Model	Results
User's Guide Tutorial			
[ ] Waste Package Pathways Events Make Transfer File			
[ ] Near Field Conditions Waste Package Description Make WP File			
[ ] Primary Container Failure Definition			
De [ ] ALTERATION/DISSOLUTION/MASS TRANSFER			
St	Effective Catchment Area	(m <sup>2</sup> ): CATCH	(50 )
If	Geometric Factor for Diffusion	(m): OMEGA	(1 )
Pr	Fraction of Waste Wetted	(0<x<1): 0.1	(Constant )
Ag			
Distribution Type : Uniform			F4=DEL
Define Failure Mode			F5=VIEW
Alteration/Dissolution/Mass Transfer			F6=HARD
Primary Container Failure Definition			» Rewetting time : 0
Secondary Container Failure Definition			» Cont. fail time : 0
Inventory			Plot axes : automatic

The "Effective Catchment Area", "Geometric Factor for Diffusion", and "Fraction of Waste Wetted" must be entered here *for this particular primary container failure mode*. These parameters were discussed above under *Mass Transfer* and *Exposure*, and have the same meaning, the only difference being that in this case they are a function of the mode of primary container failure, and hence must be specified separately for each primary container failure mode which is defined. The manner in which this information is used by RIP is discussed in detail in the RIP Theory and Capabilities Manual, Sections 3.2.3.3 and 3.2.4.1.

**Secondary Container Failure Definition:** This field provides access to an input window for describing the secondary container (e.g., cladding or pour canister) failure distribution for the waste packages. The user should refer to Section 3.2.2 of the RIP Theory and Capability Manual to review the manner in which container failure is represented in RIP. The following window is accessed by this field:

[ ] Repository Integration Program (Rev. 0.01 )			
General	PA Model	Strategy Model	Results
[ ] User's Guide Tutorial			
[ ] Waste Package Pathways Events Make Transfer File			
[ ] Near Field Conditions Waste Package Description Make WP File			
[ ] Secondary Container Failure Definition			
Description : creep rupture			
Start when Container Fails : No		Fail Mode 1 of 1	
Start Time : 0	(Constant )	for WP# 1	
Probability : 1	(Constant )	WP ID : SPENT	
Distribution Type : Exponential			F1=PREV
Define Failure Mode			F2=NEXT
			F3=ADD
			F4=DEL
			F5=VIEW
			F6=HARD
Primary Container Failure Definition		» Rewetting time : 0	
Secondary Container Failure Definition		» Cont. fail time : 0	
Inventory		Plot axes : automatic	

The user can define as many separate failure modes as desired, and these modes are then combined by RIP in an appropriate manner to produce the total failure distribution. The user can add additional failure modes with the <F3> key. Failure modes can be deleted with the <F4> key. The <F1> and <F2> keys can be used to move between defined modes. The <F5> and <F6> keys are used to view the total failure distribution and will be described in greater detail below.

**Description:** This is a 60 character description for the failure mode.

**Start when Container Fails:** This is an option field with two choices: Yes and No. If the user specifies Yes, the failure mode does not start until the primary waste package container fails. If the user specifies No, the failure mode will start at whatever time is specified in the following field.

**If no, start time:** This is the start time for the failure mode if the user specifies No above (it is ignored if the user specifies Yes above). This can be a constant, a stochastic, or a function. It must, however, be non-negative. Also, because it represents a time, it cannot be specified as a function of time. The default value is 0.

**Probability:** This represents the probability that the secondary waste package container will be susceptible to this particular failure mode (i.e., the fraction of waste packages which are susceptible to this mode). It acts as a scaling factor for the failure distribution. This can be a constant, a stochastic, or a function.

It must, however, be between 0 and 1. Furthermore, *it can not be a function of time*. The default value is 1.

For most container failure modes, this parameter will have only one of two values: 1 or 0. In this sense, it acts as an "on/off" indicator for failure modes. This might be used if failure modes depended on local environmental conditions. For example, if the particular failure mode was only active under a certain water contact mode (identified by  $\text{CONTAC} = 1$ ), and inactive in the others, the parameter could be specified as follows:

$\text{IF}(\text{CONTAC}==1,1,0)$

That is, if  $\text{CONTAC} = 1$ , the probability = 1, else the probability = 0. Unlike the primary container failure modes, secondary container failure modes *do not* have an aging rate which can also be used in this manner to control which failure modes are active.

In some cases, the user may wish to specify this parameter as a fraction (as opposed to 0 or 1). For example, if a certain percentage (say 0.01%) of the waste packages failed due to faulty emplacement (e.g., poor handling) and the user wished to represent this mode of failure by a separate distribution (as opposed to incorporating it into other failure mode distributions), he/she would simply assign a probability of 0.0001 to whatever distribution was defined.

**Distribution Type:** This specifies the form of the failure distribution. It is an option field with the following choices: Weibull, exponential, uniform, and degenerate. The mathematical forms of these four distributions are discussed in detail in Appendix F of the Theory and Capabilities Manual.

**Define Failure Mode:** This field provides access to input windows for specifying the parameters which define the failure mode distribution specified above. Each distribution type has a different input window; the windows are identical to those described above for the primary container failure modes.

The Weibull requires two parameters, the Weibull slope and the mean lifetime. The exponential and the uniform are both defined by single parameters. The degenerate (in which waste packages are failed instantaneously) requires no parameters (only a starting time). All parameters describing distributions can be specified as constants, stochastics, or functions. The parameters must be non-negative, and can not be described as a function of time.

Note that three function keys are active in the windows displayed above. <F1> allows the user to change the distribution defined in the previous field while in this window. Upon exiting out of the window, the "Distribution Type" displayed will be updated. <F2> provides a screen plot of the distribution. Note that for a PC running DOS 5.0, the plot can be sent directly to the printer by typing <Shift>-<PrintScreen>, as discussed in Section 2.4.

<F3> provides a hardcopy plot of the distribution. In this case, the user is prompted for the output format (currently only HPGL or HPGL2 can be chosen) and a filename. As was discussed above, the user must specify the rewetting time assumed for the plot, and the plot axes.

Hitting <Esc> returns the user to the Secondary Container Failure input window. We have previously discussed function keys <F1> through <F4>. <F5> and <F6> are used to plot *combined* failure distributions. After hitting <F5> or <F6>, the user is prompted for the secondary failure mode numbers which are to be included in the plot. RIP then combines the specified modes in an appropriate manner and provides a screen plot or hardcopy plot of the combined distribution. Note that for a PC running DOS 5.0, screen plots can be sent directly to the printer by typing <Shift>-<PrintScreen>, as discussed in Section 2.4. For hardcopy plots (<F6>), the user is also prompted for the output format (HPGL or HPGL2) and a filename.

#### 5.1.2.5 Describing Radionuclide and Elemental Properties

**Inventory:** This field provides access to a window for defining the characteristics of each radionuclide in the inventory of the waste package:

Repository Integration Program (Rev. 1.00)		
General	PA Model	Strategy Model
User's Guide Tutorial		
[ ]		
Waste Package	Pathways	Events
Make Transfer File		
[ ]		
Near Field Conditions	Waste Package Description	Make WP File
[ ]		
Waste Package Inventory		
RadioNuclide ID: AM241 1 of 38		
Half-Life (Yrs): 433.125 Daughter Product : NP237		
RN Inventory (Ci/Container) : 14000 (Constant )		
Inventory Distribution:		
Free Fraction : 0	(Constant )	of 1.0
Gap Fraction : 0	(Constant )	of 1
Bound Fraction : 1.0		of 1
Element Solubility (g/m <sup>3</sup> *m): 0 (Constant )		
Effective Diffusion Coefficient (m <sup>2</sup> /yr) : 0 (Constant )		
Is Element Gaseous : No		
If Gaseous, Mass Transfer Rate away from WP (g/yr):0 (Constant )		
Chemical Element Group : 1		
Define Release Pathways		

The actual radionuclides which can be considered are defined externally to the user interface in the ASCII file RN.TBL, which is provided with the RIP source code. An example of this file is shown in Appendix D. This table consists of the radionuclide ID's (e.g., AM241), the first-order decay rate (in years<sup>-1</sup>), the regulatory release limit (per 40 CFR Part 191, App. A), the specific activity (Ci/g), and the primary daughter product (if any). If RN.TBL is edited, both the front end and the back end of RIP *must be recompiled*. The radionuclide decay algorithm implemented in RIP is discussed in the Appendix D of the RIP Theory and Capabilities Manual.



Decay chains are entirely defined by the daughter products which are specified in this table. Note that RIP only allows a single daughter product to be defined for a given radionuclide (branching is not permitted), and that characteristics of that daughter must be defined within the table. Although a radionuclide can not be defined to have multiple daughters, it can have multiple parents.

Radionuclide ID's are restricted to 6 characters. Note that the radionuclide ID *must* be specified as the element abbreviation (e.g., AM, U, PU) followed by the mass number. The reason for this restriction is that solubilities are defined by element, not by isotope. Specified saturation concentrations must therefore be partitioned automatically by RIP among the defined isotopes of the element. Hence, all isotopes of the same element must be similarly named in order for RIP to recognize them as such.

The user should be aware that RIP will only compute saturation concentration based on the isotopes which are specified in RN.TBL. Isotopes which are not included in the table (e.g., non-radioactive isotopes) are not considered in the calculation. Not including a significant non-radioactive isotope results in computed saturation concentrations which are overestimated.

Referring to the input window above, note that RIP indicates how many radionuclides are included in the inventory. This is equal to the number of radionuclides specified in RN.TBL. The user can scroll through the radionuclides with the <F1> and <F2> keys.

Several other function keys are available to facilitate data input. In particular, RIP provides the capability to copy the data pertaining for the current radionuclide to all the other radionuclides. Any differences can then be added manually. The <F3> key simply provides a reminder describing the use of the two copy keys, <F4> and <F5>. <F4> copies all the information, with the exception of the release pathways, for the current radionuclide to all the other radionuclides. <F5> copies just the release pathways for the current radionuclide to all the other radionuclides.

<F6> provides a list of all radionuclides in the inventory.

Note that the Radionuclide ID, the half-life, and the daughter product are display fields only and can not be edited. The user can only add or delete radionuclides from the inventory by editing RN.TBL and recompiling RIP. As will be described below, however, radionuclides can be effectively removed from the inventory by specifying a 0 inventory.

**Radionuclide Inventory:** This is the *initial* radionuclide inventory in Curies per waste package. Note that this should represent the inventory *at the time of closure*, since this is TIME = 0 for RIP. It can be specified as a stochastic or a constant. It must, however, be non-negative, and can not be described as a function of time. Specifying this as zero will only eliminate this radionuclide from the waste package if it is not a daughter product of some other

radionuclide (with a non-zero inventory).

**Free Fraction:** This is the fraction of the inventory which is located in what is referred to in RIP as the "free inventory": that part of the inventory which is instantaneously released upon primary container failure. It can be a constant, a stochastic, or a function, but it must be greater than or equal to zero and less than or equal to one, and can not vary with time.

**Gap Fraction:** This is the *fraction of the remaining inventory* (displayed immediately to the right) which is located in what is referred to in RIP as the "gap inventory": that part of the inventory which is instantaneously released upon secondary container failure in a waste package in which the primary container has also failed. It can be a constant, a stochastic, or a function, but it must be greater than or equal to zero and less than or equal to one, and can not vary with time. Note that the bound fraction cannot be edited, as it must be equal to one (i.e., anything not in the free or the gap inventory must be in the bound inventory). The free, gap and bound inventories are discussed in detail in the RIP Theory and Capabilities Manual, Section 3.2.3.

The following five input fields are *elemental properties* as opposed to radionuclide properties. Hence, for any given element, all isotopes of the element are forced to have identical values for these five input fields. For example, if a user enters a solubility for AM241, it is automatically copied by the user interface to all other isotopes of the same element.

**Element Solubility:** This is the elemental solubility in  $\text{g/m}^3$ . This can be a constant, a stochastic, or a function, but must be non-negative.

**Effective Diffusion Coefficient:** This is the effective diffusion coefficient (from the waste form to the edge of the host rock) in  $\text{m}^2/\text{year}$ . It should include the effects of tortuosity on the molecular diffusion coefficient. It can be a constant, a stochastic or a function, but must be non-negative. The effective diffusion coefficient is discussed in detail in Section 3.2.4.1 of the RIP Theory and Capabilities Manual.

**Is Element Gaseous:** This is an option field which identifies those elements which can be transported out of the waste package in the gaseous state (e.g.,  $^{14}\text{C}$ ). It must be answered Yes or No. Answering Yes to this field does not imply that the radionuclide travels in the far-field in a gaseous transport pathway. It only implies that it can leave the waste package as a gas.

**If Gaseous, Mass Transfer Rate away from WP:** This field is only active if the user specifies Yes in the previous field. It is the gaseous mass transfer rate away from a single waste package in  $\text{g/year}$ . Specifying this input as a large number (e.g.,  $10^{10}$ ) effectively ensures that the release is always controlled by the rate of exposure (as opposed to the rate of mass transfer). It can be a constant, a stochastic, or a function, but must be non-negative.

**Chemical Element Group:** This specifies the "retardation group" to which this particular element belongs. As will be described in detail in Section 5.2, RIP currently allows the user to divide the elements in the inventory into as many as six retardation groups, each of which is defined by different retardation factors. All elements within a particular group have the same retardation factor. The value of the retardation factor for a particular group, however, is a function of the pathway in which the elements are being transported.

This is an option field which allows the user to toggle between the numbers 1, 2, 3, 4, 5 and 6 which identify the chemical element group. The user should define the retardation groups within the [Pathways] input windows before specifying this value.

### 5.1.2.6 Linking the Waste Package Release to the Far-Field Transport Pathways

**Define Release Pathways:** This provides access to the following window for specifying the far-field transport pathways to which mass for this particular radionuclide in this particular waste package type is distributed upon release from the waste package:

[ ] Repository Integration Program (Rev. 1.00 )			
General		PA Model	Strategy Model
User's Guide Tutorial			
[ ]			
Waste Package		Pathways	Events
Make Transfer File			
[ ]			
Near Field Conditions		Waste Package Description	Make WP File
[ ]			
RadioNuclide ID: AM241		1 of 38	F1=Previous
Half-Life			ext
RN Inve		Discharge Pathways	elp Copy
Pathway ID : unsat		Path 1 of 3	opy1
Balance Remaining : 1			opy2
Invento		Fraction of balance to path: 0.2 (Constant )	ist RNs
Free		F1=Prev F2=Next F3=Add F4=Del	
Gap Fraction : 0		(Constant ) of 1	
Bound Fraction : 1.0		of 1	
Element Solubility (g/m <sup>3</sup> *m): 0 (Constant )			
Effective Diffusion Coefficient (m <sup>2</sup> /yr): 0 (Constant )			
Is Element Gaseous : No			
If Gaseous, Mass Transfer Rate away from WP (g/yr):0 (Constant )			
Chemical Element Group : 1			
Define Release Pathways			

Released mass can be partitioned to as many pathways as desired. However, it can only be partitioned to pathways which have been previously defined. Hence, the user must define the pathways within the [Pathways] input windows before specifying the release pathways here. Discharge pathways can be added or deleted with the <F3> and <F4> keys, respectively. The <F1> and <F2> keys are used to move between defined discharge pathways.

For each discharge pathway, the user specifies the "Pathway ID", which must correspond to that of a previously defined pathway, and the "fraction of balance to path", which represents the fraction of the remaining balance which is to be allocated to this pathway. The remaining balance is indicated in a display field directly above.

As an example, suppose the user wished to define three discharge pathways for a particular radionuclide. If the first was specified as having 40% of the released mass, the window for the second discharge pathway would look like this:

[ ] Repository Integration Program (Rev. 1.00 )			
General		PA Model	Strategy Model
User's Guide Tutorial			
[ ]			
Waste Package		Pathways	Events
Make Transfer File			
[ ]			
Near Field Conditions		Waste Package Description	Make WP File
[ ]			
RadioNuclide ID: AM241		1 of 38	
Half-Life: 12.5		F1=Previous	
RN Inve		ext	
Pathway ID : unsat2		Path 2 of 3	
Balance Remaining : 0.6		elp Copy	
Fraction of balance to path: 1		opy1	
(Constant )		opy2	
F1=Prev F2=Next F3=Add F4=Del		ist RNs	
Gap Fraction : 0		(Constant ) of 1	
Bound Fraction : 1.0		of 1	
Element Solubility (g/m <sup>3</sup> *m <sup>3</sup> m) : 0 (Constant )			
Effective Diffusion Coefficient (m <sup>2</sup> /yr) : 0 (Constant )			
Is Element Gaseous : No			
If Gaseous, Mass Transfer Rate away from WP (g/yr):0 (Constant )			
Chemical Element Group : 1			
Define Release Pathways			

RIP would indicate that 60% of the mass released remains to be partitioned. Specifying 0.5 in this field means that 50% of the remaining balance (or 30% of the total) is partitioned to the second discharge pathway. The remaining 30% of the mass would be allocated to the third discharge pathway. Specifying 1 in this field for the second discharge pathway means that all of the remaining mass (60% of the total) would be partitioned to this pathway.

For the last discharge pathway defined, the value automatically takes on the value of 1. For other pathways, however, this field can be specified as a parameter (a constant, a stochastic, or a function), with the constraint that it must *always* be between greater than or equal to 0 and less than or equal to 1.

Entering the *fraction of remaining balance* (as opposed to the *actual fraction*) may appear somewhat awkward. However, this convention allows the fractions to be represented stochastically, while still ensuring that they total to one. This convention is used in several other places within the RIP user interface and its application is discussed further in Appendix B.

As will be discussed in Section 5.3, RIP reserves a special pathway name called AE. This refers to the accessible environment. This is the ultimate pathway to which mass can be discharged. It is not explicitly defined by the user, but can be specified as a discharge pathway.

### 5.1.3 Making a Waste Package Transfer File

RIP allows the user an option to simulate waste package behavior and radionuclide release independent from the radionuclide transport pathways. That is, if desired, the user can model just the *release from the waste packages* without considering the subsequent transport of

the radionuclides to the accessible environment. To do this, the user creates a waste package transfer file which contains only information pertaining to the waste package behavior and instructs the back end to only simulate radionuclide release from the waste packages.

If this option is selected, RIP essentially bypasses the pathways component model. All other aspects of the model, including any specified disruptive events, are still simulated.

Selecting [Make WP Transfer File] from the menu under [Waste Package] provides access to the input windows for specifying general information regarding the creation of a waste package transfer file. Having done so, the user can then create the transfer file from this window.

The input window is as follows:

[ ] Repository Integration Program (Rev. 0.01 )			
General	PA Model	Strategy Model	Results
[ ] verification test file			
[ ] Waste Package Pathways Events Make Transfer File			
[ ]			
Near Field Conditions	Waste Package Description	Make WP File	
[ ] WASTE PACKAGE MODEL			
Number of timesteps		: 100	
Length of timestep		: 10	
Number of realizations		: 2	
Run using expected values		: No	
Select Desired Output			
Define Realization Classes			
Make Waste Package Transfer File			

**Number of timesteps:** This is the number of timesteps to be simulated. It must be greater than or equal to one. The beginning of the simulation (time = 0) is defined as the time of closure.

**Length of timestep:** This is the timestep length in years. It must be non-negative.

**Number of realizations:** This is the number of Monte Carlo realizations to be simulated. It must be greater than or equal to one.

**Run using expected values:** This is an option field which is answered Yes or No. If Yes, a single realization using the expected value for all parameters is produced.

**Select Desired Output:** This provides access to an input window for defining the form of the output:

Repository Integration Program (Rev. 0.01)

Ge [ ] SELECT DESIRED OUTPUT

[ ] Waste

[ ] Near

Waste Pkg. Rel. rate  
10 CFR 60.113 : Yes

Cumulative Radionuclide Release	Sum	by RN
From all Waste Packages	No	No
From each Waste Package Group	No	No

Maximum Release Time Histories

No	No
----	----

Normalize Results ? Yes

file

: 100  
: 100  
: 10000  
: No

The user has a choice of the exact outputs which are to be saved in the results file. Choosing all of the available outputs could create very large output files. Therefore, before discussing each of the outputs in detail, we introduce the following nomenclature in order to allow the user to estimate the approximate size of the output file which will be generated:

NREAL - number of Monte Carlo realizations  
 NRN - number of radionuclides  
 NGRP - number of waste package groups  
 NPTH - number of transport pathways  
 NBNUM - number of bytes per number  
 NTIME - number of timesteps

RIP currently outputs numbers that are represented by 12 characters (bytes).

**Waste Pkg. Rel. rate (10 CFR 60.113):** This saves the maximum annual radionuclide release rate (normalized to the 10 CFR 60.113 limit) achieved by any radionuclide over the length of the realization. Normalization of results is discussed in detail in Appendix C. For each realization, a single value and the ID of the radionuclide having the maximum annual release are saved. This results in  $(\text{NBNUM} + 5) \times \text{NREAL}$  bytes being written to the output file (where 5 represents the number of bytes for the radionuclide ID).

The results under "Cumulative Radionuclide Release" refer to results that are summed over all timesteps. That is, they are cumulative with respect to time. "Sum" indicates that results are summed over all radionuclides. "by RN" indicates that a separate cumulative release is to be saved for each radionuclide.

**Normalize Results:** This determines whether or not the two cumulative radionuclide release results described below are to be normalized. It does not pertain to the "Waste Package Release Rate" described above. If the user selects Yes, the cumulative releases below are normalized to the 40 CFR 191.13 limit

(normalization is discussed in detail in Appendix C). If the user selects No, results are output in units of Curies.

**From all Waste Packages:** This saves the cumulative waste package release summed over all waste package groups. If "Sum" is selected, for each realization, the cumulative waste package release (summed over all radionuclides) is saved. This results in NBNUM x NREAL bytes being written to the output file. If "by RN" is selected, for each realization, the cumulative waste package release for each individual radionuclide is saved. This results in NBNUM x NREAL x NRN bytes being written to the output file.

**From each Waste Package Group:** This saves the cumulative waste package release for each waste package group. If "Sum" is selected, for each realization, the cumulative waste package release (summed over all radionuclides) is saved for each waste package group. This results in NBNUM x NREAL x NGRP bytes being written to the output file. If "by RN" is selected, for each realization, the cumulative waste package release for each individual radionuclide for each waste package group is saved. This results in NBNUM x NREAL x NRN x NGRP bytes being written to the output file.

**Maximum Release Time Histories:** This saves the time history of waste package release (in Curies) for the three realizations having the highest total cumulative waste package release. Time histories are saved only for those categories for which cumulative results are being saved in the column directly above. For example, consider the following screen:

[ ] Repository Integration Program (Rev. 0.01 )

Ge [ ] SELECT DESIRED OUTPUT s

[ ] Waste  
[ ] Near

Waste Pkg. Rel. rate 10 CFR 60.113 : Yes	
Cumulative Radionuclide Release	Sum by RN
From all Waste Packages	Yes No
From each Waste Package Group	No Yes
Maximum Release Time Histories	Yes Yes

file

: 100  
: 100  
: 10000  
: No

Normalize Results ? Yes

In this case, a total of  $3 + 3 \times \text{NRN} \times \text{NGRP}$  time histories would be saved: 3 time histories of total waste package release; and 3 time histories of waste package release for each individual radionuclide in each waste package group. Each time history would result in NBNUM x NTIME bytes being written to the output file.

**Define Realization Classes:** This provides access to an input window for defining *importance sampling of disruptive events*. Importance sampling is discussed in detail in Appendix C of the RIP Theory and Capabilities Manual. Importance sampling of events allows the user to emphasize realizations in which relatively low probability events occur. This increases the efficiency of the Monte Carlo sampling technique and improves the accuracy of the low probability tails of results distributions.

The user implements the event importance sampling technique by dividing all possible realizations into a number of *realization classes*, each realization class having a specified *pruning factor*. Realization classes are defined in terms of disruptive events. For example, suppose that the user has defined two types of disruptive events, referred to here as HUMAN and VOLCAN (specification of disruptive event types is discussed in Section 5.3). The user might then define one realization class to be "all realizations in which event VOLCAN occurs". By default, one class always exists which is called "all other realizations". In this case, this would include realizations in which either no disruptive events occurred or event HUMAN occurred.

Each disruptive event class is assigned a pruning factor (which must be greater than or equal to one). The pruning factor determines what fraction of the realizations in a given class are actually simulated. For example, if a class was assigned a pruning factor of 100, only one in every 100 realizations would actually be simulated, and the rest would be ignored. For a pruning factor of 1, every realization is simulated. To understand how this will affect the efficiency of the Monte Carlo sampling, refer again to the example above. Suppose that event VOLCAN was relatively rare (e.g., with a rate of occurrence of  $10^{-7}$  per year). This would mean that for a 10,000 year simulation, there is only a  $10^{-3}$  probability of realizing such an event. Hence, in order to properly represent this event in a Monte Carlo simulation, more than  $10^3$  realizations would have to be simulated. If, however, we were to specify a pruning factor of 1 for the class which consists of realizations with event VOLCAN, and a pruning factor of 100 for all other realizations, we would increase the frequency at which realizations in which VOLCAN occurred were simulated (by a factor of 100). In effect, if we had actually simulated  $10^4$  random realizations, it would seem that with respect VOLCAN type events we had simulated  $10^6$  random realizations. On the other hand, with respect to HUMAN realizations and all realizations in which no event had occurred, it would seem as if we had simulated somewhat less than  $10^4$  realizations (since we have artificially increased the rate at which we sampled VOLCAN realizations).

The user should consult Appendix C of the RIP Theory and Capabilities Manual for a more complete description of the event importance sampling algorithm.

Realization classes are defined in the following window:



Repository Integration Program (Rev. 0.01)				
General	PA Model	Strategy Model	Results	
User's Guide Tutorial				
Waste Package	Pathways	Events	Make Transfer File	
Near Field Conditions	Waste Package Description	Make WP File		
[ ] WASTE PACKAGE MODEL				
REALIZATION CLASSES				
Class 1 of 2				
Events in Class				
VOLCAN				
Pruning Factor : 1				
F1=Prev F2=Next F3=Add F4=Del F5=Display Events				

By default, a single realization class is originally defined, and this realization class is called "all other realizations". Its default pruning factor is 1. Additional classes can be added using the <F3> key. Defined classes can be browsed or deleted using the <F1>, <F2>, and <F3> keys. Note that regardless of how many classes are defined, the last class is always "all other realizations", and this name can not be changed. The pruning factor for this default class, however, can be edited.

After defining a new class, the user must specify which events the class contains. This is done by typing in the name of the event (or events). Multiple events must be separated by commas (e.g., "VOLCAN,HUMAN"). *Only events which have been previously specified within the disruptive events input window (see Section 5.3) can be specified.* As a reminder, the <F5> key provides a list of all defined events for the user.

**Make Waste Package Transfer File:** Selecting this field creates a waste package transfer file. If any disruptive events have been defined under [Events], these will also be simulated, but any input pertaining to transport pathways is ignored. That is, the output is only in terms of release from the waste package.

Prior to creating the file, RIP goes through a series of data checks to ensure a consistent set of input. If any input errors are detected, creation of the file will be halted and an appropriate error message will appear alerting the user to the input error. If no input errors are detected, creation of the transfer file will commence. The progress of the transfer file creation is tracked on the screen by indicating which realization is currently being written.

The transfer file that is created is called NAME.TNF, where NAME is the name of the last input file specified by the user.

**WARNING:** Because creation of the transfer file could potentially crash the program if severe input errors are present, the user should use <ALT>-<S> to ensure that the current data set is saved prior to creation of the file.

## 5.2 Input for the Radionuclide Transport Pathways Component Model

Selecting [Pathways] from the menu under [PA Model] provides access to input windows for describing the radionuclide transport pathways component model.

Prior to entering data into RIP, the user must first have a good understanding of manner in which RIP simulates radionuclide transport through pathways. Chapter 4 of the RIP Theory and Capabilities Manual describes the theory and discusses the assumptions on which the transport algorithms are based.

In simple terms, the transport algorithm implemented in RIP propagates mass along a network of interconnected *pathways*. The pathways reflect major features of the hydrologic system and are conduits through which mass transport occurs. In a RIP application, the user defines a pathway network consistent with what is known about the hydrologic system. The pathways may be used both for flow balance and radionuclide transport purposes. They may account for either gas or liquid phase transport through the far-field.

RIP pathways are further divided into interacting *flow modes*, which address heterogeneity at the local scale within a pathway (e.g., matrix flow and fracture flow might be two defined flow modes). The flow modes are primarily distinguished from one another based on flow velocity.

After carefully reading Chapter 4 of the RIP Theory and Capability Manual and achieving a clear understanding of the manner in which transport pathways are simulated in RIP, the user should next sketch the conceptual model for the system which is to be represented, such that the various pathways and flow modes (and their associated parameters) are clearly defined. Only after sketching a conceptual model for the pathway network can the pathway data can be entered.

Selecting [Pathways] from the main menu provides access to the following input window:

[ ] Repository Integration Program (Rev. 0.01 )			
General	<u>PA Model</u>	Strategy Model	Results
[ ] User's Guide Tutorial			
Waste Package	<u>Pathways</u>	Events	Make Transfer File
[ ] Pathway Database Editor			
ID : PATH1	Path 1 of 1		F1=Prev
DESC: unsaturated zone pathway 1			F2=Next
Pathway Length (m): 100	(Constant )		F3=Add
Total Area of Pathway (m <sup>2</sup> ): P1AREA	(1000000 )		F4=Del
Total Volumetric Flow (m <sup>3</sup> *m/yr): P1FLOW	(0.01 )		
Define Discharge Pathways			
Define Flow Modes			
Define Pathway Tolerance Values			

The user can add pathways with the <F3> key. Pathways can be deleted with the <F4> key. The <F1> and <F2> keys can be used to move between defined pathways.

The specific input fields in the pathway database editor window shown above are discussed in the following sections.

### 5.2.1 Pathway Definition

**ID:** This is a six character identifier for the pathway. Note that RIP reserves a special pathway name called AE which can not be defined by the user. This refers to the *accessible environment*, and is effectively the ultimate pathway to which mass can be discharge. It is not explicitly defined by the user, but, as will be seen below, can be specified as a discharge path.

**DESC:** This is a 60 character description for the pathway.

**Pathway Length:** This is the actual length of the pathway (in m). The pathway length can be specified as a constant, a stochastic, or a function. However, it must be specified as non-negative. Note that this parameter may often be described directly as a time variable function (or it could be affected by disruptive events, as will be discussed in Section 5.3) such that the pathway length would actually change with time. (For example, a water table rise would have the effect of decreasing the length of an unsaturated zone pathways.) The manner in which RIP simulates such a change is detailed in Chapter 4 of the RIP Theory and Capabilities Manual.

**Total Area of Pathway:** This is the cross-sectional area of the pathway normal to the flow direction (in  $m^2$ ). It can be specified as a constant, a stochastic, or a function. However, it must be specified as non-negative.

**Total Volumetric Flow:** This is the total volumetric flow rate through the pathway (in  $m^3/yr$ ). It can be specified as a constant, a stochastic, or a function. However, it must be specified as non-negative. Note, that as discussed in Section 4.4.2 of the RIP Theory and Capabilities Manual, RIP does not explicitly compute a water balance for the pathway network (although it does explicitly balance radionuclide mass). Hence, it is up to the user to define the water balance, if desired, through the specification of the pathway volumetric flow rates. These rates should be specified by the user such that they are interdependent and consistent (i.e., the flow in a particular pathway should be a function of the flows in the paths which discharge into it). The echo file created from with the [GENERAL] input window contains a concise summary of the specified flow balance.

### 5.2.2 Pathway Network Definition

The pathway network is defined by identifying for each pathway one or more discharge pathways to which the mass is released.

**Define Discharge Pathways:** This provides access to the following window for specifying the pathways to which mass from this particular pathway is distributed upon discharge:

Repository Integration Program (Rev. 0.01)		Results
General	PA Model	Strategy Model
User's Guide Tutorial		
Waste Package Pathways Events Make Transfer File		
Pathway Database Editor		
ID : PATH1	Path 1 of 1	F1=Prev F2=Next F3=Add F4=Del
DESC: unsaturated zone pathway 1		
Pathway Length	(m): 100	(Constant )
Total A [ ]	Discharge Pathways	
Total V	Pathway ID : PATH2	Path 1 of 1
Define	Balance Remaining : 1	
Define	Fraction of balance to path: 1	(Constant )
Define	F1=Prev F2=Next F3=Add F4=Del	

Released mass can be partitioned to as many pathways as desired. However, it can only be partitioned to pathways which have been previously defined. Hence, the user must define the pathways within the main [Pathways] input window before specifying the release pathways here. Selected discharge pathways can be added or deleted with the <F3> and <F4> keys, respectively. The <F1> and <F2> keys are used to move between defined discharge pathways.

For each discharge pathway, the user specifies the "Pathway ID", which must correspond to that of a previously defined pathway, and the "fraction of balance to path", which represents the fraction of the remaining balance which is to be allocated to this pathway. The remaining balance is indicated in a display field directly above.

For the last discharge pathway defined, the value automatically takes on the value of 1. For other pathways, however, this field can be specified as a parameter (a constant, a stochastic, or a function), with the constraint that it must *always* be between greater than or equal to 0 and less than or equal to 1.

Entering the *fraction of remaining balance* (as opposed to the *actual fraction*) may appear somewhat awkward. However, this convention allows the fractions to be represented stochastically, while still ensuring that they total to one. This convention is used in several other places within the RIP user interface and its application is discussed further in Appendix B.

As mentioned above, RIP reserves a special pathway name called AE. This refers to the accessible environment. This is the ultimate pathway to which mass can be discharged. It is not explicitly defined by the user, but can be specified as a discharge pathway.

### 5.2.3 Flow Mode Definition

For each pathway, it is necessary to specify one or more flow modes. As discussed in Chapter 4 of the RIP Theory and Capabilities Manual, the flow modes represent small-scale heterogeneities in the pathways.

**Define Flow Modes:** This provides access to the following window for specifying the characteristics of the pathway flow modes:

Repository Integration Program (Rev. 0.01)			
General	PA Model	Strategy Model	Results
User's Guide Tutorial			
Waste Package Pathways Events Make Transfer File			
Pathway Database Editor			
ID : PATH1	Path 1 of 1		F1=Prev F2=Next F3=Add F4=Del
DESC: unsaturated zone pathway 1			
Pathway Length (m): 100	(Constant )		
Total Area of Pathway (m <sup>2</sup> ): P1AREA	(1000000 )		
Total Volumetric Flow (m <sup>3</sup> m/yr): P1FLOW	(0.01 )		
Define Discharge Pathways			
Define Flow Modes			
Flow Modes		Mode 1 of 2	F1=Prev F2=Next F3=Add F4=Del
DESC: matrix flow			
Fraction of Total Flow remaining : 1			
Fraction of remaining Flow (0<x<1): FMATRX	(0.9 )		
Dispersivity (m): 0	(Constant )		
Poisson Transition Rate (m): MRATE	(1.2 )		
Choose data type to input : Velocity			
Enter data value : MVEL	(0.001 )		
Define Retardation Factors			

The user can add flow modes with the <F3> key. Flow modes can be deleted with the <F4> key. The <F1> and <F2> keys can be used to move between defined flow modes.

The specific input fields associated with a particular flow mode are discussed below.

**Fraction of Remaining Flow:** For each flow mode, the user specifies the fraction of the remaining balance which is to be allocated to this particular flow mode. The remaining balance is indicated in a display field directly above.

For the last flow mode defined, the value automatically takes on the value of 1. For other modes, however, this field can be specified as a parameter (a constant, a stochastic, or a function), with the constraint that it must *always* be between greater than or equal to 0 and less than or equal to 1.

Entering the *fraction of remaining balance* (as opposed to the *actual fraction*) may appear somewhat awkward. However, this convention allows the fractions to be represented stochastically, while still ensuring that they total to one. This convention is used in several other places within the RIP user interface and its application is discussed further in Appendix B.

**Dispersivity:** This is the dispersivity (in m) for the one-dimensional advection-dispersion equation which is used to compute the breakthrough curve for the pathway. Note, however, that this field is relevant only when a single flow mode is defined for the pathway. If more than one flow modes are defined,

this field is ignored and the breakthrough curve is calculated based on a Markov process algorithm. If only one mode is defined, however, and the dispersivity is non-zero, the breakthrough curve is computed based on a one-dimensional solution for a tracer slug in an infinite domain with uniform flow (Bear, 1979, p. 266). If a single flow mode is defined and the dispersivity is zero, the Markov process algorithm is used (in this case it would result in a plug-flow representation). The dispersivity can be specified as a constant, a stochastic, or a function. However, it must be specified as non-negative.

**Poisson Transition Rate:** This is the Poisson transition rate described in Section 4.3.2 of the RIP Theory and Capability Manual. The transition rate has units of inverse length ( $m^{-1}$ ). In simple terms, it represents the inverse of the length that we would expect a particle to travel in a particular flow mode before it transitions into a different flow mode within the pathway. The transition rate must be defined for one and only one flow mode within a pathway. (Whenever a flow mode's transition rate is specified, all other flow modes' rates are automatically set to zero in the user interface). The other transition rates are computed internally based on the one specified transition rate and the fraction of flow in each mode. The assumptions on which this calculation is based is detailed in Appendix G of the RIP Theory and Capabilities Manual. The transition rate which is defined by the user can be a constant, a stochastic, or a function, but it must be non-negative.

**Choose data type to input:** For each flow mode, the user can choose to input either a velocity or an effective porosity to describe flow in the mode. The user uses the <PgUp> or <PgDn> keys to toggle between these two choices. The velocity and porosity are related as follows:

$$v_i = \frac{f_{Qi} Q}{An_i}$$

where:

$v_i$  = the average linear fluid velocity for flow mode  $i$  [m/yr];

$f_{Qi}$  = the fraction of the pathway volumetric flow in flow mode  $i$ ;

$Q$  = the total pathway volumetric flow [ $m^3$ /yr];

$A$  = the total pathway area normal to the flow direction [ $m^2$ ]; and

$n_i$  = the effective porosity in flow mode  $i$ .

**Enter value data:** This is either the average linear flow velocity (in m/yr) or the effective porosity (unitless) for the flow mode, depending on the specification in the previous field. The value can be a constant, a stochastic, or a function. If specified as a velocity, it must be greater than zero. If specified as a porosity,

it must be greater than zero and less than or equal to one.

**Define Retardation Factors:** This provides access to the following window for specifying retardation factors:

Repository Integration Program (Rev. 1.00)			
General		PA Model	Results
User's Guide Tutorial			
Waste Package Pathways Events Make Transfer File			
Pathway Database Editor			
ID : PATH1	Path 1 of 1		F1=Prev
DESC: unsaturated zone pathway 1			F2=Next
Pathway Length (m): 100	(Constant )		F3=Add
Total Area [ ]	Retardation Factors		F4=Del
Total Volu	Group 1 of 6		
Define Dis	Sorption : 10	(Constant )	
Define Flo	Matrix Diffusion : 20	(Constant )	
F1=Prev F2=Next			
DESC: matrix flow		Mode 1 of 2	F1=Prev
Fraction of Total Flow remaining : 1			F2=Next
Fraction of remaining Flow (0<x<1): FMATRX	(0.9 )		F3=Add
Dispersivity (m): 0	(Constant )		F4=Del
Poisson Transition Rate (m): MRATE	(1.2 )		
Choose data type to input : Velocity			
Enter data value : MVEL	(0.001 )		
Define Retardation Factors			

As discussed in Section 5.1.2.5, every element in the inventory of the specified waste packages is assigned to a *chemical element group*. Each chemical element group is referenced by an integer. Currently, RIP allows the user to specify up to six chemical element groups. In this window the user specifies retardation factors for each chemical element group. Note that, as a result, the retardation factors for any chemical element group can vary as a function of pathway and flow mode.

Two retardation factors must be specified: **Sorption** describes retardation due to chemical or physical sorption processes. **Matrix Diffusion** describes retardation due to matrix diffusion. The two retardation factors are combined as described in Appendix J of the RIP Theory and Capability Manual. Both can be specified as constants, stochastics, or functions. They must, however, be greater than or equal to one.

#### 5.2.4 Definition of Pathway Tolerance Values

The last field in the pathway database editor window allows the user to specify pathway tolerance values:

**Define Pathway Tolerance Values:** This field provides access to the following window for specifying pathway tolerance values:

[ ] Repository Integration Program (Rev. 1.00)		Results	
General		PA Model	Strategy Model
User's Guide Tutorial			
[ ]			
Waste Package		Pathways	Events
		Make Transfer File	
[ ] Pathway Database Editor			
ID : PATH1	Path 1 of 1		F1=Prev
DESC: unsaturated zone pathway 1			F2=Next
Pathway Length	[ ] PATHWAY TOLERANCE VALUES		F3=Add
Total Area of Pa	Pathway Length : 1	)	F4=Del
Total Volumetric	Flow Mode Velocity : 1	)	
Define Discharge	Flow Mode Porosity : 1	)	
Define Flow Mode	Fraction of Total Flow : 1	)	
Define Pathway T	Transition Rate : 1	)	

The use of tolerance values is discussed in Section 4.3.3 of the RIP Theory and Capabilities Manual. Tolerance values are simply a mechanism to increase the efficiency of the computational algorithm. Because transient changes in the pathway parameter values may cause additional calculations for be performed in the RIP model (due to various defined dependencies), minimum (threshold) changes are required before the change is actually implemented in the simulation. The minimum required change is defined by the user in terms of the fractional difference between a new value and an old value. This fractional difference is computed as follows:

$$\Delta p = \text{absolute value of } \left( \frac{p^{\text{new}}}{p^{\text{old}}} - 1 \right)$$

where:

$\Delta p$  = the fractional difference for parameter  $p$ ;

$p^{\text{new}}$  = the new value for parameter  $p$ ; and

$p^{\text{old}}$  = the old value for parameter  $p$ .

The fractional difference,  $\Delta p$ , is compared to the user specified tolerance value to determine if the change should be implemented in the solution. If  $\Delta p$  does not exceed the tolerance value,  $p^{\text{old}}$  is not updated to  $p^{\text{new}}$  (the change is ignored).

As can be seen in the previous window, there are five tolerance values which must be specified. These must be specified as non-negative constants. Specification of 0 tolerance values ensures that all changes are implemented. The user should carefully define these values, considering the importance of different changes in the parameters if computational efficiency is of concern.

### 5.3 Input for the Disruptive Events Component Model

Selecting [Events] from the menu under [PA Model] provides access to input windows for describing the disruptive events component model.



Prior to entering data into RIP, the user must first have a good understanding of the manner in which RIP simulates the occurrence and consequences of disruptive events. Chapter 5 of the RIP Theory and Capabilities Manual describes the theory and discusses the assumptions on which the RIP disruptive events algorithms are based.

*Disruptive events* are defined here as discrete perturbations of the repository system. That is, disruptive events are discrete occurrences which have some quantifiable effect on the processes described by the other two component models. Note that discrete is a relative term, and does not necessarily imply instantaneous (e.g., given the long time scales of interest, something taking place over a period of 100 years could be considered as a discrete event). Examples of disruptive events under this definition include volcanism, faulting, and human intrusion. In general, however, the disruptive events component model is intended to represent relatively rare occurrences. Events that occur continuously throughout the time period of interest are more efficiently modeled as processes within the other two component models.

The occurrence of disruptive events is described as follows:

- The user defines all significant events (i.e., all events which are both credible and consequential).
- Each significant event must be assigned a *rate of occurrence*. Event occurrences are simulated as Poisson processes.
- Each disruptive event can be assigned one or more *descriptor parameters*. Descriptor parameters define the characteristics and/or the magnitude of the event. For example, the list of descriptor parameters for a volcanic event might include dike length and dike width.

For each disruptive event, there are four types of internally-defined disruptive event *consequences* which describe *discrete responses*:

- The radionuclides in a number of waste packages are moved directly to the accessible environment.
- The radionuclides in a number of waste packages are moved directly to a specified pathway (e.g., the saturated zone).
- A number of waste packages are disrupted in place.
- A portion of the mass (previously released from waste packages) contained within a pathway is immediately discharged to the accessible environment.

In addition to the four discrete response consequences described above, it is also possible to directly influence parameters defined in the waste package and transport pathways components, and this capability can be used to specify *long-term consequences* (e.g., a permanent water table rise).

Selecting [Events] from the main menu provides access to the following input window:

[ ] Repository Integration Program (Rev. 1.00 )	
General	PA Model Strategy Model Results
[ ] User's Guide Tutorial	
Waste Package	Pathways Events Make Transfer File
[ ] Disruptive Event Window	
Parameter ID : VOLCAN	(must be "Event" type) Event 1 of 1
Description : volcanic disruption of repository	
Annual Rate of Occurrence : 1.000E-6 (Constant )	
Can Event Reoccur : Yes	
Define Event Descriptors	
Event Consequences	
Move WPs to AE	
Move WPs to a pathway	
Disrupt WPs	
Move Contents of pathway to AE	
Modify Parameters	
F1=Prev F2=Next F3=Add F4=Del	

The user can add disruptive events with the <F3> key. Events can be deleted with the <F4> key. The <F1> and <F2> keys can be used to move between defined events.

The specific input fields in the disruptive event window shown above are discussed in the following sections.

### 5.3.1 Event Definition

**Parameter ID:** This is the name of the disruptive event parameter. This must correspond to an event type parameter in the parameter module data base. (Recall that the user can enter the parameter module to define such a parameter by hitting <Alt>-<P>.) As described in Section 4.3.4, definition of an event parameter requires a simple description, a rate of occurrence, and specification of whether the event can reoccur.

Note that the fields in the disruptive events window directly below "Parameter ID" ("Description", "Annual Rate of Occurrence", and "Can Event Reoccur") can not be edited here. They are simply displayed based on the information specified when the event parameter was defined within the parameters module.

Recall that the value of the event parameter is automatically computed by RIP as a function of time, and for any given timestep is equal to 1 if the events has occurred that timestep and 0 if it has not.

### 5.3.2 Definition of Event Descriptors

**Define Event Descriptors:** This field provides access to the following window for defining event descriptors:

Repository Integration Program (Rev. 1.00)			
General	PA Model	Strategy Model	Results
User's Guide Tutorial			
[ ] Waste Package Pathways <u>Events</u> Make Transfer File			
[ ] Disruptive Event Window			
Parameter ID : VOLCAN (must be "Event" type)		Event 1 of 1	
Description : volcanic disruption of repository			
[ ] Event Descriptors			
Parameter ID : DIKE (2000 )		Descriptor 1 of 1	
Description : dike length			
F1=Prev F2=Next F3=Add F4=Del			
Event Consequences			
Move WPs to AE			
Move WPs to a pathway			
Disrupt WPs			
Move Contents of pathway to AE			
Modify Parameters			
		F1=Prev F2=Next F3=Add F4=Del	

The user can add descriptor parameters with the <F3> key. Descriptor parameters can be deleted with the <F4> key. The <F1> and <F2> keys can be used to move between defined descriptors.

Descriptor parameters can be any user-defined parameter, and normally are a stochastic type used to define an uncertain aspect of the event. The user must define these parameters, of course, using the parameters module. As discussed in Chapter 5 of the RIP Theory and Capability Manual, descriptor parameters allow the user to define the characteristics and/or the magnitude of a disruptive event. It is intended that descriptor parameters be used to help define the magnitude of the consequences. The magnitude of a consequence can then be described as a function of the characteristics of that particular event. For example, for a volcanic intrusion event, we might want to describe the number of waste packages which are disrupted as a function of the geometry and size of the dike. Descriptor parameters would then be defined which describe the dike size and geometry. *Descriptor parameters are realized every time the particular event occurs.*

Descriptor parameters should only be used to define event consequences. That is, they should not be referenced elsewhere within the model (in cases where they are, they simply retain a single value for the entire realization).

With the event descriptors window, the user specifies the parameter ID and a short description.

### 5.3.3 Definition of Event Consequences

As shown in the main disruptive event window, there are five types of event consequences which can be defined. These each have a separate input window. The first four describe discrete event occurrences. The last window allows the user to directly modify other system parameters.

Note that consequences (and any parameters on which they depend) are realized every time a particular event occurs within a realization. If the event has not occurred, all consequences are automatically set to zero. The default value for all consequences is zero.

**Move WPs to AE:** This provides access to the following input window:

[ ] Repository Integration Program (Rev. 1.00)	
General	PA Model
Strategy Model	
User's Guide Tutorial	
[ ] Waste Package Pathways Events Make Transfer File	
[ ] Disruptive Event Window	
Parameter ID : VOLCAN	(must be "Event" type) Event 1 of 1
Description : volcanic disruption of repository	
Annual Rat [ ]	Move WPs to AE
Can Event	Number of WPs moved : WPMOV (2)
Define Event Descriptors	
Event Consequences	
Move WPs to AE	
Move WPs to a pathway	
Disrupt WPs	
Move Contents of pathway to AE	
Modify Parameters	
F1=Prev F2=Next F3=Add F4=Del	

This moves a specified number of waste packages directly to the accessible environment. The waste packages which are moved are chosen randomly from each specified waste package group. It is assumed that the inventory released from each package which is moved is equal to the inventory of an *unfailed* package at the time of the event. Note that the total number of waste packages remaining in the repository is not updated (reduced) subsequent to an event (it is assumed that the number of waste packages which are moved is a small fraction of the total).

The user can specify a constant, a stochastic, or a function (e.g., of descriptor parameters). The value must, however, be greater than or equal to 0.

**Move WPs to a pathway:** This provides access to the following input window:

Repository Integration Program (Rev. 1.00)			
General	<u>PA Model</u>	Strategy Model	Results
User's Guide Tutorial			
[ ] Waste Package Pathways Events Make Transfer File			
[ ] Disruptive Event Window			
Parameter ID : VOLCAN (must be "Event" type)		Event 1 of 1	
Description : volcanic disruption of repository			
Annual Rat. [ ] Move WPs to a Pathway			
Can Event	Pathway Name : PATH1	Path 1 of 5	
	Number of WPs moved: 1	(Constant )	
Define Eve	F1=Prev F2=Next		
Event Consequences			
Move WPs to AE			
Move WPs to a pathway			
Disrupt WPs			
Move Contents of pathway to AE			
Modify Parameters			
			F1=Prev F2=Next F3=Add F4=Del

This moves a specified number of waste packages directly to a specified pathway. The waste packages which are moved are chosen randomly from each specified waste package group. It is assumed that the inventory released from each package which is moved is equal to the inventory of an *unfailed* package at the time of the event. It is also assumed that all of the waste package's inventory is immediately released to the pathway (i.e., the waste package is completely disrupted during the movement and release of the contents is not limited by any alteration, dissolution, or mass transfer processes at the waste package). Note that the total number of waste packages remaining in the repository is not updated (reduced) subsequent to an event (it is assumed that the number of waste packages which are moved is a small fraction of the total).

Waste packages can be moved to any of the pathways which have been previously defined by the user. The user browses through the pathways using the <F1> and <F2> keys.

The user can specify a constant, a stochastic, or a function (e.g., of descriptor parameters). The value must, however, be greater than or equal to 0.

**Disrupt WPs:** This provides access to the following input window:

[ ] Repository Integration Program (Rev. 1.00 )	
General	PA Model Strategy Model Results
[ ] User's Guide Tutorial	
[ ] Waste Package Pathways Events Make Transfer File	
[ ] Disruptive Event Window	
Parameter ID : VOLCAN (must be "Event" type)	Event 1 of 1
Description : volcanic disruption of repository	
Annual R [ ] Disrupt WPs	
Can Event	Number of WPs disrupted : DISRUP (4 )
Define Event Descriptors	
Event Consequences	
Move WPs to AE	
Move WPs to a pathway	
Disrupt WPs	
Move Contents of pathway to AE	
Modify Parameters	
F1=Prev F2=Next F3=Add F4=Del	

This instantaneously disrupts a specified number of waste packages in place. The waste packages which are disrupted are chosen randomly from each specified waste package group. It is assumed that both the primary and secondary container are disrupted. The model also conservatively assumes that only previously unfailed containers are affected. Only an integer number of waste packages can be disrupted.

The user can specify a constant, a stochastic, or a function (e.g., of descriptor parameters). The value must, however, be greater than or equal to 0.

**Move Contents of Pathway to AE:** This provides access to the following input window:

[ ] Repository Integration Program (Rev. 1.00 )	
General	PA Model Strategy Model Results
[ ] User's Guide Tutorial	
[ ] Waste Package Pathways Events Make Transfer File	
[ ] Disruptive Event Window	
Parameter ID : VOLCAN (must be "Event" type)	Event 1 of 1
Description : volcanic disruption of repository	
Annual Rat [ ] Move contents of Pathway to AE	
Can Event	Pathway Name : PATH1 Path 1 of 5
	Fraction of mass moved:1 (Constant )
Define Eve	F1=Prev F2=Next
Event Consequences	
Move WPs to AE	
Move WPs to a pathway	
Disrupt WPs	
Move Contents of pathway to AE	
Modify Parameters	
F1=Prev F2=Next F3=Add F4=Del	

This moves a specified fraction of the total mass in a specified pathway directly to a the accessible environment. Mass can be moved from any of the pathways which have been previously defined by the user. The user browses through the pathways using the <F1> and <F2> keys.

The user can specify a constant, a stochastic, or a function (e.g., of descriptor parameters). The value must, however, be greater than or equal to 0 and less than or equal to one.

Note that the fraction of the mass which is moved pertains to the remaining balance. For example, if two events occur in a single timestep, each of which moves 0.6 of the mass from a particular pathway (which, say, totals 100 units), this would be interpreted as follows: first move  $0.6(100) = 60$  units of mass would be moved; then 0.6 of the remaining balance (40 units) would be moved (24 units), leaving 16 units of mass in the pathway.

**Modify Parameters:** This provides access to the following input window:

[ ] Repository Integration Program (Rev. 1.00 )	
General	PA Model
Strategy Model	
User's Guide Tutorial	
[ ]	
Waste Package	Pathways
Events	Make Transfer File
[ ] Disruptive Event Window	
Parameter ID : VOLCAN	(must be "Event" type) Event 1 of 1
Description : volcanic disruption of repository	
Annual Rat	[ ] Parameters modified by Event
Can Event	Parameter ID : WLEVEL
	Sensitivity : Add
Define Eve	Value : DELH (20 )
	F1=Prev F2=Next F3=Add F4=Del
Event Consequences	
Move WPs to AE	
Move WPs to a pathway	
Disrupt WPs	
Move Contents of pathway to AE	
Modify Parameters	
F1=Prev F2=Next F3=Add F4=Del	

This window allows a user to modify a system parameter contained in the parameter module data base by specifying its sensitivity to the occurrence of the disruptive event. The user does this by modifying the specified sensitive parameter using an *influence parameter* (shown as "Value" in the screen above). The sensitivity can take one of three forms: whenever the disruptive event occurs, the user-defined influence parameter can *replace*, *multiply*, or be *added* to the original parameter value.

Note that, unlike other consequences, the influence parameter is not necessarily realized every time its associated event occurs. In particular, *it is only realized every time the event occurs if it is specified as a descriptor parameter (or a function of a descriptor parameter)*. If it is specified as a regular parameter (or a function of a regular parameter), the same value for the influence parameter will be used for every event

within a particular realization (although it can change from realization to realization).

The user can modify as many parameters as desired. The user can add sensitive parameters with the <F3> key. Sensitive parameters can be deleted with the <F4> key. The <F1> and <F2> keys can be used to move between defined sensitive parameters.

For each sensitive parameter, the following information is required:

**Parameter ID:** This is the name of the sensitive parameter. Note that this must be a previously defined parameter within the parameter data base. Furthermore, it can not be a function or event type parameter. Only stochastic and constant parameters can be modified (a function type parameter must be modified by modifying its independent variables). If a function or event parameter is specified as the sensitive parameter, the modification is ignored.

**Sensitivity:** This is an option field which the user toggles (using the <PgUp> or <PgDn> keys) to choose the manner in which the influence parameter modifies the sensitive parameter (Add, Multiply, or Replace).

**Value:** This is the influence parameter. It can be specified as a constant or a parameter. If specified as a parameter, it can be a constant, a stochastic, or a function.

It is worth noting here how *Modify Parameters* can be used to keep a running total of the number of events which have occurred during the current realization. The user simply defines a parameter (e.g., TOTVOL) and specifies it as a constant equal to 0. He/she then specifies that this parameter is modified as follows:

[ ] Repository Integration Program (Rev. 1.00 )	
General	PA Model Strategy Model Results
[ ] Waste Package Pathways Events Make Transfer File	
[ ] Disruptive Event Window	
Parameter ID : VOLCAN	(must be "Event" type) Event 1 of 1
Description : volcanic disruption of repository	
Annual Rat [ ]	Parameters modified by Event
Can Event	Parameter ID : TOTVOL Parameter 1 of 4
Define Eve	Sensitivity : Add Value : VOLCAN (NA )
F1=Prev F2=Next F3=Add F4=Del	
Event Consequences	
Move WPs to AE	
Move WPs to a pathway	
Disrupt WPs	
Move Contents of pathway to AE	
Modify Parameters	
F1=Prev F2=Next F3=Add F4=Del	

Note that the influence parameter is the event parameter itself, and its value is simply added to TOTVOL every time the event occurs. TOTVOL will then be the running total of the



number of occurrences of event VOLCAN for each realization. The user may then wish use describe other parameters as a function of TOTVOL (e.g., a particular process may be a function of how many times the event has occurred to date), or may simply use this during post processing to identify correlations between system performance and the total number of event occurrences.

#### 5.4 Making a Total System Transfer File

Selecting [Make Transfer File] from the main menu provides access to the input windows for specifying general information regarding the creation of total system transfer file. Having done so, the user can then create the transfer file from this window.

The input window is as follows:

[ ] Repository Integration Program (Rev. 1.00 )			
General	PA Model	Strategy Model	Results
[ ] verification test file			
Waste Package	Pathways	Events	Make Transfer File
[ ] PERFORMANCE MODEL			
Number of timesteps		: 100	
Length of timestep		: 100	
Number of realizations		: 10000	
Run using expected values		: No	
WP Release			
Select Desired Output			
Define Realization Classes			
Create Transfer .TNF file			

**Number of timesteps:** This is the number of timesteps to be simulated. It must be greater than or equal to one. The beginning of the simulation (time = 0) is defined as the time of closure.

**Length of timestep:** This is the timestep length in years. It must be non-negative.

**Number of realizations:** This is the number of Monte Carlo realizations to be simulated. It must be greater than or equal to one.

**Run using expected values:** This is an option field which is answered Yes or No. If Yes, a single realization using the expected value for all stochastic parameters is produced.

**WP Release:** This provides access to the following window:

written to the output file.

The results under "Cumulative Radionuclide Release" refer to results that are summed over all timesteps. That is, they are cumulative with respect to time. "Sum" indicates that results are summed over all radionuclides. "by RN" indicates that a separate cumulative release is to be saved for each radionuclide.

**Normalize Results:** This determines whether or not the two cumulative radionuclide release results described below are to be normalized. It does not pertain to the "Waste Package Release Rate" described above. If the user selects Yes, the cumulative releases below are normalized to the 40 CFR 191.13 limit (normalization is discussed in detail in Appendix C). If the user selects No, results are output in units of Curies.

**From all Waste Packages:** This saves the cumulative waste package release summed over all waste package groups. If "Sum" is selected, for each realization, the cumulative waste package release (summed over all radionuclides) is saved. This results in  $NBNUM \times NREAL$  bytes being written to the output file. If "by RN" is selected, for each realization, the cumulative waste package release for each individual radionuclide is saved. This results in  $NBNUM \times NREAL \times NRN$  bytes being written to the output file.

**From each Waste Package Group:** This saves the cumulative waste package release for each waste package group. If "Sum" is selected, for each realization, the cumulative waste package release (summed over all radionuclides) is saved for each waste package group. This results in  $NBNUM \times NREAL \times NGRP$  bytes being written to the output file. If "by RN" is selected, for each realization, the cumulative waste package release for each individual radionuclide for each waste package group is saved. This results in  $NBNUM \times NREAL \times NRN \times NGRP$  bytes being written to the output file.

**Total to Accessible Environment:** This saves the total cumulative release to the accessible environment. If "Sum" is selected, for each realization, the cumulative (summed over all radionuclides) is saved. This results in  $NBNUM \times NREAL$  bytes being written to the output file. If "by RN" is selected, for each realization, the cumulative release for each individual radionuclide is saved. This results in  $NBNUM \times NREAL \times NRN$  bytes being written to the output file. Note that if results are being normalized, "Sum" provides the same results as "Cum. release to A.E (40 CFR 191.13)" specified above.

**From each Pathway to AE:** This saves the cumulative release from each specific pathway which discharges to the AE. If "Sum" is selected, for each realization, the cumulative release (summed over all radionuclides) is saved for each pathway. This results in  $NBNUM \times NREAL \times NPTHAE$  bytes being written to the output file. If "by RN" is selected, for each realization, the cumulative release for each individual radionuclide for each pathway is saved. This results in  $NBNUM \times NREAL \times NRN \times NPTHAE$  bytes being written to the

output file. Note that results for pathways which are not directly connected (i.e., do not discharge) to the AE (and are therefore always 0) are not written.

**From all Pathways:** This saves the cumulative release from each specific pathway. If "Sum" is selected, for each realization, the cumulative release (summed over all radionuclides) is saved for each pathway. This results in  $NBNUM \times NREAL \times NPTH$  bytes being written to the output file. If "by RN" is selected, for each realization, the cumulative release for each individual radionuclide for each pathway is saved. This results in  $NBNUM \times NREAL \times NRN \times NPTH$  bytes being written to the output file.

**Maximum Release Time Histories:** This saves the time history of release (in Curies) for the three realizations having the highest total cumulative waste package release. Time histories are saved only for those categories for which cumulative results are being saved in the column directly above. For example, consider the following screen:

Repository Integration Program (Rev. 1.00)			
SELECT DESIRED OUTPUT			
Waste Pkg. Rel. rate 10 CFR 60.113 : Yes	Cum. release to A.E. 40 CFR 191.13 : No		
Cumulative Radionuclide Release	Sum	by RN	
From all Waste Packages	No	No	
From each Waste Package Group	No	No	
Total to Accessible Environment (AE)	Yes	No	
From each Pathway to AE	No	No	
From all Pathways	No	Yes	
Maximum Release Time Histories	Yes	Yes	
Normalize Results ? No			

In this case, a total of  $3 + 3 \times NRN \times NPTH$  time histories would be saved: 3 time histories of total release to the AE; and 3 time histories of release for each individual radionuclide from each individual pathway. Each time history would result in  $NBNUM \times NTIME$  bytes being written to the output file.

**Define Realization Classes:** This provides access to an input window for defining importance sampling of disruptive events. Importance sampling is discussed in detail in Appendix C of the RIP Theory and Capabilities Manual. Importance sampling of events allows the user to emphasize realizations in which relatively low probability events occur. This increases the efficiency of the Monte Carlo sampling technique and improves the accuracy of the low probability tails of results distributions.

The user implements the event importance sampling technique by dividing all possible realizations into a number of *realization classes*, each realization class having a specified *pruning factor*. Realization classes are defined in terms of disruptive events. For example, suppose that the user has defined two types of disruptive events, referred to here as HUMAN and VOLCAN (specification of disruptive event types is discussed in Section 5.3). The user might then define one realization class to be "all realizations in which event VOLCAN occurs". By default, one class always exists which is called "all other realizations". In this case, this would include realizations in which either no disruptive events occurred or event HUMAN occurred.

Each disruptive event class is assigned a pruning factor (which must be greater than or equal to one). The pruning factor determines what fraction of the realizations in a given class are actually simulated. For example, if a class was assigned a pruning factor of 100, only one in every 100 realizations would actually be simulated, and the rest would be ignored. For a pruning factor of 1, every realization is simulated. To understand how this will affect the efficiency of the Monte Carlo sampling, refer again to the example above. Suppose that event VOLCAN was relatively rare (e.g., with a rate of occurrence of  $10^{-7}$  per year). This would mean that for a 10,000 year simulation, there is only a  $10^{-3}$  probability of realizing such an event. Hence, in order to properly represent this event in a Monte Carlo simulation, more than  $10^3$  realizations would have to be simulated. If, however, we were to specify a pruning factor of 1 for the class which consists of realizations with event VOLCAN, and a pruning factor of 100 for all other realizations, we would increase the frequency at which realizations in which VOLCAN occurred were simulated (by a factor of 100). In effect, if we had actually simulated  $10^4$  random realizations, it would seem that with respect VOLCAN type events we had simulated  $10^6$  random realizations. On the other hand, with respect to HUMAN realizations and all realizations in which no event had occurred, it would seem as if we had simulated somewhat less than  $10^4$  realizations (since we have artificially increased the rate at which we sampled VOLCAN realizations).

The user should consult Appendix C of the RIP Theory and Capabilities Manual for a more complete description of the event importance sampling algorithm.

Realization classes are defined in the following window:

Repository Integration Program (Rev. 1.00)			
General	<u>PA Model</u>	Strategy Model	Results
User's Guide Tutorial			
[ ]			
<u>Waste Package</u>	Pathways	Events	Make Transfer File
[ ]			
Near Field Conditions	Waste Package Description	<u>Make WP File</u>	
[ ]			
REALIZATION CLASSES			
Events in Class			Class 1 of 2
VOLCAN			
Pruning Factor : 1			
F1=Prev F2=Next F3=Add F4=Del F5=Display Events			

By default, a single realization class is originally defined, and this realization class is called "all other realizations". Its default pruning factor is 1. Additional classes can be added using the <F3> key. Defined classes can be browsed or deleted using the <F1>, <F2>, and <F3> keys. Note that regardless of how many classes are defined, the last class is always "all other realizations", and this name can not be changed. The pruning factor for this default class, however, can be edited.

After defining a new class, the user must specify which events the class contains. This is done by typing in the name of the event (or events). Multiple events must be separated by commas (e.g., "VOLCAN,HUMAN"). *Only events which have been previously specified within the disruptive events input window (see Section 5.3) can be specified.* As a reminder, the <F5> key provides a list of all defined events for the user.

**Create Transfer File:** Selecting this field creates a transfer file. Prior to creating the file, RIP goes through a series of data checks to ensure a consistent set of input. If any input errors are detected, creation of the file will be halted and an appropriate error message will appear alerting the user to the input error. If no input errors are detected, creation of the transfer file will commence. The progress of the transfer file creation is tracked on the screen by indicating which realization is currently being written.

The transfer file that is created is called NAME.TNF, where NAME is the name of the last input file specified by the user.

**WARNING:** Because creation of the transfer file could potentially crash the program if severe input errors are present, the user should use <ALT>-<S> to ensure that the current data set is saved prior to creation of the file.

## 5.5 Implementing the Back End Program

After creating a transfer file, the back end program must be implemented to carry out the actual simulations and produce a results file (as discussed in Section 2.3). After selecting the appropriate machine on which to run the back end (and installing the program as described in the README file on the distribution disk), and copying the transfer file created by the front end to the back end, the back end can be run by typing the following command:

```
RIP2 inputfile >outputfile
```

where "inputfile" is the name of the *transfer file* and "output file" is the name of the *results file* which will be produced.

A number of command line flags can also be set in order to modify the behavior of the back end programs. A summary of all the command line flags can be seen by typing

```
RIP2 -?
```

```
usage: rip2 [-2 file] [-d file] [-o opt] [-c col] [-eEfgmrsvz1?] [file]
```

The following command line flags are recognized:

```
-2 file  Redirect stderr (standard error) to file. *
-d file  Debug output directed to "file".
-o opt   Debug option.
-c col   This column gets all the mass.
-e       Echo input to stderr, other results to stdout. *
-E       Echo input to stdout -- don't do anything. *
-f       Flush stdout after each realization.
-g       Produce time history in grams.
-m       Report available memory (DOS only).
-r       Terminate lines with CR/LF.
-s       Status report to display. *
-v       Display program version. *
-z       Turn off decay.
-1       Run one realization only. *
-?       Display this help message.
```

Most of these flags are primarily programmer's tools and are not documented here. Those flags which the user may find useful are highlighted (with a \*) above and summarized below:

```
RIP2 test.tnf -2>err.out >test.res
```

This command will read the transfer file "test.tnf", send standard error messages to "err.out", and send the results file to "test.res". Note that if the results and the standard error messages are not redirected to files, by default they will be printed to the screen.

```
RIP2 test.tnf -E >echo.out
```

This command will read the transfer file "test.tnf" and create an echo file (which is sent to "echo.out"). After writing the echo file, it will terminate (i.e., it will not actually carry out a simulation). Hence, this is simply a way to create an echo file to verify that the transfer file has been read properly. The content of the echo file is similar (although more terse) than

has been read properly. The content of the echo file is similar (although more terse) than that of the front end echo file created using the [GENERAL] input window.

```
RIP2 test.tnf -e -2>err.out >test.res
```

This command will read the transfer file "test.tnf", send standard error messages to "err.out", and send the results file to "test.res". In addition, however, the input is echoed to the error message file "err.out".

```
RIP2 test.tnf -s -2>err.out >test.res
```

This command will read the transfer file "test.tnf", send standard error messages to "err.out", and send the results file to "test.res". In addition, the progress of the simulation (realization number and timestep) is sent to the display screen. Note, however, that this option will tend to slow execution somewhat (on the order of 10%).

```
RIP2 -v
```

This command will simply display the version number of the program which you have.

## 5.6 Input Documentation

It is useful to summarize here the manner in which the input into RIP can be documented by the user. As has been discussed in previous sections, the front end provides three mechanisms for documentation:

- The user can create an echo file from within the [General] input window. This creates an ASCII file summarizing in readable format all of the input parameters specified by the user. This is the most complete and easiest method for documenting input files.
- The user can create summaries of the parameter database using the <F6> from within the parameters module. This essentially results in a subset of the information contained in the echo file described above.
- The user can save and/or print every input window using the <Ctrl>-<S> and <Ctrl>-<P> keys.
- The user can create an echo file using the "-E" or "-e" command line flag when running the back end (RIP2).

**6. DATA ENTRY AND APPLICATION OF THE STRATEGY EVALUATION  
MODEL**

(in preparation)



## **7. RESULTS POST-PROCESSING AND DISPLAY**

(in preparation)

## 8. REFERENCES

Bear, J., 1979, Hydraulics of Groundwater, McGraw-Hill Book Co., New York.

Golder Associates Inc., 1977, *Development of Site Suitability Criteria for the High Level Waste Repository*, prepared for Lawrence Livermore National Laboratory.

Golder Associates Inc., 1986, *Application of Performance-Based Decision Process in Selection of Crystalline Rock Sites for Deep Geologic Disposal of Nuclear Waste*, prepared for Battelle Memorial Institute, Office of Crystalline Repository Development.

Golder Associates Inc., 1992, *The RIP Repository Performance Assessment and Strategy Evaluation Model: Theory and Capabilities*, 903-1371, prepared for OCRWM under contract to Argonne National Laboratory.

National Research Council, 1990, *Rethinking High-Level Radioactive Waste Disposal - A Position Paper of the Board on Radioactive Waste Management*, National Academy Press, Washington, D.C.

Nuclear Regulatory Commission, 1983, *In Situ Test Programs Related to Design and Construction of High-Level Nuclear Waste (HLW) Deep Geologic Repositories*, NUREG/CR-3065, prepared by Golder Associates for the Office of Nuclear Material Safety and Safeguards, U.S. Nuclear Regulatory Commission.

Nuclear Waste Technical Review Board, 1990, *First Report to the U.S. Congress and the U.S. Secretary of Energy*, U.S. Government Printing Office, Washington, D.C.