PNNL-13932- Volume 1

Pacific Northwest National Laboratory
Operated by Battelle for the
U.S. Department of Energy

User Instructions for the Systems Assessment Capability, Rev. 0, Computer Codes

Volume 1: Inventory, Release, and Transport Modules

P.W. Eslinger
D.W. Engel
L.H. Gerhardstein
C.A. Lo Presti
W.E. Nichols
D.L. Strenge

June 2002

Prepared for the U.S. Department of Energy
under Contract DE-AC06-76RL01830

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC06-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service,
U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161
ph: (800) 553-6847
fax: (703) 605-6900
email: orders@ntis.fedworld.gov
online ordering: http://www.ntis.gov/ordering.htm

This document was printed on recycled paper.

PNNL-13932-Volume 1

User Instructions for the Systems Assessment Capability, Rev. 0, Computer Codes

Volume 1: Inventory, Release, and Transport Modules

Paul W. Eslinger
David Engel
Larry Gerhardstein

Charles Lo Presti
William Nichols
Dennis Strenge

June 2002

Prepared for
the U.S. Department of Energy
under Contract DE-AC06-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Contents

Figures

Tables

1.0 Introduction and Background

1.1 The Groundwater/Vadose Zone Project

In late 1997, the U.S. Department of Energy (DOE) established the Groundwater/Vadose Zone Integration Project (Integration Project). One activity of the Integration Project is an assessment of cumulative impacts for Hanford Site wastes on the subsurface environment and the Columbia River. Through the application of a system assessment capability (SAC), decisions for each cleanup and disposal action will be able to take into account the composite effect of other cleanup and disposal actions.

1.2 Hanford Site

The Hanford Site lies within the semi-arid Pasco Basin of the Columbia Plateau, in southeastern Washington State (see Figure 1-1). The site occupies approximately 1,517 km^2 (586 square miles) and is located north of the city of Richland, Washington. About 6% of the land area has been disturbed and is actively used. The site is located upstream of the confluence of the Yakima and Snake Rivers with the Columbia River, approximately 25 miles north and upstream of the Oregon border. A dry area known for its sandy soils, basalt ridges, and shrub-steppe vegetation, the Hanford Site is bordered by the Columbia River on the north and east. The Yakima River flows near a portion of the southern boundary of the site before it joins the Columbia River south of the city of Richland.

1.3 Overview of the SAC Systems Code

The SAC Systems Code is a tool used to simulate the migration of contaminants (analytes) present on the Hanford Site and to assess the potential impacts of the analytes, including dose to humans, socio-cultural impacts, economic impacts, and ecological impacts. The system of codes includes existing computer programs, new computer programs, electronic data libraries, and data formatting processors (or data translators). The relationships among code modules that make up the SAC Systems Code are illustrated in Figure 1-2.

Major modules appearing on the left side of the diagram perform inventory and transport calculations providing estimates of the concentrations of analytes in various media. Modules shown on the right perform calculations related to impacts from the contaminated media. Impacts include potential effects on humans, the ecology of the area, the economy of the region, and the proximity of contaminants to social and cultural resources.

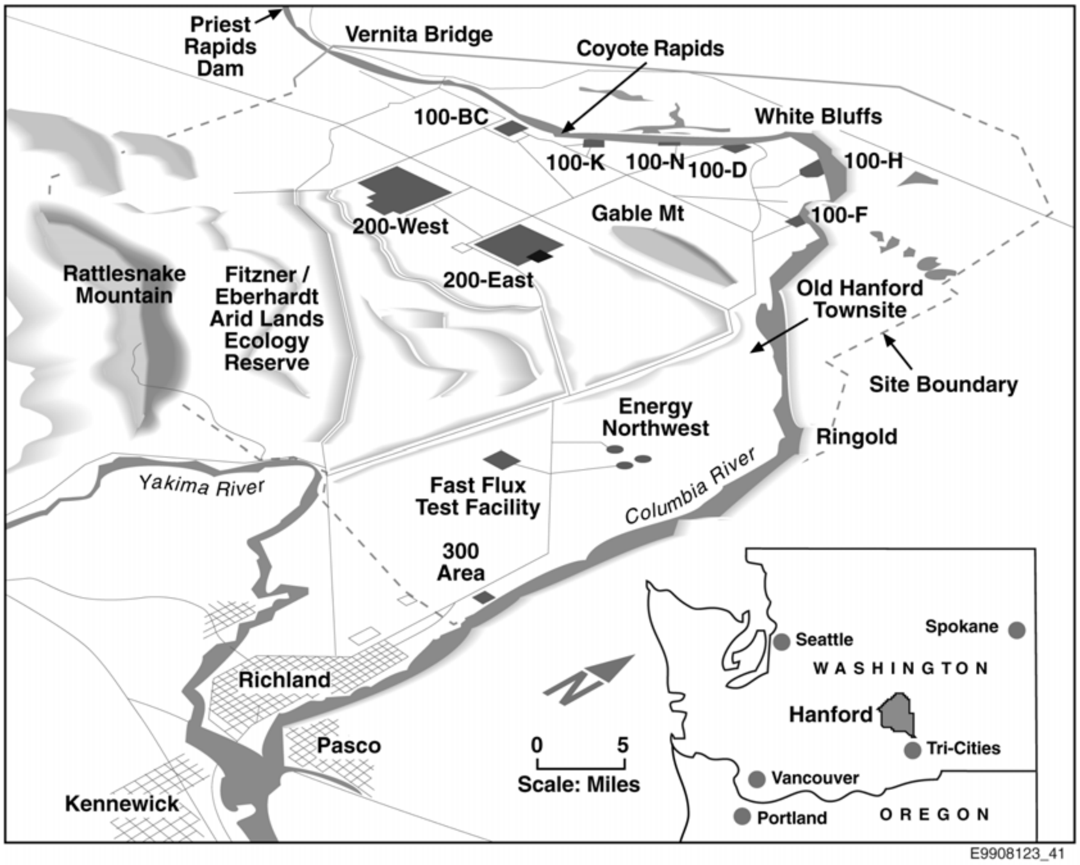


Figure 1-1 Hanford Site

The general approach to handling uncertainty in SAC, Rev. 0, is a Monte Carlo approach. Conceptually, one generates a value for every stochastic parameter in the code (the entire sequence of modules from inventory through transport and impacts) and then executes the simulation, obtaining an output value, or result. This process is often called *one realization*. One then repeats the entire process, obtaining another result that is different from the first, but as equally likely to occur as the first result. After repeating this process a number of times, one has a set of equally likely consequences that represent the statistical distribution of all outcomes. Several specialized sampling techniques have been developed to reduce the number of realizations required in a Monte Carlo analysis to obtain a satisfactory description of the output distribution. One of these techniques, called *Latin Hypercube Sampling* (Iman and Conover 1982), has proven successful for mass transport applications in groundwater systems. The general Monte Carlo approach still applies, and the specific values of the input parameters are chosen from the same statistical distributions, but the sampling scheme spreads the values in such a way as to reduce sampling variability while also supporting a correlation structure between input variables.

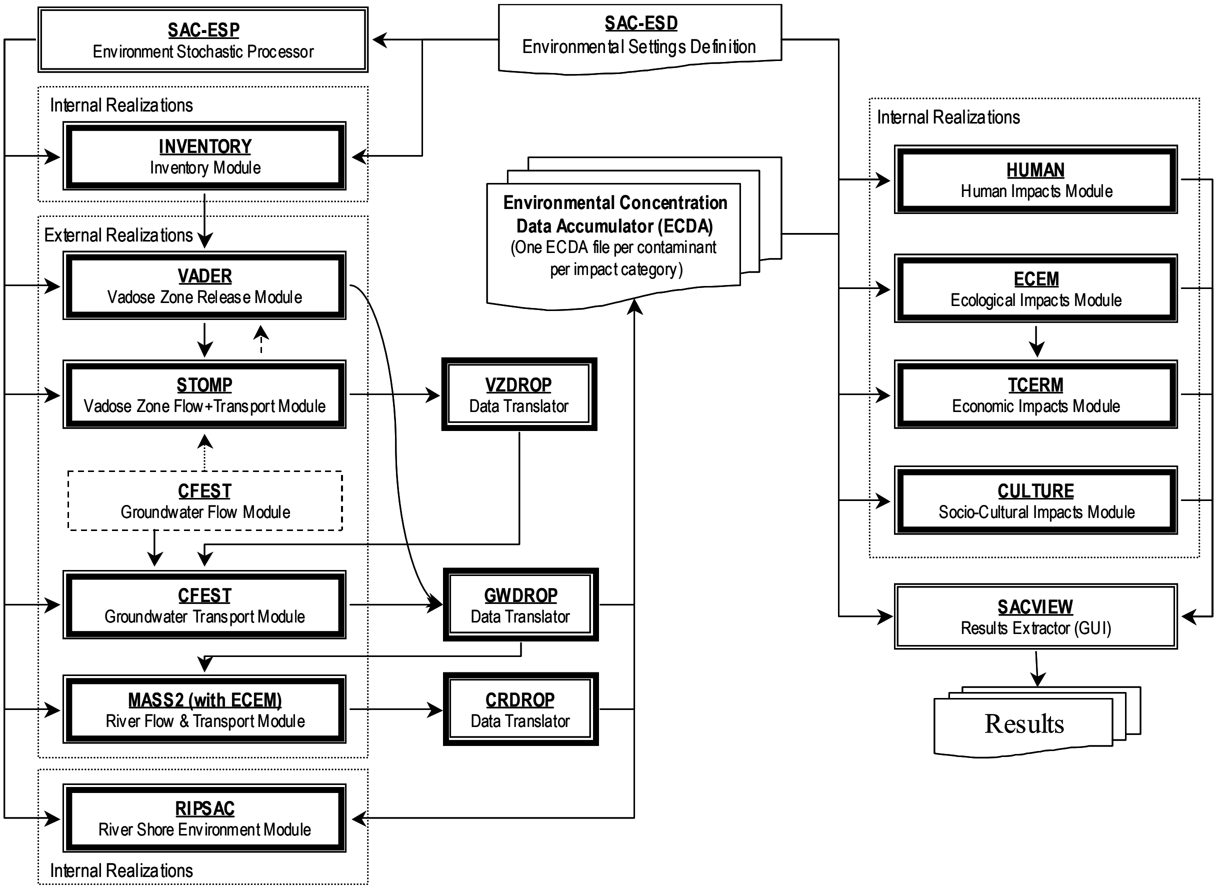

SAC-ESP
Environment Stochastic Processor
SAC-ESD
Environmental Settings Definition
Internal Realizations
INVENTORY
Inventory Module
External Realizations
VADER
Vadose Zone Release Module
STOMP
Vadose Zone Flow+Transport Module
CFEST
Groundwater Flow Module
CFEST
Groundwater Transport Module
MASS2 (with ECEM)
River Flow & Transport Module
RIPSAC
River Shore Environment Module
Internal Realizations
VZDROP
Data Translator
GWDROP
Data Translator
CRDROP
Data Translator
Environmental Concentration Data Accumulator (ECDA)
(One ECDA file per contaminant per impact category)
Internal Realizations
HUMAN
Human Impacts Module
ECEM
Ecological Impacts Module
TCERM
Economic Impacts Module
CULTURE
Socio-Cultural Impacts Module
SACVIEW
Results Extractor (GUI)
Results

Figure 1-2 Module Information Flow for the SAC Rev. 0 Systems Code

1.4 Purpose of This Document

The SAC codes on the left side of Figure 1-2 address inventory tracking, release of contaminants to the environment, and transport of contaminants through the unsaturated zone, saturated zone, and the Columbia River. This document contains detailed user instructions for the following computer codes:

- SAC ESP: Environmental stochastic processor. This processor controls the execution of all the codes on the left side of Figure 1-2.
- INVENTORY: Inventory tracking and aggregation code.
- VADER: Vadose zone release module. The function of this code includes tracking movement of waste during remediation activities.
- VZDROP: Utility code to pass mass flux from the unsaturated zone transport code (STOMP) to the groundwater transport code (CFEST96).
- GWDROP: Utility code to pass mass flux from the groundwater transport code (CFEST96) to the river transport code (MASS2). In addition, this code saves groundwater concentrations for use by the impacts codes.
- CRDROP: Utility code to extract concentration data from the river module (MASS2) and save it for use by the impacts codes.

- RIPSAC: Riparian zone module. The function of this code is to calculate concentrations of contaminants in seep water and soil near the edge of the river.

Instructions for some of the codes on the left side of Figure 1-2 and some utility codes are not provided in this document. The status of user instructions for these codes is as follows:

- STOMP: A standalone user's guide is provided in White and Oostrom (2000). Section 3.4.1.7 documents the few changes made to integrate STOMP into the SAC framework.
- CFEST: A user's guide for the CFEST code has been published separately (CFEST 2001).
- MASS2: No published user instructions are available at this time for the MASS2 code.
- ECDA: This utility code is called by the ESP (or it can be executed in standalone mode) to create most of the files identified on the ESD FILE keyword (Section 2.1.10). No published user instructions are available at this time for the ECDA code.
- HEADER: This utility code is called by the ESP (or it can be executed in standalone mode) to create the HEADER files identified on the ESD FILE keyword (Section 2.1.10). No written user instructions are available at this time for the HEADER code.

The suite of computer codes for Rev. 0 of SAC is much broader than just the inventory, release, and transport functions. The codes also address socio-cultural impact assessment, ecological impacts assessment, human impacts assessment, and Tri-Cities economic impacts assessment. User instructions for the impacts codes are provided in Eslinger et al. (2002).

The assumption is made that users of this document are knowledgeable computer users. In this case, the computer system runs the Linux operating system. The user will have to create directories, create and edit files, copy files to subdirectories, and change directories in the system.

2.0 Environmental Settings Definition (ESD) Files

As can be seen in Figure 1-2, the SAC Rev. 0 systems code consists of a number of components that can be executed separately. A number of pieces of information, such as the start time and stop time of a simulated problem, are needed by more than one component of the systems code. The environmental settings definition (ESD) keyword file was designed to contain this common information. Generally, if information is needed by more than one module of the suite of codes, it will be entered in the ESD keyword file.

The river transport model can be run with or without background concentrations from upriver sources. A typical method of analyzing a Hanford-related problem is to run the river code with two data sets. First, a transport run is made using background values, but no source term from the Hanford site is introduced. Then, another transport run is made that is identical to the first run except that a Hanford source term is introduced. If the impacts codes are run using these two data sets, their results can be differenced to determine the contribution from Hanford sources. The groundwater model in SAC, Rev. 0, is not set up to model background concentrations; thus differencing does not currently apply to impacts based on groundwater concentrations. A major effect of using background concentrations is that concentration data must be saved for both the runs: the one where only background concentrations are modeled and the one where both background and Hanford concentrations are modeled. The current implementation uses two separate ESD keyword files to control these runs, and the concentration data are saved in totally separate files.

2.1 ESD Keywords Used by Environmental Codes

In the following keyword descriptions, some data are optional for a particular problem definition and some are required. Data that are required are enclosed in square brackets. If AB were required, it would be denoted by [AB]. If only one of the three items AB, BC, CD were required, it would be written as [AB|BC|CD]. The vertical bars indicate that the user must choose one of the items in the list. Optional items are enclosed in normal brackets; for example, if DE were an optional entry, it would be denoted by {DE}. The {} or [] brackets do not need to be entered when the keyword is constructed. The keyword name can contain any number of characters; however, only the first eight characters are used (for example, REALIZAT has the same effect as REALIZATION). In some instances, numerical values or quote strings are associated with a modifier. In this description, the association is indicated by using the = symbol. The = symbol is not required but may be used when the keyword is constructed. When a numerical value or quote string is associated with a modifier, it must be entered on the input line directly after the modifier.

2.1.1 AGGREGATE Keyword in the ESD Keyword File

The AGGREGATE keyword identifies the disposal sites where disposal actions will be defined and aggregated by the INVENTORY program for use by the VADER program. The syntax of the keyword is the following:

```
AGGREGATE [ID="quote 1"] {COMPUTE} [TITLE= "quote 2"] {OFFSITE}
   {NW_NORTH} {SE_NORTH} {NW_EAST} {SE_EAST} {AREAX}
```

Table 2.1 provides the modifiers and associated data for the LOCATION keyword.

Table 2.1 Modifiers Associated with the AGGREGATE Keyword in the ESD File

Modifier	Description
AREAX	Multiplier to apply to the vadose zone area for purposes of vadose zone transport simulations; a negative value will invoke the K_s-dependent area method (see below for details). This modifier is not used when the OFFSITE modifier is used.
COMPUTE	If the COMPUTE modifier is entered, this aggregation site will be included in the simulation. If COMPUTE is not entered, this location will be ignored.
ID	The location identification string is entered using the ID modifier. This string is limited to 15 characters and must be unique. It is used to associate other data with a specific aggregation site.
NW_EAST	Easting coordinate (in meters) of the northwest corner of the vadose zone release site. This modifier is not used when the OFFSITE modifier is used.
NW_NORTH	Northing coordinate (in meters) of the northwest corner of the vadose zone release site. This modifier is not used when the OFFSITE modifier is used.
OFFSITE	When this entry is present, the aggregation site is defined to be off the Hanford Site. For such sites, mass balance information will be generated (if requested by other keyword records), but the waste will not be disposed to any aggregation site on Hanford and will not contribute to any environmental releases or impacts.
SE_NORTH	Northing coordinate (in meters) of the southeast corner of the vadose zone release site. This modifier is not used when the OFFSITE modifier is used.
SE_EAST	Easting coordinate (in meters) of the southeast corner of the vadose zone release site. This modifier is not used when the OFFSITE modifier is used.
TITLE	A descriptive title for this aggregation site is entered using the TITLE modifier. This quote string is limited to 72 characters. It is used only for output labeling purposes.

The coordinates define the locations of a northwest and southeast corner of the vadose zone site and will usually be taken from the Waste Information Database (WIDS). These coordinates are expressed in terms of the Lambert projection of the Washington State Plane North American Datum of 1983, expressed in meters. They are used to define an area for the vadose zone release (VADER) model and the vadose zone flow and transport (STOMP) model. For SAC, the vadose zone site is presumed to be represented by a rectangular area oriented to the cardinal directions and constrained at the northwest and southeast vertices by the coordinates specified.

The result files of INVENTORY (.res files) will include an entry for aggregation sites that have COMPUTE and OFFSITE on the AGGREGATE keyword, but no release values will be written. There will be one line in the file stating, “Offsite location, no data provided for this location.”

For the vadose zone transport calculations, an optional AREAX factor can be used to specify a multiplier for the vadose zone area. This is used to represent lateral spreading that would occur where large artificial liquid discharges happen, wherein the actual area of vadose zone transport is larger than the facility footprint given by WIDS coordinates. When ESP creates input files for STOMP runs, the area derived from the specified coordinates will be multiplied by AREAX; then the x- and y- extents written to the STOMP input file for one-dimensional cases will be the square of the resulting scaled area. A negative value may be specified for AREAX to direct the code to compute the AREAX factor for each site and each realization using the K_s-dependent approach. For this approach, ESP will compute the effective wetted area of a vadose zone site using the equation

$$A_X = \frac{|Q_{\max}|}{K_{s|\min} A_0}, A_X \geq 1$$

where A_X is the multiplier of the facility footprint area (dimensionless), Q_{max} is the maximum artificial liquid discharge rate (m/s), $K_{s|min}$ is the minimum hydraulic conductivity of all layers for the given site (m/s) and realization, A_0 is the nominal area (m^2) defined by the coordinates in the AGGREGATE keyword. AREAX is constrained by SAC ESP to be equal to or greater than 1.0 (cannot reduce the effective area to less than the facility footprint area). The horizontal extent of the vadose zone wetted area written in the STOMP input (STOMP ~Grid Card) file is

$$x = y = \sqrt{A_X A_0}$$

An example AGGREGATE keyword for specification of an offsite disposition to the Waste Isolation Pilot Plant (WIPP) to be included in the analyses is as follows.

```
AGGREGATE ID=“WIPP” TITLE=“Waste Isolation Pilot Plant” OFFSITE COMPUTE
```

An example AGGREGATE keyword for specification of an onsite disposition is the following:

```
AGGREGATE ID="116-B-2" NW_NORTH=144515.408 SE_NORTH=144507.060
   NW_EAST=565409.014 SE_EAST=565417.362 TITLE="Trench"
   AREAX=-1.0 COMPUTE
```

2.1.2 ANALYTE Keyword in the ESD Keyword File

The ANALYTE keyword is used to define the analytes to be used in the simulation. The syntax for this keyword is the following:

```
ANALYTE [ID=“quote 1”] [TYPE=“quote 2”] [NAME=“quote 3”] {COMPUTE}
   {MOLDIFF=N1}  {HENRY=N2}  {DFIMM=N3}  {DFSED=N4}
   {GAMMA=N5} {HALFLIFE=N6} {MOLWGT=N7} {SPECIFIC=N8}
```

A separate ANALYTE keyword must be entered for every analyte to be included in the simulation. The optional modifier COMPUTE can be entered. If COMPUTE is not present, the analyte will not be included in the run although information is included in the environmental settings file.

Table 2.2 provides the modifiers associated with the ANALYTE keyword.

Table 2.2 Modifiers Associated with the ANALYTE Keyword in the ESD File

Modifier	**Description**
ID	The quote string associated with the ID modifier is an analyte identification string up to six characters in length. The analyte identification string is case sensitive, and spaces or hyphens change the definition. All data in the analyte identification strings must satisfy the following conventions: • Only the first entry in the analyte identification string is capitalized • No embedded spaces or hyphens are used, even for radionuclides • Individual elements are defined using the standard element abbreviation
TYPE	The quote string associated with the TYPE modifier string is a two-character analyte type indicator. The following are the valid entries for this string: • NR – if the analyte is a radioactive element or an inorganic compound containing a radionuclide • NS – if the analyte is a stable (nonradioactive) element or inorganic compound • OR – if the analyte is an organic compound containing a radionuclide • OS – if the analyte is an organic compound, containing a stable (nonradioactive) elemental analyte or compound
NAME	The quote string associated with the NAME modifier is an analyte name or description up to 72 characters in length.
MOLDIFF	The numerical entry associated with the MOLDIFF modifier is the molecular diffusivity of the analyte. This value has units of cm^2/sec and is used only in the ecological modules. Entry of this modifier is optional. If it is not present, the value of MOLDIFF defaults to zero.
MOLWGT	The numerical entry associated with the MOLWGT modifier is the molecular weight of the analyte. This value has units of g/mole.
HENRY	The numerical entry associated with the HENRY modifier is the Henry's law coefficient for organic analytes. This value has units of Pa-m^3/mole and is used only in the impacts modules. Entry of this modifier is optional. If it is not present, the value of HENRY defaults to zero.
HALFLIFE	The numerical entry associated with the HALFLIFE modifier is the half-life of the analyte. This value has units of years. Entry of this modifier is necessary when defining a radioactive analyte but should be omitted for nonradioactive analytes. If it is not present, the value of HALFLIFE defaults to infinity (decay constant value of zero).

Modifier	**Description**
SPECIFIC	The numerical entry associated with the SPECIFIC modifier is the specific activity of the analyte. This value has units of curies per gram. Entry of this modifier is required if the analyte is radioactive. It is not used for nonradioactive analytes.
DFIMM	The numerical entry associated with the DFIMM modifier is the immersion external dose factor for radioactive analytes. This value has units of mrad/yr per uCi/m^3 and is used only in the impacts modules. Entry of this modifier is optional. If it is not present, the value of DFIMM defaults to zero.
DFSED	The numerical entry associated with the DFSED modifier is the sediment external dose factor for radioactive analytes. This value has units of Sv-m^3/sec-Bq and is used only in the impacts modules. Entry of this modifier is optional. If it is not present, the value of DFSED defaults to zero.
GAMMA	The numerical entry associated with the GAMMA modifier is the gamma decay energy for radioactive analytes. This value has units of MeV/nt and is used only in the impacts modules. Entry of this modifier is optional. If it is not present, the value of GAMMA defaults to zero.
COMPUTE	The optional modifier COMPUTE can be entered. If COMPUTE is not present, the analyte will not be included in the run although information is included in the environmental settings file.

It is expected that common chemical formulae or acronyms would be used for the analyte identification string. The analyte identification string is case sensitive and all symbols, including spaces and hyphens, are significant. The analyte identification string is also used as a directory name, so naming conventions in Windows or Linux must also be considered when assigning the identification string. The following conventions apply:

- No embedded spaces or hyphens are used, even for radionuclides (for example, *Np237* could be used for the nuclide neptunium-237 rather than *Np 237* or *Np-237*).
- Individual elements should be defined using the standard abbreviation (for example, use *U* for uranium).
- The TYPE modifier specifies whether the analyte is radioactive or a nonradioactive chemical.

The following ANALYTE keywords select the analytes carbon tetrachloride and strontium-90 for analysis.

```
ANALYTE ID="CCl4"   NAME="Carbon Tetrachloride" TYPE="NS"  COMPUTE
  HENRY  = 2.99E+03   MOLDIFF=5.0E-08    MOLWGT = 153.8  HALFLIFE = 0.0

ANALYTE ID="Sr90"  NAME="Strontium 90"  TYPE="NR"          COMPUTE
  MOLWGT    = 89.9077    MOLDIFF  = 5.0E-11
  HALFLIFE   = 28.78        SPECIFIC  = 1.3976E+2
  DFIMM        = 0                DFSED      = 3.72000E-21
  GAMMA      = 0                HENRY    = 0
```

2.1.3 BACKGROUND Keyword in the ESD Keyword File

The BACKGROUND keyword tells the main processor (SAC-ESP) that the background case will be modeled. The following is the keyword's syntax:

```
BACKGROUND {COMPUTE}
```

The background case will not be modeled unless the modifier COMPUTE is present.

2.1.4 BALANCE Keyword in the ESD Keyword File

The BALANCE keyword identifies the years for which mass balance data will be generated. The syntax of the keyword is the following:

BALANCE [ALL | [year$_1$] {year$_2$} {year$_3$} … {year$_n$}]

If the modifier *ALL* is present, then mass balance information is saved for all years defined in the PERIOD keyword record (Section 2.1.18) from the start year though the end year. One or more years must be listed explicitly if the modifier *ALL* is not present. If a year is listed twice, an error message is written.

The following is an example BALANCE keyword that saves mass balance information for the years 1950, 1960, 1970, 1980, 1990, and 2000:

```
BALANCE 1950 1960 1970 1980 1990 2000
```

To include a mass balance calculation for all years in the simulation, the following BALANCE keyword can be used:

```
BALANCE ALL
```

2.1.5 CREATDIR Keyword in the ESD Keyword File

The CREATEDIR keyword tells SAC-ESP to only create the subdirectory structure for the analysis. The following is the keyword's syntax:

```
CREATEDIR {COMPUTE}
```

When this keyword is entered and the modifier COMPUTE is present, the directory structure for the analysis will be created, and then execution of SAC-ESP will terminate. If the modifier COMPUTE is not present, then no directory actions are taken. The directory structure is defined by the contents of the following ESD keywords: MODULE (see Section 2.1.16), ANALYTE (see Section 2.1.2), and AGGREGATE (see Section 2.1.1).

2.1.6 DEBUG Keyword in the ESD Keyword File

The DEBUG keyword identifies the file in which to store the stochastic values generated by the SAC-ESP for use in other modules or processes. For example, the groundwater recharge rate in the vadose zone is used by both VADER and STOMP. The generated values will be output to the file identified using the DEBUG keyword. The following is the keyword's syntax:

```
   DEBUG [STOCHASTIC="quote"]
```

A file name must be supplied ("quote") if the values are to be printed in the file. For example, when storing the generated stochastic values in the file esp_stoch.out under the stoch subdirectory (relative to the assessment directory), the following keyword would be used:

```
   DEBUG STOCHASTIC="stoch/esp_stoch.out"
```

The debug keyword in the ESD keyword file is only used by the SAC-ESP program. Debug options in other codes are activated by specific inputs for those codes contained in other files.

2.1.7 DILUTE Keyword in the ESD Keyword File

The DILUTE keyword is used to enter the definition of a statistical distribution for stochastic water dilution variables. The general syntax for the DILUTE record is the following:

```
   DILUTE ["quote1"] [Dist_Index Parameters] {TRUNCATE U1 U2}  {"quote2"}
```

The entry for quote$_1$ must be a unique character string of up to 20 characters that will be used to identify this stochastic variable in subsequent uses. It is case sensitive and embedded spaces are significant. The entry for quote$_2$ is a description for the stochastic variable that can be up to 64 characters long. An entry for quote$_2$ is not required, although it is used for labeling purposes if present. Section 11.0 contains information on the statistical distributions available in the simulation.

A dilution factor that is triangular on the triple (0.2, 0.5, 0.99) could use the following keyword entry:

```
   DILUTE "ID#1"  6  0.2  0.5  0.99 "Example dilution factor for users guide"
```

The information on the DILUTE keyword is used only in RIPSAC. Each riverbank location where seep water concentrations are to be calculated must point to a DILUTE keyword by using the dilution ID in quote$_1$ on the LOCATION card entered in the RIPSAC keyword file. Multiple locations can point to the same DILUTE keyword.

2.1.8 END Keyword in the ESD Keyword File

The END keyword signifies the end of all keyword data. It should be the last keyword in the keyword file. Any data in the keyword file after the END keyword will be ignored. The syntax for this keyword record is the following:

```
END
```

There are no modifiers or quote strings associated with the END keyword.

2.1.9 EXEDIR Keyword in the ESD Keyword File

The EXEDIR keyword identifies the directory location of the program executable files. The syntax for this keyword record is the following:

```
EXEDIR [“quote”]
```

The EXEDIR keyword is used to identify the locations where programs reside that get executed through a call from the main processor SAC-ESP. These programs include the following:

- **ecda.exe** for setup of the concentration data files for later use by the impacts codes
- **header.exe** for setup of the header file for SACView for reading the concentration data files used by the impacts codes
- **ripsac.exe** for calculation of riparian zone concentrations
- **back_mass.exe** for setup of background concentration data files for use in the river transport model
- **inventory.exe** for calculation of the inventory by site and analyte.

These programs must reside under the specified directory, and the file names must be identical to those shown above (all in lowercase). The following is a sample keyword:

```
EXEDIR “/home/CODES/bin”
```

The location of other programs that are executed (run by a script generated by the SAC-ESP) is also identified by the EXEDIR keyword. If the EXEDIR keyword is not entered or if the path is blank, then the executable programs must reside under a directory that is specified by the path environmental variable for the user who is running SAC-ESP. For instance, on a Unix system, the user can specify a path name in the user’s definition file (.cshrc). The following programs must reside in a directory on this path. The program names must be presented exactly as shown below (all in lowercase): vader.exe, prestomp.exe, stomp.exe, vzdrop.exe, lprog3i_001.exe, prog3_001.exe, gethheld_001.exe, wdrop.exe, mass2, runmass2NB, crdropNB.pl, crdrop.exe, and crdrop_index.exe.

2.1.10 FILE Keyword in the ESD Keyword File

The FILE keyword is used to enter the names of many of the files used in a simulation run. The syntax for the FILE keyword record is the following:

```
FILE [NAME=“quote 1”] {ANALYTE=“quote 2”} {HEADER|KDSOIL|DILUTE}
   { C_HUMAN | C_ECONOM | C_CULTUR | C_ECOLOG |  C_HUMAN |
   C_ECONOM | C_CULTUR | C_ECOLOG } {CREATE} {SEED=N1}
```

The file names are entered in quote strings, which must be enclosed in double quotes. Path names up to 72 characters long are supported. At least one FILE keyword is required for every run of the code if concentrations are to be generated. Every file definition requires the entry of a separate FILE keyword. The table below provides the file type modifiers associated with the FILE keyword.

Table 2.3 Modifiers Associated with the FILE Keyword in the ESD File

Modifier	**Description**
C_HUMAN	This modifier indicates that the FILE keyword is defining a concentration file for human impacts. Concentrations for each analyte are contained in separate files. The ANALYTE modifier is used to associate an analyte with this file (see the ANALYTE keyword in Section 2.1.2).
C_CULTUR	This modifier indicates that the FILE keyword is defining a concentration file for cultural impacts. Concentrations for each analyte are contained in separate files. The ANALYTE modifier is used to associate an analyte with this file (see the ANALYTE keyword in Section 2.1.2).
C_ECONOM	This modifier indicates that the FILE keyword is defining a concentration file for economic impacts. Concentrations for each analyte are contained in separate files. The ANALYTE modifier is used to associate an analyte with this file (see the ANALYTE keyword in Section 2.1.2).
C_ECOLOG	This modifier indicates that the FILE keyword is defining a concentration file for ecological impacts. Concentrations for each analyte are contained in separate files. The ANALYTE modifier is used to associate an analyte with this file (see the ANALYTE keyword in Section 2.1.2).
CREATE	When this modifier is present, the file will be created under the control of the ESP. If the CREATE modifier is not present, no ESP-related actions occur for that file.
I_HUMAN	This modifier indicates that the FILE keyword is defining a record index file for mapping into all binary files of concentrations supporting human impacts.
I_CULTUR	This modifier indicates that the FILE keyword is defining a record index file for mapping into all binary files of concentrations supporting cultural impacts.
I_ECONOM	This modifier indicates that the FILE keyword is defining a record index file for mapping into all binary files of concentrations supporting economic impacts.
I_ECOLOG	This modifier indicates that the FILE keyword is defining a record index file for mapping into all binary files of concentrations supporting ecological impacts.
KDSOIL	This modifier indicates that the FILE keyword is defining a file to contain stochastic realizations of all of the random variables defined using the KDSOIL keyword.
DILUTE	This modifier indicates that the FILE keyword is defining a file to contain stochastic realizations of all of the random variables defined using the DILUTE keyword.
HEADER	This modifier indicates that the FILE keyword is defining a header file for use by the SACView graphical user interface that allows extraction of human-readable concentration data.

Modifier	Description
SEED	If the file name is associated with the KDSOIL or DILUTE modifiers, the SEED modifier must also be entered. The numerical value associated with the SEED modifier is the value for the random number generator and must have an entry in the range 1 to 999999.

The following are example entries that define the concentration files for carbon-14 and zinc for the full suite of cultural, ecological, economic, and human impacts:

```
FILE C_HUMAN ANALYTE=“C14”  NAME=“\Test\Human\C14\Conc.dat”
FILE C_CULTUR ANALYTE=“C14”  NAME=“\Test\Cultural\C14\Conc.dat”
FILE C_ECONOM ANALYTE=“C14”  NAME=“\Test\Ecomonics\C14\Conc_ec.dat”
FILE C_ECOLOG ANALYTE=“C14”  NAME=“\Test\Ecology\C14\Conc_bugs.dat”
FILE C_HUMAN ANALYTE=“Zn”  NAME=“\Test\Human\Zinc\Conc.dat”
FILE C_CULTUR ANALYTE=“Zn”  NAME=“\Test\Cultural\Zinc\Conczn.dat”
FILE C_ECONOM ANALYTE=“Zn”  NAME=“\Test\Ecomonics\Zinc\Conc_ec.dat”
FILE C_ECOLOG ANALYTE=“Zn”  NAME=“\Test\Ecology\Zinc\Conc_bugs.dat”
```

If present, the CREATE flag causes the following actions by the ECDA or HEADER codes under control of the ESP:

- Deletion of any existing file by that name and creation of a new file
- If the file is associated with the HEADER, DILUTE, KDSOIL, I_HUMAN, I_ CULTUR, I_ ECONOM, or I_ ECOLOG modifiers, new data are written to the file
- If the file is associated with the C_HUMAN, C_CULTUR, C_ECONOM, or C_ECOLOG modifiers, the concentration data are initialized to the value –1.0, except for groundwater concentrations, which are initialized to 0.0.

The following are example entries that define the files for soil-water Kd values and the water dilution factors for the river-shore module:

```
FILE KDSOIL NAME="KDSOIL.CSV" SEED=23232.0 CREATE
FILE DILUTE NAME="DILUTE.CSV" SEED=12345.0 CREATE
```

The following are example entries that define the ECDA record number index files for all four impact types:

```
FILE I_HUMAN  "HUMA_MAP.CSV" CREATE
FILE I_ECOLOG "ECOL_MAP.CSV" CREATE
FILE I_ECONOM "ECON_MAP.CSV" CREATE
FILE I_CULTUR "CULT_MAP.CSV" CREATE
```

2.1.11 FILLECDA Keyword in the ESD Keyword File

The FILLECDA keyword controls filling ECDA files with random concentrations as they are initialized, rather than filling them with negative or zero concentrations. The syntax of the keyword is the following:

```
FILLECDA  [N1]
```

The value N_1 is the seed for the random number generator. A value between 1 and 999999 must be entered. Every concentration entry in the concentration data files will set to a random value on the (0,1) interval if this keyword is used.

This keyword is useful for generating concentration files for stress testing of the codes. There are no modifiers or quote strings for the FILLECDA keyword. This keyword should not be used during a production run.

2.1.12 GROWTH Keyword in the ESD Keyword File

The GROWTH keyword defines the growth rate equation for water respiring species. These data are used in the biotic transport portion of the MASS2 code as well as the ecological impacts code ECEM. The following is the keyword's syntax:

```
GROWTH  [DELTA=N1] [BETA=N2]
```

Both the delta and beta parameters must be specified. The following is an example of this keyword:

```
GROWTH DELTA=0.002 BETA =0.25
```

A description of the mathematical model for aquatic species is provided in Eslinger et al. (2002). In summary, the growth rate of a given water-respiring species is provided by the regression equation $G = \delta\ w^{-\beta}$ (Thomann et al. 1992) where δ and β are regression parameters and w is the body weight of the species (kg wet weight).

2.1.13 IOONLY Keyword in the ESD Keyword File

The IOONLY keyword instructs the main processor SAC-ESP to only create input files, without running any analysis. The following is the keyword's syntax:

```
IOONLY {COMPUTE}
```

No action is taken if the modifier COMPUTE is not present. The modules for which input files are to be created are specified using the MODULE keyword (see Section 2.1.16).

2.1.14 KDSOIL Keyword in the ESD Keyword File

The KDSOIL keyword is used to enter the definition of a statistical distribution for the solid-aqueous distribution coefficient (K_d) to be used for calculating soil concentrations from groundwater concentrations. The general syntax for the KDSOIL keyword is the following:

```
KDSOIL [“quote1”] [Dist_Index Parameters] {TRUNCATE U1 U2}  {“quote2”}
```

The entry for $quote_1$ must be a unique character string of up to 20 characters that will be used to identify this stochastic variable in subsequent uses. It is case sensitive and embedded spaces are significant. The entry for $quote_2$ is a description for the stochastic variable that can be up to 64 characters long. An entry for $quote_2$ is not required, although it is used for labeling purposes if present. Section 11.0 contains information on the statistical distributions available in the simulation.

A K_d that is triangular on the triple (0.2, 0.5, 0.99) could use the following keyword entry:

```
   KDSOIL “ID#1”  6  0.2  0.5  0.99 “Example Kd for users guide”
```

Note: The data entered by this keyword are used in the river model MASS2 and in the riparian zone model RIPSAC. User instructions for the RIPSAC code are not provided in this document; however, each riverbank location where soil concentrations are to be calculated must point to a KDSOIL keyword by using the ID in $quote_1$ with the KDSOIL modifier on the LOCATION card entered in the RIPSAC keyword file. Multiple locations can point to the same KDSOIL keyword.

2.1.15 LOCATION Keyword in the ESD Keyword File

The LOCATION keyword identifies the locations where concentrations will be generated for use in the impacts modules. The following is the keyword’s syntax:

```
   LOCATION [ID=“quote1”] [EASTING=N1] [NORTHING=N2] {AREA=N3}
      [HUMAN|ECOLOGICAL|ECONOMIC|CULTURAL] {COMPUTE} {POP=N4}
      {GRO_WAT}  {SUR_WAT} {POR_WAT} {SEDIMENT} {SOIL} {SEE_WAT}
      [TITLE= “quote2”]
```

The table below provides the modifiers and associated data for the LOCATION keyword.

Table 2.4 Modifiers Associated with the LOCATION Keyword in the ESD File

Modifier	**Description**
AREA	This entry identifies the area (square meters) associated with the location.
COMPUTE	If the COMPUTE modifier is entered, this location will be included in the simulation. If COMPUTE is not entered, this location will be ignored.
ID	The location identification string is entered using the ID modifier. This string is limited to 15 characters and must be unique. It is used to associate other data with a specific location.
TITLE	A descriptive title for this location is entered using the TITLE modifier. This quote string is limited to 72 characters. It is used only for output labeling purposes.
EASTING	This entry is associated with the easting coordinate for the location. These coordinates are expressed in terms of the Lambert projection of the Washington State Plane North American Datum of 1983, expressed in meters.
NORTHING	This entry is associated with the northing coordinate for the location. These coordinates are expressed in terms of the Lambert projection of the Washington State Plane North American Datum of 1983, expressed in meters.

Modifier	Description
---	One of the four modifiers HUMAN, ECOLOGICAL, ECONOMIC, or CULTURAL must be entered. This modifier identifies the impacts module for which this location is being defined.
GRO_WAT	Presence of this optional modifier indicates that groundwater concentrations will be computed at this location.
SUR_WAT	Presence of this optional modifier indicates that surface water concentrations will be computed at this location.
POR_WAT	Presence of this optional modifier indicates that river bottom pore water concentrations will be computed at this location.
SEE_WAT	Presence of this optional modifier indicates that seep water concentrations (on the land surface) will be computed at this location.
SOIL	Presence of this optional modifier indicates that soil concentrations (on the land surface) will be computed at this location.
SEDIMENT	Presence of this optional modifier indicates that sediment concentrations (on the river bottom) will be computed at this location.

Locations for cultural, ecological, economic, and human analyses may be different. The following example keywords define three locations for human impacts.

```
LOCATION HUMAN ID="HL0151" NAME="Upland human location"
   EASTING =  594737.5  NORTHING=  127827.4  POP=0  GRO_WAT

LOCATION HUMAN ID="HL0417" NAME="Riparian zone human location"
   EASTING =  557375.3  NORTHING=  144885.2  POP=0  GRO_WAT SOIL SEE_WAT

LOCATION HUMAN ID="HL0413" NAME="Richland municipal water intake"
  EASTING =  595445.1  NORTHING=  109753.5  POP=0  SUR_WAT POR_WAT SEDIMENT
```

The following example keywords define a location for economic impacts.

```
LOCATION ID=“Loc #04”  EASTING=123402  NORTHING=12424 COMPUTE
   ECONOMIC GRO_WAT SEE_WAT SOIL
   TITLE=“Example of a title line for location #04”
```

The following example keywords define four locations for ecological impacts.

```
LOCATION ID=“Loc #06”  EASTING=12346  NORTHING=62423 COMPUTE
   ECOLOGICAL TITLE=“Example of a title line for location #06” GRO_WAT

LOCATION ID=“Loc #07”  EASTING=723402  NORTHING=72424 COMPUTE
   ECOLOGICAL GRO_WAT SEE_WAT SOIL
   TITLE=“Example of a title line for location #07”

LOCATION ID=“Loc #08”  EASTING=823400  NORTHING=128423 COMPUTE
   ECOLOGICAL TITLE=“Example of a title line for location #08”
   SUR_WAT  POR_WAT  SEDIMENT
```

```
LOCATION ID="Loc #09"  EASTING=193402  NORTHING=12924 COMPUTE
   ECOLOGICAL GRO_WAT SEE_WAT SOIL
   TITLE="Example of a title line for location #09"
```

The following example keywords define two locations for cultural impacts.

```
LOCATION Cultural ID="CL0001" NAME="Upland cultural location"
   EASTING =  574521.6  NORTHING=  154366.6  AREA= 8.86290E+05
   GRO_WAT

LOCATION Cultural ID="CL3981" NAME="Cultural location in the river"
   EASTING =  593572.8  NORTHING=   66654.6  AREA= 7.96586E+03
   SUR_WAT POR_WAT SEDIMENT
```

2.1.16 MODULE Keyword in the ESD Keyword File

The MODULE keyword identifies the processes to be used in the current run of SAC_ESP. The following is the keyword's syntax:

```
MODULE [ID="quote"] {FLOW|TRAN} {COMPUTE}
```

Multiple MODULE keywords are required to specify execution of more than one module. The module will be computed only if the modifier COMPUTE is present. The table below identifies the modules allowed.

Table 2.5 Module Options for the MODULE Keyword

Quote String	**Description**
INVENTORY	The quote string "INVENTORY" identifies the inventory module.
VADER	The quote string "VADER" identifies the vadose zone release module
STOMP	The quote string "STOMP", when accompanied by the modifier FLOW, identifies a flow-only run for the vadose zone module.
STOMP	The quote string "STOMP", when accompanied by the modifier TRAN, identifies a transport-only run for the vadose zone module.
CFEST	The quote string "CFEST" identifies the groundwater transport module. Groundwater flow runs are performed outside the control of SAC-ESP.
GWDROP	The quote string "GWDROP" identifies the utility program to convert VADER and CFEST outputs into inputs for the river module.
MASS2	The quote string "MASS2" identifies the river flow and transport module.
RIPSAC	The quote string "RIPSAC" identifies the riparian zone concentration module.
ECDA	The quote string "ECDA" identifies the module that sets up the environmental concentration data files for later use by a number of other modules.

The following is a set of keywords that executes all of the available modules:

```
MODULE ID="INVENTORY"       COMPUTE
MODULE ID="VADER"           COMPUTE
MODULE ID="STOMP" FLOW      COMPUTE
MODULE ID="STOMP" TRAN      COMPUTE
MODULE ID="CFEST"           COMPUTE
MODULE ID="GWDROP"          COMPUTE
MODULE ID="MASS2"           COMPUTE
MODULE ID="RIPSAC"          COMPUTE
MODULE ID="ECDA"            COMPUTE
```

2.1.17 OS Keyword in the ESD Keyword File

The OS keyword identifies the operating system under which the current run is being processed. The following is the keyword's syntax:

```
OS [“Unix” | “Windows”]
```

The default operating system is Unix; the other allowable operating system is Windows. To run on a Unix or Linux system, the keyword would appear as follows:

```
OS “Unix”
```

To run on a Windows system, the keyword would appear as follows:

```
OS “Windows”
```

2.1.18 PERIOD Keyword in the ESD Keyword File

The PERIOD keyword identifies the start and stop times for the entire simulation. The following is the keyword's syntax:

PERIOD [START=year$_1$] [STOP=year$_2$] [CLOSURE=year$_3$]

The modifier START and the value year$_1$identify the start of the simulation period. The start of the simulation period must be 1944 or later or the inventory code will error terminate. The modifier STOP and the value year$_2$ identify end of the simulation period. Start and stop years should be entered as whole numbers with the stop year no smaller than the start year. The modifier CLOSURE and the value year$_3$ identify the year that site closure occurs. The year of site closure cannot be smaller than the start year. The following is an example PERIOD keyword that simulates from 1944 through 3050 with site closure occurring at 2050:

```
PERIOD START=1944 STOP=3050 CLOSURE 2050
```

2.1.19 PROCESSOR Keyword in the ESD Keyword File

The PROCESSOR keyword identifies the processors to be used for the current run. The following is the keyword's syntax:

```
PROCESSOR [MACHINE=”quote 1”] [RUNDIR=”quote 2”] {COMPUTE}
```

The main processor SAC-ESP distributes the computational load of the environmental release, flow, and transport portion of the analysis to external processors. These processors can reside on the same computer or be external (networked) to the computer on which SAC-ESP is being run. The COMPUTE modifier must be present for the PROCESSOR keyword to have any effect. The following is an example PROCESSOR keyword:

```
PROCESSOR MACHINE=”c0-0” RUNDIR=”/home/ANALYSIS/assessment” COMPUTE
```

The external processor is accessed by the name specified by the MACHINE qualified. On a Linux system, a job on the processor is invoked as follows:

```
ssh –f c0-0 source /home/ANALYSIS/assessment/processors/p001.com
```

This command tells the external processor (c0-0) to execute (source) the command file (/home/ANALYSIS/assessment/processors/p001.com). The table below shows a sample command file for running VADER and STOMP.

Table 2.6 Sample Command File for Running VADER and STOMP Under Linux

```
#!/bin/csh
# execute vader and stomp for: 20020409143414
cd /home/ANALYSIS/Initial2/
if (-e processors/p001.done ) rm -f processors/p001.done
if (-e processors/p001.fail ) rm -f processors/p001.fail
echo 0 > processors/p001.run
echo 0 > processors/p001.start

cd /home/ANALYSIS/Initial2/vadose/216-N-3/H3/11/
# run vader
if (-e vader.done ) rm -f vader.done
if (-e vader.fail ) rm -f vader.fail
vader.exe /home/ANALYSIS/Initial2/vadose/216-N-3/H3/11
if ( -e vader.fail ) then
   cd /home/ANALYSIS/Initial2/
   echo "vader failed: /home/ANALYSIS/Initial2/vadose/216-N-3/H3/11/" > processors/p001.fail
   goto done
endif

# modify input for AreaX
prestomp.exe input  -1.0000E+00
if ( -e prestomp.fail ) then
   cd /home/ANALYSIS/Initial2/
   echo "prestomp failed: /home/ANALYSIS/Initial2/vadose/216-N-3/H3/11/" >
processors/p001.fail
   goto done
endif

# run stomp
rm -f plot.*
rm -f restart.*
if (-e stomp.done ) rm -f stomp.done
if (-e stomp.fail ) rm -f stomp.fail
stomp.exe > stomp.log
if ( -e stomp.fail ) then
   cd /home/ANALYSIS/Initial2/
   echo "stomp failed: /home/ANALYSIS/Initial2/vadose/216-N-3/H3/11/" > processors/p001.fail
   goto done
endif
rm -f restart.*
```

```
cd /home/ANALYSIS/Initial2/
echo 0 > processors/p001.done
done:
rm -f processors/p001.run
echo done with vadose modeling: vadose/216-N-3/H3/11/
```

The SAC project runs a Linux cluster of 128 Intel Pentium III 1.0-GHz processors. The following are the first 10 of 128 keywords defining the available processors for an analysis in the Initial2 subdirectory:

```
PROCESSOR  MACHINE="c0-0"  RUNDIR="/home/ANALYSIS/Initial2"  compute
PROCESSOR  MACHINE="c0-1"  RUNDIR="/home/ANALYSIS/Initial2"  compute
PROCESSOR  MACHINE="c0-2"  RUNDIR="/home/ANALYSIS/Initial2"  compute
PROCESSOR  MACHINE="c0-3"  RUNDIR="/home/ANALYSIS/Initial2"  compute
PROCESSOR  MACHINE="c0-4"  RUNDIR="/home/ANALYSIS/Initial2"  compute
PROCESSOR  MACHINE="c0-5"  RUNDIR="/home/ANALYSIS/Initial2"  compute
PROCESSOR  MACHINE="c0-6"  RUNDIR="/home/ANALYSIS/Initial2"  compute
PROCESSOR  MACHINE="c0-7"  RUNDIR="/home/ANALYSIS/Initial2"  compute
PROCESSOR  MACHINE="c0-8"  RUNDIR="/home/ANALYSIS/Initial2"  compute
PROCESSOR  MACHINE="c0-9"  RUNDIR="/home/ANALYSIS/Initial2"  compute
```

2.1.20 REALIZAT Keyword in the ESD Keyword File

The REALIZAT keyword identifies the number of realizations to be simulated. The following is the keyword’s syntax:

REALIZAT N_1

The number of realizations is given by the single numerical entry N_1. The valid number of realizations is 1 to 9999. Run times and disk storage requirements are directly proportional to the number of realizations. The following is an example REALIZAT keyword requesting 25 realizations:

```
REALIZAT 25
```

2.1.21 REMEDIAT Keyword in the ESD File

The REMEDIAT keyword identifies remediation actions between waste sites. These actions are computed by VADER and STOMP. The following is the REMEDIATION keyword’s syntax:

```
REMEDIAT {YEAR=N1} [FROM=“quote1”] [TO=“quote2”] [DEPTH=N1]
  {SOIL=N2} {CAKE=N3} {RIVER=N4} {CEMENT=N5} {CORE=N6} {LIQUID=N7}
```

VADER always remediates waste in the VADER working site inventory and send it to another site. In contrast, the STOMP code remediates soil from the vadose zone and send it to another site. This creates the possibility of transfers to sites of quantities from previous runs of both VADER and STOMP. Table 2.7 describes the modifiers for the REMEDIAT keyword for remedial action exports. One or more waste forms may be set for exports. The export fractions for unspecified waste forms are set to 0.

Table 2.7 REMEDIAT Keyword Modifiers

Modifier	Description
YEAR	The numerical value associated with the YEAR modifier identifies the year in which the remediation action takes place. A separate keyword must be entered for each year a remediation event occurs.
FROM	The quote string associated with the FROM modifier identifies the release site from which waste or soil will be exported.
TO	The quote string associated with the TO modifier identifies the site to which material will be exported. The FROM and TO sites must be different.
DEPTH	The numerical value associated with the DEPTH modifier identifies the depth (meters) of soil to be excavated from the FROM site by STOMP and moved to the TO site.
SOIL	The numerical value associated with the SOIL modifier identifies the fraction of soil-debris waste to be exported by VADER. Valid values are in the range 0 to 1.
CAKE	The numerical value associated with the CAKE modifier identifies the fraction of salt cake/sludge waste to be exported by VADER. Valid values are in the range 0 to 1. The SLUDGE and SALT modifiers can be used in place of the CAKE modifier.
CMNT	The numerical value associated with the CMNT modifier identifies the fraction of cement waste to be exported by VADER. Valid values are in the range 0 to 1. The CEMENT modifier can be used in place of the CMNT modifier.
CORE	The numerical value associated with the CORE modifier identifies the fraction of the remaining reactor/component waste to be exported by VADER. Valid values are in the range 0 to 1. The REACTOR modifier can be used in place of the CORE modifier.
LIQUID	The numerical value associated with the LIQUID modifier identifies the fraction of remaining liquid waste to be exported by VADER. Valid values are in the range 0 to 1.
RIVER	The numerical value associated with the RIVER modifier identifies the fraction of remaining liquid waste to be exported by VADER. Valid values are in the range 0 to 1.

The following example REMEDIAT keyword causes VADER to export portions of the SOIL, CAKE, and CMNT wastes from the site MEMFIS to the site Kairo in the year 1960:

```
REMEDIAT YEAR=1960 FROM="MEMFIS" TO="Kairo" SOIL=.11 CAKE=.28 CMNT=.15
```

The following example REMEDIAT keyword causes STOMP to excavate the top 3 meters of soil at the site MEMFIS and send it to the site Kairo in the year 1965:

```
REMEDIAT YEAR=1965 FROM="MEMFIS" TO="Kairo" DEPTH=3
```

2.1.22 REPORT Keyword in the ESD Keyword File

The REPORT keyword defines the file that will contain diagnostic messages and log the progress for the current run of SAC-ESP. The following is the keyword's syntax:

```
REPORT [“quote”]
```

The REPORT keyword must be the first keyword in the file. There can be comment lines before this keyword, but if it is not the first keyword, then the program will error terminate. The name of the report file is entered in a quote string. File names up to 72 characters long are supported, and path names can be optionally included. The following is an example REPORT keyword:

```
REPORT “/home/ANALYSIS/Initial2/ESD_Initial2.rpt”
```

Programs other than SAC-ESP do not use the REPORT keyword in the ESD keyword file.

2.1.23 RESITE Keyword in the ESD Keyword File

The RESITE keyword defines a subset of the sites defined by the AGGREGATE keywords (see Section 2.1.1) to be processed. The following is the keyword’s syntax:

```
RESITE [“site1”] {“site2”} ... {“siten”} {COMPUTE}
```

No action will be taken for the RESITE keyword unless the modifier COMPUTE is present. One or more sites can be identified on the RESITE keyword. The site IDs from the AGGREGATE keyword are the source of data for entry in the quote strings “site_1”, “site_2”, etc. The following RESITE keyword identifies four sites to be calculated in the run of SAC-ESP:

```
RESITE "216-B-14" "241-A-102" "UPR-300-@R6-4" "US_Ecology" COMPUTE
```

2.1.24 RESPIRE Keyword in the ESD Keyword File

The RESPIRE keyword defines the oxygen respiration for animal species. These data are used in the biotic transport portion of the MASS2 code as well as the ecological impacts code ECEM. The following is the keyword’s syntax:

```
RESPIRE [GAMMA=N1] [PHI=N2]
```

Both the gamma and phi parameters must be specified. The following is an example of this keyword:

```
RESPIRE GAMMA=0.2 PHI=0.032
```

Eslinger et al. (2002) describe the mathematical model for oxygen respiration of animal species. In summary, the oxygen respiration rate for a given species is calculated from the equation $\rho = \phi\ w^{-\gamma}$, where ϕ and γ are regression parameters (Thomann 1989) and w is the body weight of the species (kg wet weight).

2.1.25 RESTART Keyword in the ESD Keyword File

The RESTART keyword defines a subset of the realizations defined by the REALIZATION keyword (see Section 2.1.20) to be processed. The following is the keyword’s syntax:

```
RESTART [real1] {real2} … {realn} {COMPUTE }
```

This keyword has no effect unless the COMPUTE modifier is present. The following is an example keyword that identifies that realizations 1, 17, and 24 be computed:

```
RESTART 1 17 24
```

The realization numbers can be entered in any order.

2.1.26 SPECIE Keyword in the ESD Keyword File

The SPECIE keyword is used to enter definitions for possible species to be simulated. The ECEM, TCERM, and MASS2 codes can use species from this master list. The following is this keyword's syntax:

```
SPECIE [ID="quote 1"] [TYPE="quote 2"] [NAME="quote 3"]
   {Modifier=N1} … {Modifier=N36} {EMERGENT}
```

A separate SPECIE keyword must be entered for every species being defined. Table 2.8 describes the modifiers associated with the SPECIE keyword.

Table 2.8 Modifiers Associated with the SPECIE Keyword in the ESD File

Modifier	Description
ID	The quote string associated with the ID modifier is a unique species identification string up to six characters in length. The analyte identification string is case sensitive, and spaces or hyphens change the definition.
TYPE	The quote string associated with the TYPE modifier string is a two-character analyte type indicator. The following are the valid entries for this string: TA – if the species is a terrestrial animal TP – if the species is a terrestrial plant QA – if the species is an aquatic animal QP – if the species is an aquatic plant
NAME	The quote string associated with the NAME modifier is a species name or description up to 72 characters in length.
AE	The numerical entry associated with the AE modifier is the assimilation efficiency of the species. This value is unitless. Entry of this modifier is optional. If it is not present, the value of AE defaults to zero.
AWD	The numerical entry associated with the AWD modifier is the wet-to-dry weight ratio of the species. This value has units of g wet/g dry. Entry of this modifier is optional. If it is not present, the value of AWD defaults to zero.

Modifier	**Description**
DIFFHT	The numerical entry associated with the DIFFHT modifier is the diffusion height of the species. This value has units of meters. Entry of this modifier is optional. If it is not present, the value of DIFFHT defaults to zero.
EMERGENT	There is no numerical entry associated with the EMERGENT modifier. Use of this modifier indicates that the species is an emergent plant. An emergent plant is one that has roots in water but grows in the air. This type of plant does not have rain splash of contaminated soil onto the plant's leaves.
ETWATER	The numerical entry associated with the ETWATER modifier is the exposure time to water of the species. This value has units of hr/day. Entry of this modifier is optional. If it is not present, the value of ETWATER defaults to zero.
FLIPID	The numerical entry associated with the FLIPID modifier is the fraction lipid of the species. This value has units of g lipid/g wet. Entry of this modifier is optional. If it is not present, the value of FLIPID defaults to zero.
FMR	The numerical entry associated with the FMR modifier is the metabolic rate of predator species. This value has units of kcal/day. Entry of this modifier is optional. If it is not present, the value of FMR defaults to zero.
FOC	The numerical entry associated with the FOC modifier is the fraction organic carbon of the species. This value has units of g organic carbon/g dry weight. Entry of this modifier is optional. If it is not present, the value of FOC defaults to zero.
FPA	The numerical entry associated with the FPA modifier is the volume fraction of plant tissue that is air for the species. This value is unitless. Entry of this modifier is optional. If it is not present, the value of FPA defaults to zero.
FPL	The numerical entry associated with the FPL modifier is the volume fraction of plant tissue that is lipid for the species. This value is unitless. Entry of this modifier is optional. If it is not present, the value of FPL defaults to zero.
FPW	The numerical entry associated with the FPW modifier is the volume fraction of plant tissue that is water for the species. This value is unitless. Entry of this modifier is optional. If it is not present, the value of FPW defaults to zero.
FW	The numerical entry associated with the FW modifier is the water weight fraction of plant tissue for the species. This value is unitless. Entry of this modifier is optional. If it is not present, the value of FW defaults to zero.
FWATER	The numerical entry associated with the FWATER modifier is the fraction exposure to water for the species. This value is unitless. Entry of this modifier is optional. If it is not present, the value of FWATER defaults to zero.
FDW	The numerical entry associated with the FDW modifier is the conversion from dry weight to wet weight for the species. This value has units of kg dry/kg wet. Entry of this modifier is optional. If it is not present, the value of FDW defaults to zero.

Modifier	Description
GE	The numerical entry associated with the GE modifier is the gross energy for the species. This value has units of kcal/kg wet weight. Entry of this modifier is optional. If it is not present, the value of GE defaults to zero.
INHRATE	The numerical entry associated with the INHRATE modifier is the resting inhalation rate for the species. This value has units of m^3/day. Entry of this modifier is optional. If it is not present, the value of INHRATE defaults to zero.
OCAR	The numerical entry associated with the OCAR modifier is the organic carbon assimilation rate for the species. This value has units of g organic carbon assimilated/g ingested. Entry of this modifier is optional. If it is not present, the value of OCAR defaults to zero.
PCS	The numerical entry associated with the PCS modifier is the fraction of surface area in contact with soil for the species. This value has units of 1/day. Entry of this modifier is optional. If it is not present, the value of PCS defaults to zero.
PCW	The numerical entry associated with the PCW modifier is the fraction of surface area available to water soil for the species. This value is unitless. Entry of this modifier is optional. If it is not present, the value of PCW defaults to zero.
PSI	The numerical entry associated with the PSI modifier is the seasonality factor for the species. This value is unitless. Entry of this modifier is optional. If it is not present, the value of PSI defaults to zero.
RADIUS	The numerical entry associated with the RADIUS modifier is the radius of the species. This value has units of cm. Entry of this modifier is optional. If it is not present, the value of RADIUS defaults to zero.
RHOP	The numerical entry associated with the RHOP modifier is the plant tissue density of the species. This value has units of kg/m^3. Entry of this modifier is optional. If it is not present, the value of RHOP defaults to zero.
SA	The numerical entry associated with the SA modifier is the surface area of the species. This value has units of cm^2. Entry of this modifier is optional. If it is not present, the value of SA defaults to zero.
SADHER	The numerical entry associated with the SADHER modifier is the skin adherence factor for the species. This value has units of mg/cm^2. Entry of this modifier is optional. If it is not present, the value of SADHER defaults to zero.
SEDING	The numerical entry associated with the SEDING modifier is the fraction of diet that is sediment for the species. This value is unitless. Entry of this modifier is optional. If it is not present, the value of SEDING defaults to zero.
SOILING	The numerical entry associated with the SOILING modifier is the soil ingestion rate for the species. This value has units of kg soil ingested/kg dry weight. Entry of this modifier is optional. If it is not present, the value of SOILING defaults to zero.

Modifier	Description
THETA	The numerical entry associated with the THETA modifier is the area use factor for the species. This value is unitless. Entry of this modifier is optional. If it is not present, the value of THETA defaults to zero.
WATERING	The numerical entry associated with the WATERING modifier is the water ingestion rate for the species. This value has units of liters/day. Entry of this modifier is optional. If it is not present, the value of WATERING defaults to zero.
WBMASS	The numerical entry associated with the WBMASS modifier is the wet body mass for the species. This value has units of grams. Entry of this modifier is optional. If it is not present, the value of WBMASS defaults to zero.
WEIGHT	The numerical entry associated with the WEIGHT modifier is the wet body weight for the species. This value has units of kg (wet). Entry of this modifier is optional. If it is not present, the value of WEIGHT defaults to zero.

The following SPECIE keywords define the American Coot and Channel Catfish as species available for simulation:

```
SPECIE ID=“AMCOOT” TPYE= “TA” NAME=“American coot”
   AE=0.81  DIFFHT=0.2  ETWATER=20 FMR=163.0867887 FWATER=0.5 FDW=0.17
   GE=1900   INHRATE=.6159264 PCS=0.25 PCW=0.5 PSI=1 RADIUS=5
   SA=784.0626481 SADHER=1.45   SOILING=0.18 THETA=1
   WATERING=0.046102353 WEIGHT=0.692

SPECIE ID=“CHCATF” TPYE= “QA” NAME=“Channel catfish”
   AE=0.85  AWD=5 FLIPID=0.178  FOC=0.45  GE=1200  OCAR=0.8  RADIUS=10
   SEDING=0.19 WBMASS=6802.5
```

2.1.27 TIMES Keyword in the ESD Keyword File

The TIMES keyword identifies the times at which concentration data will be generated to support the calculations by each of the impacts codes. The following is the keyword’s syntax:

```
TIMES [HUMAN | ECOLOGICAL | ECONOMIC | CULTURAL] [T1] {T2} … {Tn}
```

The numerical entries T_1, T_2, …, T_n are the times (whole number years) when outputs are desired. Multiple TIMES keywords can be entered. Only one of the modifiers HUMAN, ECOLOGICAL, ECONOMIC, and CULTURAL is allowed on an individual keyword entry.

The following is an example TIMES keyword that requests concentration data for the three years 2020, 2075, and 3014 for the cultural impacts module:

```
TIMES CULTURAL 2020 2075 3014
```

The following are example time definitions for each of the four impacts module:

```
TIMES CULTURAL 2023 2075 3014
TIMES ECONOMIC 2020 2075
```

```
   TIMES ECOLOGICAL 2010 2015 2020 2031 2075 3015 3050
   TIMES HUMAN 2022 2072 3215 3014
```

2.1.28 TITLE Keyword in the ESD Keyword File

The TITLE keyword is used to define a single-line problem title for the ESD run. The problem title will be written to output files. If the title is not supplied, the program will error terminate. The following is this keyword's syntax:

```
   TITLE ["quote"]
```

The title is entered in a quote string, which must be enclosed in double quotes. Titles up to 72 characters long are supported. The following example defines a title for a run of the code.

```
   TITLE "Example title line for the environmental settings control file."
```

There are no modifiers associated with the TITLE keyword.

2.1.29 USER Keyword in the ESD Keyword File

The USER keyword is used to identify the user of the SAC-ESP program. The user name will be written to output files. Some of the other programs that read the ESD keyword file require the USER keyword to be present, but not all of them do. For example, the USER keyword is not required by INVENTORY, CRDROP, or CRDROP_INDEX. However, ECDA, HEADER, RIPSAC, and all of the impacts codes will error terminate if the USER keyword is not present. The following is this keyword record's syntax:

```
   USER [“quote”]
```

The user name is entered in a quote string, which must be enclosed in double quotation marks. User names up to 16 characters long are supported. The following example defines John Q. Public as the user running the code.

```
   USER “John Q. Public”
```

There are no modifiers associated with the USER keyword.

2.2 Environmental Concentration Data Accumulator

The purpose of the Environmental Concentration Data Accumulator is to provide a central storage for all concentrations of analytes at environmental locations and times needed to perform impacts calculations.

2.2.1 Format of the ECDA Files

The general approach and file structure for the ECDA files have the following characteristics:

- Data storage for different analytes and different impacts types are provided in separate files (9 analytes and 4 impacts types would result in 36 concentration files). This design feature allows addition of an analyte to a data set without recalculation of the other analytes.
- The files have a binary, fixed record length, direct-access format.
- A mapping scheme is used to store only actual data (no placeholders with wasted storage space).

Each data record in a concentration data file contains the following information: Year, Location ID, Media ID, Concentration data (realizations 1 to the maximum). The order in which media information appear in the file is always the following:

- GW: concentrations in groundwater (Ci/m^3 or kg/m^3)
- SP: concentrations in seep water (Ci/m^3 or kg/m^3)
- SO: concentrations in soil (land surface) (Ci/kg$_{\text{soil}}$ or kg$_{\text{analyte}}$/kg$_{\text{soil}}$)
- SW: concentrations in surface water (river) (Ci/m^3 or kg/m^3)
- PW: concentrations in river bottom pore water (Ci/m^3 or kg/m^3)
- SD: concentrations in river bottom sediment (Ci/kg$_{\text{sediment}}$ or kg$_{\text{analyte}}$/kg$_{\text{sediment}}$).

Table 2.9 provides an overview of the structure of a concentration data file for human impacts assuming that 25 realizations were used. The structure for this file is identical for all impacts types.

Table 2.9 Structure of a Concentration Data File

Analyte ID	**Human**	
Units for the 6 media as character strings		
Time 1	Location 1	Media L1-1, 25 realizations
		…
		Media L1-L1MAX, 25 realiz.
	Location 2	Media L2-1, 25 realizations
		…
		Media L2-L2MAX, 25 realiz.
	…	…
	Max Human Locations	Media LL-1, 25 realizations
		…
		Media LL-LLMAX, 25 realiz.
Time 2	Location 1	Media L1-1, 25 realizations
		…
		Media L1-L1MAX, 25 realiz.
	Location 2	Media L2-1, 25 realizations
		…
		Media L2-L2MAX, 25 realiz.
	…	…
	Max Human Locations	Media LL-1, 25 realizations
		…
		Media LL-LLMAX, 25 realiz.

…	…	…
Max Time	Location 1	Media L1-1, 25 realizations
Human		…
		Media L1-L1MAX, 25 realiz.
	Location 2	Media L2-1, 25 realizations
		…
		Media L2-L2MAX, 25 realiz.
	…	…
	Max Human Locations	Media LL-1, 25 realizations
		…
		Media LL-LLMAX, 25 realiz.

2.2.2 Format of the Record Map File for Concentration Data

The record map file is an ASCII file that provides indexing information for the storage locations in the binary concentration data files. There is one record map file per impact type because the structure for the concentration data for a given impact type (human, ecological, cultural, or economic) is identical for all analytes. This file is generated by the ECDA code. The user should never modify this file.

The record map file has several header lines that are followed by indexing data. All character data are enclosed in double quotation marks, and records that contain more than one value have the data separated by commas. The definition of the header lines is as follows:

- **Problem title:** The title from the ESD keyword file.
- **Code Name:** The name of the code that generated the file
- **Code Version:** The version number of the code that generated the file
- **Code Date:** The modification date of the code that generated the file
- **User Name:** The user name from the ESD keyword file
- **Run ID:** The run ID from the code run that generated the file
- **Record Length:** Record length for the binary data file
- **Number of times:** The number of solution times requested for this impact type
- **List of times:** A list of solution times for this impact type, with one time per line
- **Number of locations:** The number of locations requested for this impact type
- **List of locations:** A list of locations IDs, with one ID per line
- **Number of media:** The number of media where contaminated data may be saved
- **List of media:** A list of the media IDs, with one value per line

The header data are followed by record map index data. There are as many lines of index data as there are locations. The index data for locations come in the same order as the location IDs provided earlier in the file. Each line of index data contains the location ID followed by as many indices as there are media types. The indices give an offset record number. If a particular media is not saved at a location, then the index value is set to negative 1. Table 2.10 provides an example record map file for a very small problem.

Table 2.10 Example Record Index Map File

```
"ECOLOGICAL: Hanford Solid Waste EIS - 10,000 Years"
"ECDA"
"1.00.D.3"
" 9 Mar 2001"
"Paul W. Eslinger"
"20020225135708"
18,"Records in a time block"
16,"Record Length"
2,"Number of times"
8000
10000
6,"Number of locations"
"EL0001"
"EL0002"
"EL0003"
"EL0004"
"EL0005"
"EL0006"
6,"Number of media"
"GW"
"SP"
"SO"
"SW"
"PW"
"SD"
"EL0001",8,9,10,-1,-1,-1
"EL0002",-1,-1,-1,11,12,13
"EL0003",14,15,16,-1,-1,-1
"EL0004",-1,-1,-1,17,18,19
"EL0005",20,21,22,-1,-1,-1
"EL0006",-1,-1,-1,23,24,25
```

2.3 Shared Environmental Stochastic Data

Some stochastic data are shared among several programs, including that defined by the KDSOIL (Section 2.1.14) and DILUTE (Section 2.1.7) keywords in the ESD keyword file. The following sections provide descriptions of the format of the data files containing the generated stochastic data.

2.3.1 Format of the DILUTE Data File

Entry of one or more DILUTE keywords in the ESD keyword file (see Section 2.1.7) causes the ECDA code to generate a data file of DILUTE values for use in other codes. The table below provides an example of this file for one DILUTE keyword and two realizations. The file is a text file that starts with seven header lines. The entry on line 8 is the number of DILUTE definitions – 1 in this example. The next line contains the number of realizations – 2 in this example. Each succeeding line contains data for a single DILUTE definition in the form of index, identification string, and generated values for each realization. Multiple data on a single line are separated by commas, and all text data are enclosed in double quotation marks.

Table 2.11 Example DILUTE File

```
DILUTE"
"SAC Rev. 0 Shakedown2 Assessment"
"ECDA"
"1.00.D.3"
" 9 Mar 2001"
"20020314153720"
"Engel, Eslinger,"
1
2
1,"DF5m", 4.87928E-01, 4.61883E-01
```

2.3.2 Format of the KDSOIL Data File

Entry of one or more KDSOIL keywords in the ESD keyword file (see Section 2.1.14) causes the ECDA code to generate a data file of KDSOIL values for use in other codes. Table 2.12 provides an example of this file for 19 KDSOIL keywords and two realizations. The file is a text file that starts with seven header lines. The entry on line 8 is the number of KDSOIL definitions – 19 in this example. The next line contains the number of realizations – 2 in this example. Each succeeding line contains data for a single KDSOIL definition in the form of index, identification string, and generated values for each realization. Multiple data on a single line are separated by commas, and all text data are enclosed in double quotation marks.

Table 2.12 Example KDSOIL File

```
"KDSOIL"
"SAC Rev. 0 Shakedown2 Assessment"
"ECDA"
"1.00.D.3"
" 9 Mar 2001"
"20020314153720"
"Engel, Eslinger,"
19
2
1,"KDH", 0.00000E+00, 0.00000E+00
2,"KDCCl4", 3.63214E-04, 1.64047E-04
3,"KDPu", 4.54387E-01, 1.22489E+00
4,"KDU", 4.67235E-04, 1.05881E-03
5,"KDSr", 1.63468E-02, 2.80513E-02
6,"KDCs", 6.83420E-01, 9.52710E+00
7,"KDI", 1.39666E-03, 5.42975E-04
8,"KDCr", 0.00000E+00, 7.06699E-05
9,"KDTc", 0.00000E+00, 7.03645E-06
10,"KDZero", 1.00000E-10, 1.00000E-10
11,"KDSUSH", 1.00000E-10, 1.00000E-10
12,"KDSUSCr", 4.72011E-04, 2.20753E-04
13,"KDSUSCs", 1.52260E+01, 4.32762E+01
14,"KDSUSI", 8.94922E-03, 1.25821E-03
15,"KDSUSPu", 2.12136E+01, 2.09212E+00
16,"KDSUSSr", 2.41585E-01, 9.20855E-02
17,"KDSUSTc", 1.78929E-04, 5.34989E-05
18,"KDSUSU", 7.56287E-03, 1.50449E-02
19,"KDSUSCCl4", 1.18195E-03, 4.08758E-04
```

3.0 ESP – Environmental Stochastic Preprocessor

3.1 Code Purpose

All of the environmental codes shown on the left side of Figure 1-2 are run as deterministic models except INVENTORY and RIPSAC. This means that they require a single set of inputs and produce a single set of outputs for every realization. In order to model this system in a stochastic framework, ESP was developed to prepare inputs for each code, create a set of input files for each realization, and repeatedly execute the codes.

The INVENTORY and RIPSAC codes were developed as standalone stochastic codes, which generate all of their own input variables. The INVENTORY module simulates all of the waste to be released and typically will be executed outside of the ESP framework before the ESP is run.

The ESP has been developed to run under a Linux system (Linux Red Hat 7.0). Many of the other codes shown in Figure 1-2 have been developed to run under either a Windows environment or a Unix environment. However, the ESP has been optimized for the operating system it is running under and thus was not developed to run under both systems.

ESP is, by design, extremely flexible in how the vadose zone simulations are handled. The complexity of simulating the vadose zone at hundreds of individual sites in a SAC assessment made it necessary to make the ESP code able to handle everything from the rerun of a single VADER or STOMP simulation to a simulation of all sites for all realizations and all analytes. Consequently, the description of the control functions provided here is not exhaustive; rather, it provides some basic guidelines to managing calculations in SAC and illustrates typical approaches.

3.2 Algorithms and Assumptions

ESP serves three basic purposes: simulate the stochastic input variables for the deterministic codes, create input files for each code, and run the codes in the proper sequence. The simulations are done using the statistical routines described in Section 11.0.

During an analysis (which will be referred to as an assessment throughout this section), ESP executes the different modules for all combinations of waste sites, analytes, and realizations. To keep all of these pieces in some sort of order, a directory structure is developed for the assessment. The user will create a master directory (e.g., /Disk1/Projects/MasterRun, which is referred to as …/assessment in this document). ESP is then used to create subdirectories under the main directory, as shown below:

```
.../assessment/cfest/analyte/realization.../assessment/ecda
.../assessment/inventory
.../assessment/mass2/analyte/realization
.../assessment/processors
.../assessment/ripsac
.../assessment/vadose/site/analyte/realization
```

The boldface names in the directory path refer to subdirectories created by the ESP that are fixed names (the user cannot modify these names). The non-boldface names refer to subdirectories whose names are based on entries in the ESD keyword file (see Section 2.1). The variable subdirectories (site, analyte, and realization) represent a potentially large number of lower-level subdirectories. For example, if the assessment has two sites (216-H-8 and memfis), two analytes (H3 and nasty) and two realizations, then the vadose subdirectory structure will contain the following entries:

```
.../assessment/vadose/216-H-8/H3/1
.../assessment/vadose/216-H-8/H3/2
.../assessment/vadose/216-H-8/nasty/1
.../assessment/vadose/216-H-8/nasty/2
.../assessment/vadose/memfis/H3/1
.../assessment/vadose/memfis/H3/2
.../assessment/vadose/memfis/nasty/1
.../assessment/vadose/memfis/nasty/2
```

The number of characters that comprise the realization subdirectory name is a function of the total number of realizations. The number is equal to the number of places in the total number of realizations, with leading zeros inserted as needed. For instance, if there were 150 realizations, then the subdirectory for memfis, H3, and the second realization would look like the following:

```
.../assessment/vadose/memfis/H3/002
```

ESP will execute the modules VADER, STOMP, CFEST, and MASS2 once for every lower-level subdirectory. In the example above, both VADER and STOMP will be executed eight times ({number of sites} x {number of analytes} x {number of realizations}), and CFEST and MASS2 will be executed four times ({number of analytes} x {number of realizations}).

ESP will execute INVENTORY only once. The INVENTORY module can also be executed in a standalone mode external to control by ESP. The RIPSAC module only requires one execution. It can be controlled by the ESP or executed in a standalone mode. The ecda subdirectory will contain the ECDA database files. The ECDA module gets called once by ESP to create this database.

ESP was developed to use distributed processing. Each execution of a module (VADER or STOMP, for example) will be run on an external process. ESP will distribute the runs based on priority – completion of other modules – and the list of processors described in the PROCESSORS keyword in the ESD file (see Section 2.1.19).

3.2.1 ESP Major Functions

ESP is the central processor of the environmental modules. As such, ESP is the main program for scheduling models of inventor, release, and transport in groundwater and the Columbia River. The following is a typical sequence for performing a Hanford plus background analysis:

1. Create the ESD keyword file.

2. Create the directory structure (run ESP with the CREATEDIR keyword enabled, and all of MODULE, ANALYTE, and AGGREGATE keywords enabled). These keywords are enabled by entering a COMPUTE modifier on the keyword line.

3. Run the INVENTORY module (external) and ensure that the inventory result files (inv1.res, inv2.res, …) are located in the inventory subdirectory (…/assessment/inventory).

4. Create the input template and stochastic definition files for all the modules and copy them to the appropriate subdirectory. Most of the input files have a specified name. In the following list of required files, the files that have a variable name appear in italics:

   - …/assessment/inventory/inv###.dat – realization-specific INVENTORY result files (filename is based on total number of realizations, e.g., inv5.dat could be realization 5 of 9, inv05.dat could be realization 5 of 25, and inv005.dat could be realization 5 of 150)
   - …/assessment/vadose/stochastic_stomp.key – STOMP stochastic definitions
   - …/assessment/vadose/site/stochastic_vader.key – site specific VADER stochastic definitions
   - …/assessment/vadose/site/template_stomp.key – site specific STOMP template input file
   - …/assessment/vadose/site/template_vader.key – site specific VADER template input file
   - …/assessment/cfest/stochastic.key – CFEST stochastic definitions
   - …/assessment/cfest/analyte/cfest.key – analyte specific CFEST template input file
   - …/assessment/cfest/analyte/cfest.l3i – analyte specific CFEST template input file
   - …/assessment/cfest/analyte/cfest.lp1 – analyte specific CFEST template input file
   - …/assessment/cfest/analyte/input.hbc – analyte specific CFEST template input file
   - …/assessment/cfest/analyte/ZTOP.DAT – analyte specific CFEST template input file
   - …/assessment/cfest/analyte/bincf – subdirectory that needs to contain the flow field calculated from a separate run of CFEST
   - …/assessment/cfest/analyte/cf_tmpbinary – subdirectory that needs to contain the flow field calculated from a separate run of CFEST
   - …/assessment/mass2/biota.stoch – MASS2 stochastic definitions for biotic transport
   - …/assessment/mass2/mass2.key – MASS2 stochastic definitions (Kds)
   - …/assessment/mass2/index.key – input definition file for CRDROP_INDEX (create cross index for MASS2 grid to the ECDA locations)
   - *…/assessment/mass2/CRDROP_grid.dat* – MASS2 grid file (specified in the index.key file)
   - …/assessment/mass2/gwdrop.key – GWDROP template input file
   - *…/assessment/mass2/col-river-elem.dat* – GWDROP river elements (specified in gwdrop.key)
   - *…/assessment/mass2/col-river-node.dat* – GWDROP river nodes (specified in gwdrop.key)
   - *…/assessment/mass2/hanfnad83m-pt-000* – GWDROP river cells (specified in gwdrop.key)
   - …/assessment/ripsac/ripsac.key – RIPSAC control and stochastic definitions

5. Create the ECDA database and the following realization-specific input files (run ESP with CREATEDIR keyword without COMPUTE, IOONLY keyword enabled, and all of the MODULE, ANALYTE, and AGGREGATE keywords enabled):

- …/assessment/vadose/site/analyte/realization/vader.key
- …/assessment/vadose/site/analyte/realization/input-esp
- …/assessment/cfest/analyte/realization/cfest.ctl
- …/assessment/mass2/analyte/realization/biota.key
- …/assessment/mass2/analyte/realization/realize.dat
- …/assessment/mass2/analyte/realization/cfest/gwdrop.esp.

6. Model the release and transport through the vadose zone (run ESP, disable IOONLY keyword, enable and modify the RESTART keyword to run the realizations of interest, enable only the VADER, STOMP Flow, and STOMP Transport modules of the MODULES keyword), creating the following result files:

   - …/assessment/vadose/site/analyte/realization/vader.table
   - …/assessment/vadose/site/analyte/realization/input
   - …/assessment/vadose/site/analyte/realization/release.

7. Check the vadose results (run VZGRAB for all realizations).

8. If errors occurred in vadose modeling, fix and rerun the problem runs (run ESP, enable and modify the RESITE keyword to include the sites to rerun, modify the RESTART keyword to only rerun needed realizations, and disable ANALYTE keywords for analytes not needed to rerun).

9. Model the transport through the groundwater (run ESP, disable RESITE keyword, enable realizations of interest on the RESTART keyword, enable only the CFEST and GWDROP modules of the MODULES keyword, and enable all ANALYTE keywords), creating the following result files:

   - …/assessment/cfest/analyte/realization/hheldm001.tab
   - …/assessment/cfest/analyte/realization/hheldq001.tab
   - …/assessment/mass2/analyte/realization/cfest/COV-000.DAT
   - …/assessment/mass2/analyte/realization/cfest/TMS-node-FLOW.DAT
   - …/assessment/mass2/analyte/realization/cfest/TMS-node-analyte.DAT.

10. Check the groundwater results (run GRGRAB for all realizations).

11. If errors occurred in groundwater modeling, fix and rerun the problem runs (run ESP, modify the RESTART keyword to only rerun needed realizations, and disable ANALYTE keywords for analytes not needed to rerun).

12. Model the river transport (run ESP, enable realizations of interest in the RESTART keyword, enable only the MASS2 module of the MODULES keyword), creating the following result files:

    - …/assessment/mass2/analyte/realization/crdrop/analyte_realization.dat.

13. Model the river shore (run ESP, enable realizations of interest in the RESTART keyword, enable only the RIPSAC module of the MODULES keyword).

The following is a typical background-only analysis:

1. Create a new main directory (…/assessment/bg).

2. Create the ESD keyword file.

3. Create the directory structure (run ESP, enable CREATEONLY and BACKGROUND keywords, enable the ECDA, MASS2, and RIPSAC modules of the MODULES keyword).

4. Create the input template and stochastic files:

   - …/assessment/bg/mass2/biota.stoch – MASS2 stochastic definitions for biotic transport
   - …/assessment/bg/mass2/mass2.key – MASS2 stochastic definitions (Kds)
   - …/assessment/bg/mass2/index.key – input definition file for CRDROP_INDEX (create cross index for MASS2 grid to the ECDA locations)
   - *…/assessment/bg/mass2/CRDROP_grid.dat* – MASS2 grid file (specified in the index.key file)
   - …/assessment/bg/mass2/gwdrop.key – GWDROP template input file, specify an existing directory containing an existing CFEST results file (i.e., hheldm001.tab) for the CFDIRECT keyword (e.g., …/assessment/cfest/Cs137/01)
   - *…/assessment/bg/mass2/col-river-elem.dat* – GWDROP river elements (specified in gwdrop.key)
   - *…/assessment/bg/mass2/col-river-node.dat* – GWDROP river nodes (specified in gwdrop.key)
   - *…/assessment/bg/mass2/hanfnad83m-pt-000* – GWDROP river cells (specified in gwdrop.key)
   - …/assessment/bg/ripsac/ripsac.key – RIPSAC stochastic definitions.

5. Create the input files and ECDA database (run ESP, disable CREATEONLY keyword, enable the IOONLY keyword, enable the ECDA, GWDROP, and MASS2 modules of the MODULES keyword).

6. Create the MASS2 node files (run ESP, enable realizations of interest in the RESTART keyword, enable only the GWDROP module of the MODULES keyword).

7. Model the river transport (run ESP, enable realizations of interest in the RESTART keyword, enable only the MASS2 module of the MODULES keyword).

8. Model the river shore (run ESP, enable realizations of interest in the RESTART keyword, enable only the RIPSAC module of the MODULES keyword).

The main difference in the background-only modeling is with the GWDROP module. Because the groundwater module isn't run in this case, previously run CFEST result files (hheldq001.tab and hheldm001.tab) will be needed. ESP will read only the first time step (which should have zero mass and flow) and create new files (.tab files) containing only the first time step. Therefore, the CFDIRECT keyword in the gwdrop.key file needs to point to a subdirectory that contains these files.

3.2.2 Where Vadose Zone Flow and Transport Files Reside in SAC

In SAC, the vadose zone is modeled separately for each vadose zone site identified to ESP through the ESD keyword input file's AGGREGATE keyword. SAC defines a special subdirectory, always named /vadose, to manage vadose zone calculations. Both the VADER and STOMP codes share the /vadose subdirectory and all lower-level subdirectories therein. Figure 3-1 depicts the structure of the /vadose subdirectory.

```
.../assessment/vadose
 │
 ├stochastic_stomp.key
 └.../vadose/site
     │
     ├template_stomp.key
     ├.../vadose/site/analyte
     │    │
     │    └.../vadose/site/analyte/realization
     │              │
     │              ├input
     │              └restart
     └.../vadose/site/flow
          │
          └.../vadose/site/realization
                  │
                  ├input
                  └restart
```

Figure 3-1 Structure of the /vadose directory

In this illustration of the /vadose subdirectory structure, fixed (hard-coded) names are bold, and variable names are plain text. For example, "**/vadose**" is literal, but "/site" could be /216-H-8 or /memfis, where the names of sites are defined by the AGGREGATE keyword in the ESD keyword input file. Similarly, "/analyte" could be /H3, /CCl4, or /nasty, where the names are defined by the ANALYTE keyword in the ESD keyword input file. Realizations are numbered, and the number of digits used depends on the total number of realizations. Thus, for a 25-realization run, /realization could be /01, /02, or /25, but for a five-realization run it could be /1, /2, or /5. The /site, /analyte, and /realization directories are repeated for as many sites, analytes, and realizations as are specified in the ESD keyword input file.

The depiction in Figure 3-1 also indicates the appropriate location of two critical types of input files for the vadose zone flow and transport module: the stochastic_stomp.key and the template_stomp.key files. The stochastic_stomp.key file defines the values of all parameters treated stochastically in the vadose zone in a SAC assessment. The template_stomp.key file is a baseline STOMP input file that defines the deterministic vadose zone flow and transport simulation inputs for a site; one of these is required for each simulated vadose zone site. These files are discussed in more detail in Section 3.4.1.5. To illustrate the /vadose subdirectory structure further, consider an example SAC application, the “big” assessment, involving two sites (alpha and beta), two analytes (nitrate and molybdenum), and two realizations. The following would be the defining keywords in the ESD keyword file for this assessment (see Section 2.1):

```
REALIZAT 2 compute
ANALYTE ID=“Ni” NAME=“Nitrate” … compute
ANALYTE ID=“Mo” NAME=“Molybdenum” … compute
AGGREGATE “alpha” … compute
AGGREGATE “beta” … compute
```

In this assessment, the /vadose zone subdirectory would appear as shown in Figure 3-2.

```
.../big/vadose
   │
   ├─stochastic_stomp.key
   ├─/alpha
   │  │
   │  ├─template_stomp.key
   │  ├─/Mo
   │  │  │
   │  │  ├─/01
   │  │  └─/02
   │  ├─/Ni
   │  │  │
   │  │  ├─/01
   │  │  └─/02
   │  └─/flow
   │     │
   │     ├─/01
   │     └─/02
   └─/beta
      │
      ├─template_stomp.key
      ├─/Mo
      │  │
      │  ├─/01
      │  └─/02
      ├─/Ni
      │  │
      │  ├─/01
      │  └─/02
      └─/flow
         │
         ├─/01
         └─/02
```

Figure 3-2 Example /vadose zone Subdirectory

It is obvious even from this very simple example that SAC manages a very large directory structure for vadose zone calculations. The actual STOMP input and output files reside at the lowest level, the /realization level. Notice the extra directory at the /site level in the above depiction. The /site/flow directory is where ESP prepares a special STOMP simulation to solve the initial flow conditions for a given site and realization. The end product of each flow solution is a STOMP restart file that is copied from /site/flow/realization directories into each corresponding /site/analyte/realization directory. Thus, in the above example "big" assessment, the file /vadose/alpha/flow/1/restart is copied to /vadose/alpha/Mo/1 and /vadose/alpha/Mo/2. This provides the initial conditions for the transport solutions for Mo and Ni.

3.3 Code Environment

Running a SAC simulation is a multi-step, iterative process. The user will create directories and files, run ESP, create more files and copy files into the subdirectory structure, rerun ESP, modify more files, rerun ESP, and so on. This section discusses a typical execution path. The ESP can run an entire assessment in two passes. The first pass creates the subdirectory structure using the CREATEDIR keyword of the ESD file (see Section 2.1.5). The second pass runs the entire simulation. Alternatively, each process can be run separately using the MODULE keyword of the ESD file (see Section 2.1.16).

In practice, the user will typically run one process at a time (for example, inventory, then vadose, then groundwater transport, then river transport, and then finally river shore), inspect the results of the specific process, and then either rerun the process or proceed to the next process.

3.3.1 How Code Is Invoked

Under the Linux operating system, ESP is executed from a command window through the following Bourne Shell or C Shell command:

```
esp.exe ESD_Initial.key
```

This method will cause many diagnostic results to be written to the screen. To reduce the amount of output to the screen and log it for later examination, the following command could be used:

```
(esp.exe ESD.key > esp.log) >& errors.log &
```

This method will place the execution of ESP in the background (freeing the screen), log all diagnostics that would have come to the screen to the file esp.log, and log any system messages to the error.log file.

ESP distributes the majority of the work across a network of processors. All of the processors are defined using the PROCESSOR keyword of the ESD file (see Section 2.1.19). The directory for the work to be done on the external processor is also contained on the PROCESSOR keyword. An example of a PROCESSOR keyword is shown below:

```
PROCESSOR MACHINE="c0-0" RUNDIR="…/assessment" compute
```

In this example, the processor is named "c0-0" and the analysis will be done under the …/assessment subdirectory. This distributed modeling technique will only work if the external processors can access the same disk with the same path names as those used by the main processor.

During execution, ESP will create batch procedures for performing portions of the analysis. These batch procedures will be submitted using a remote (secure) shell command. The following is an example of this command:

```
ssh  -f c0-0 source …/assessment/processors/p121.com
```

As a result of this command, the batch procedure (…/assessment/processors/p121.com) will be submitted on the c0-0 processor. The 121 in the batch procedure file name (p121.com) is derived from the list of processors. Processor c0-0 must be the 121st entry in the processor list (as specified by the PROCESSOR keyword of the ESD file). Table 3.1 shows a sample batch procedure for modeling release and transport in the vadose.

Table 3.1 Batch Procedure …/processors/p121.com

```
#!/bin/csh
# execute vader and stomp for: 20020412090209
cd /home/ANALYSIS/Initial2/
if (-e processors/p121.done ) rm -f processors/p121.done
if (-e processors/p121.fail ) rm -f processors/p121.fail
echo 0 > processors/p121.run
echo 0 > processors/p121.start

cd /home/ANALYSIS/Initial2/vadose/307_RB/U238/23/
# run vader
if (-e vader.done ) rm -f vader.done
if (-e vader.fail ) rm -f vader.fail
vader.exe /home/ANALYSIS/Initial2/vadose/307_RB/U238/23
if ( -e vader.fail ) then
   cd /home/ANALYSIS/Initial2/
   echo "vader failed: /home/ANALYSIS/Initial2/vadose/307_RB/U238/23/" > processors/p121.fail
   goto done
endif

# modify input for AreaX
prestomp.exe input  -1.0000E+00
if ( -e prestomp.fail ) then
   cd /home/ANALYSIS/Initial2/
   echo "prestomp failed: /home/ANALYSIS/Initial2/vadose/307_RB/U238/23/" > processors/p121.fail
   goto done
endif

# run stomp
rm -f plot.*
rm -f restart.*
if (-e stomp.done ) rm -f stomp.done
if (-e stomp.fail ) rm -f stomp.fail
stomp.exe > stomp.log
if ( -e stomp.fail ) then
   cd /home/ANALYSIS/Initial2/
   echo "stomp failed: /home/ANALYSIS/Initial2/vadose/307_RB/U238/23/" > processors/p121.fail
   goto done
endif
rm -f restart.*

cd /home/ANALYSIS/Initial2/
echo 0 > processors/p121.done
done:
rm -f processors/p121.run
echo done with vadose modeling: vadose/307_238/23/
```

Table 3.2 provides a sample procedure for modeling the groundwater transport (running the VZDROP, CFEST, and GWDROP codes).

Table 3.2 Batch Procedure …/assessment/processors/p002.com

```
#!/bin/csh
# script to run GW transport for:20020414091255
cd /home/ANALYSIS/Median2/
if (-e processors/p002.done ) rm -f processors/p002.done
if (-e processors/p002.fail ) rm -f processors/p002.fail
echo 0 > processors/p002.run
echo 0 > processors/p002.start

cd /home/ANALYSIS/Median2/cfest/CrVI/1/
# run vzdrop.exe
if (-e vzdrop.done ) rm -f vzdrop.done
vzdrop.exe vzdrop.key
if ( -e vzdrop.fail ) then
   cd /home/ANALYSIS/Median2/
   echo "vzdrop failed: cfest/CrVI/1/" > processors/p002.fail
   goto done
endif

echo 0 > cfest.run
echo 0 > cfest.start
cp ../bincf/cfest.* bincf
cp ../cf_tmpbinary/*.* cf_tmpbinary
cp ../cf_runs.out .
lprog3i_001.exe
lprog3_001.exe
lgethheld_001.exe
cp hheldq/hheldq001.tab .
cp hheldm/hheldm001.tab .

cd /home/ANALYSIS/Median2/
rm  /home/ANALYSIS/Median2/mass2/CrVI/1/cfest/COV*.DAT
rm  /home/ANALYSIS/Median2/mass2/CrVI/1/cfest/TMS*.DAT
rm  /home/ANALYSIS/Median2/mass2/CrVI/1/cfest/polygons.dat
gwdrop.exe /home/ANALYSIS/Median2/mass2/CrVI/1/cfest/gwdrop.inp

cd /home/ANALYSIS/Median2/
echo 0 > processors/p002.done
done:
rm -f processors/p002.run
echo done with GW modeling: cfest/CrVI/1/
```

Table 3.3 provides a sample procedure for modeling the river transport (running MASS2 and CRDROP).

Table 3.3 Batch Procedure…/assessment/processors/p005.com

```
#!/bin/csh
# script to run Mass2 transport for:20020327161427
cd /home/ANALYSIS/Median2/Bg/
if (-e processors/p005.done ) rm -f processors/p005.done
if (-e processors/p005.fail ) rm -f processors/p005.fail
echo 0 > processors/p005.run
echo 0 > processors/p005.start

cd /home/ANALYSIS/Median2/Bg/mass2/Cs137/1/
```

```
# run mass2
if (-e mass2.done ) rm -f mass2.done
if (-e mass2.fail ) rm -f mass2.fail
(exec /home/CODES/mass2/bin/runmass2NB -l /home/ANALYSIS/Median2/Bg/mass2/Cs137/1/)
if ( -e mass2.fail ) then
   cd /home/ANALYSIS/Median2/Bg/
   echo "mass2 failed" > processors/p005.fail
   goto done
endif

# run crdrop
cd crdrop
if (-e crdrop.done ) rm -f crdrop.done
if (-e crdrop.fail ) rm -f crdrop.fail
cp   /home/ANALYSIS/Median2/Bg/ESD_Median2_Bg.key .
cp   /home/ANALYSIS/Median2/Bg/mass2/CrossIndex.grd .
crdrop.exe ESD_Median2_Bg.key 20020327161427    1 1 Cs137
rm  ESD_Median2_Bg.key
rm  CrossIndex.grd
if ( -e crdrop.fail ) then
   cd /home/ANALYSIS/Median2/Bg/
   echo "crdrop failed" > processors/p005.fail
   goto done
endif

cd /home/ANALYSIS/Median2/Bg/
echo 0 > processors/p005.done
done:
rm -f processors/p005.run
echo done with GW modeling: /home/ANALYSIS/Median2/Bg/mass2/Cs137/1/
```

3.3.2 Code Control and Keyword Descriptions

ESP is controlled entirely by the ESD keyword file. Section 2.1 describes each keyword of this file. Table 3.4 provides an example ESD keyword file.

Table 3.4 Example ESD Keyword File for Use by ESP

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Purpose:
!   Run SAC-ESP
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
REPORT "setup.rpt"                            ! Define the ESD report file name
TITLE  "test - ESP"                           ! Define the problem title
USER   "name"                                 ! Define the user
DEBUG STOCHASTIC="stoch/esp_stoch.out"        ! Output stochastic values
REALIZATION 10                                ! Number of realizations to run
! RESTART 1 2 3 4 5 6 7 8 9 10 compute        ! Restart (realization) mode only
! RESITE  “600-148”  compute                  ! Restart (site) mode only
PERIOD START=1944 STOP=3050 CLOSURE=2050      ! Time period for the analysis
BALANCE 2000 2100 2200 2300 2400 2500 2600 2800 3000  ! mass balance times

OS "Unix"
EXEDIR "/home/CODES/bin"
FILLECDA 0.0001

!******************** ESP SPECIFIC INPUT **********************************
!
! BACKGROUND compute
```

```
!
! create inputs
! IOONLY compute
!
! Create Subdirectory -- 1st time only
! CREATEDIR compute
!
! ESP modules being used
! MODULE ID="INVENTORY"       compute
! MODULE ID="VADER"           compute
! MODULE ID="STOMP" FLOW      compute
! MODULE ID="STOMP" TRAN      compute
! MODULE ID="CFEST"           compute
! MODULE ID="GWDROP"          compute
! MODULE ID="MASS2"           compute
! MODULE ID="RIPSAC"          compute
! MODULE ID="ECDA"            compute

! ESP processors:
PROCESSOR MACHINE="c0-0" RUNDIR="/home/ANALYSIS/Initial2" compute
PROCESSOR MACHINE="c0-1" RUNDIR="/home/ANALYSIS/Initial2" compute
PROCESSOR MACHINE="c0-2" RUNDIR="/home/ANALYSIS/Initial2" compute
PROCESSOR MACHINE="c0-3" RUNDIR="/home/ANALYSIS/Initial2" compute

!********************************* Analytes ***************************
! Definition of the analytes to be included in the run
ANALYTE ID="H3"    NAME="Tritium"       TYPE="NR" COMPUTE
  MOLWGT   = 3.0161        MOLDIFF  = 5.0E-08
  HALFLIFE = 12.33         SPECIFIC = 9.7245E+3
  DFIMM    = 0             DFSED    = 0
  GAMMA    = 0             HENRY    = 0
!
ANALYTE ID="Sr90"  NAME="Strontium 90"  TYPE="NR" COMPUTE
  MOLWGT   = 89.9077       MOLDIFF  = 5.0E-11
  HALFLIFE = 28.78         SPECIFIC = 1.3976E+2
  DFIMM    = 0             DFSED    = 3.72000E-21
  GAMMA    = 0             HENRY    = 0

!***************************** Sites *************************************
AGGREGATE ID="116-B-5" NW_NORTH=144765.809 SE_NORTH=144757.691
  NW_EAST=565285.566 SE_EAST=565293.684 TITLE="Crib" AREAX=1.0 compute
AGGREGATE ID="116-B-7" NW_NORTH=145327.654 SE_NORTH=145321.596
  NW_EAST=565254.409 SE_EAST=565260.467 TITLE="Outfall" AREAX=1.0 compute
AGGREGATE ID="118-B-8" NW_NORTH=144543.637 SE_NORTH=144480.801
  NW_EAST=565256.770 SE_EAST=565319.606 TITLE="Reactor" AREAX=1.0 compute
AGGREGATE ID="216-B-17" NW_NORTH=134401.958 SE_NORTH=134377.574
  NW_EAST=573570.620 SE_EAST=573595.004 TITLE="Crib" AREAX=1.0 compute
AGGREGATE ID="241-A-101" NW_NORTH=136053.427 SE_NORTH=136033.167
  NW_EAST=575312.245 SE_EAST=575332.505 TITLE="Single-Shell Tank"
  AREAX=1.0 compute
AGGREGATE ID="600-148" NW_NORTH=134691.150 SE_NORTH=134376.600
  NW_EAST=568708.787 SE_EAST=569023.337 TITLE="Landfill (Lined)"
  AREAX=1.0 compute

RECHARGE SITE="116-B-5" START=1944 END=1949 1 1.50 UNIT="mm/yr" SEED=4321.98
  "Pre-Operations" compute
RECHARGE SITE="116-B-5" START=1950 END=1996 1 55.40 UNIT="mm/yr" SEED=4321.98
  "Operations " compute
RECHARGE SITE="116-B-5" START=1997 END=2027 1 4.00 UNIT="mm/yr" SEED=4321.98
  "Barrier " compute
RECHARGE SITE="116-B-5" START=2028 END=2035 1 3.50 UNIT="mm/yr" SEED=4321.98
```

```
  "Degrade 1 " compute
RECHARGE SITE="116-B-5" START=2036 END=2042 1 3.50 UNIT="mm/yr" SEED=4321.98
  "Degrade 2 " compute
RECHARGE SITE="116-B-5" START=2043 END=2050 1 2.50 UNIT="mm/yr" SEED=4321.98
  "Degrade 3 " compute
RECHARGE SITE="116-B-5" START=2051 END=2057 1 2.00 UNIT="mm/yr" SEED=4321.98
  "Degrade 4 " compute
RECHARGE SITE="116-B-5" START=2058 END=3050 1 1.50 UNIT="mm/yr" SEED=4321.98
  "Long Term " compute
RECHARGE SITE="116-B-7" START=1944 END=3050 1 1.00 UNIT="mm/yr" SEED=4321.98
  "Pre-Operations" compute
RECHARGE SITE="118-B-8" START=1944 END=3050 1 0.10 UNIT="mm/yr" SEED=4321.98
  "Operations " compute
RECHARGE SITE="216-B-17" START=1944 END=3050 1 0.90 UNIT="mm/yr" SEED=4321.98
  "Pre-Operations" compute
RECHARGE SITE="241-A-101" START=1944 END=3050 1 0.90 UNIT="mm/yr" SEED=4321.98
  "Pre-Operations" compute
RECHARGE SITE="600-148" START=1944 END=1995 1 4.00 UNIT="mm/yr" SEED=4321.98
  "Pre-Operations" compute
RECHARGE SITE="600-148" START=1996 END=2013 1 55.40 UNIT="mm/yr" SEED=4321.98
  "Operations " compute
RECHARGE SITE="600-148" START=2014 END=3050 1 4.00 UNIT="mm/yr" SEED=4321.98
  "Long Term " compute

REMEDIATION YEAR=1997 FROM="116-B-5" TO="600-148" DEPTH=4.5 SOIL=1.0
  DEBRIS=1.0  CAKE=1.0 SLUDGE=1.0 GLASS=1.0 CEMENT=1.0 compute

!******************************* ECDA Concentration Related Files
! SACVIEW header file for ECDA concentration files
FILE HEADER NAME="/home/ANALYISIS/Initial2/ecda/Sacview_Initial.Hdr" CREATE
!
! ECDA record number index map files for concentration data by impact type
FILE I_HUMAN  NAME="/home/ANALYISIS/Initial2/ecda/HUMA_Initial.Map"  CREATE
FILE I_ECOLOG NAME="/home/ANALYISIS/Initial2/ecda/ECOL_Initial.Map"  CREATE
FILE I_ECONOM NAME="/home/ANALYISIS/Initial2/ecda/ECON_Initial.Map"  CREATE
FILE I_CULTUR NAME="/home/ANALYISIS/Initial2/ecda/CULT_Initial.Map"  CREATE
!
! ECDA Concentration data files for each impact type and analyte
FILE C_HUMAN  ANALYTE="H3"
  NAME="/home/ANALYISIS/Initial2/ecda/HUMA_H3_Initial.Dat"     CREATE
FILE C_ECOLOG ANALYTE="H3"
  NAME="/home/ANALYISIS/Initial2/ecda/ECOL_H3_Initial.Dat"     CREATE
FILE C_ECONOM ANALYTE="H3"
  NAME="/home/ANALYISIS/Initial2/ecda/ECON_H3_Initial.Dat"     CREATE
FILE C_CULTUR ANALYTE="H3"
  NAME="/home/ANALYISIS/Initial2/ecda/CULT_H3_Initial.Dat"     CREATE
!
FILE C_HUMAN  ANALYTE="Sr90"
  NAME="/home/ANALYISIS/Initial2/ecda/HUMA_Sr90_Initial.Dat"   CREATE
FILE C_ECOLOG ANALYTE="Sr90"
  NAME="/home/ANALYISIS/Initial2/ecda/ECOL_Sr90_Initial.Dat"   CREATE
FILE C_ECONOM ANALYTE="Sr90"
  NAME="/home/ANALYISIS/Initial2/ecda/ECON_Sr90_Initial.Dat"   CREATE
FILE C_CULTUR ANALYTE="Sr90"
  NAME="/home/ANALYISIS/Initial2/ecda/CULT_Sr90_Initial.Dat"   CREATE

!****************************** Dilution factor data for RIPSAC
ECHO DILUTE ! Echo dilution data to the ECDA report file
FILE DILUTE NAME="/home/ANALYISIS/Initial2/ecda/DILUTE_Initial.Dat"
  SEED=123445.0 CREATE
DILUTE "DF5m" 6  0.36 0.47 0.63
```

```
  "Dilution factor for all sites 5 meters from the river"

!************************* Soil-water Partition Coefficients (Kd's)
ECHO KDSOIL ! Echo soil-water Kd values to the ECDA report file
FILE KDSOIL NAME="/home/ANALYISIS/Initial2/ecda/KDSOIL_Initial.Dat"
  SEED=232323.0 CREATE
!
Mass2 can't handle Kd=0, so set = 1.0E-10
KDSOIL "KDSr" 9 -3.9120 1.4903 "Soil-water Kd for strontium (m^3/kg or L/g)"
KDSOIL "KDH"  1  1.0E-10       "Soil-water Kd for tritium (m^3/kg or L/g)"
!
KDSOIL "KDSUSSr" 9 -1.7210 1.6786
  "Suspended sediment Kd for strontium (m^3/kg or L/g)"
KDSOIL "KDSUSH"  1  1.0E-10
  "Suspended sediment Kd for tritium (m^3/kg or L/g)"

!****************** ECDA concentration solution times for human impacts
TIMES HUMAN  1980 2000 2100 2200 2300 2400 2500 2600 2700 2800 2900 3000

!****************** ECDA concentration solution times for cultural impacts
TIMES CULTURAL  1980 2000 2100 2200 2300 2400 2500 2600 2700 2800 2900 3000

!***************** ECDA concentration solution times for ecological impacts
TIMES ECOLOGICAL 1980 2000 2100 2200 2300 2400 2500 2600 2700 2800 2900 3000

!**************** ECDA concentration solution times for economic impacts
TIMES ECONOMIC 1980 2000 2100 2200 2300 2400 2500 2600 2700 2800 2900 3000

!************************** Human Locations ******************************
LOCATION HUMAN ID="HL0001" NAME="Human location"
  EASTING =  591756.6  NORTHING=  105412.1  POP=0 GRO_WAT ! Upland
!
LOCATION HUMAN ID="HL0002" NAME="Human location"
  EASTING =  594021.6  NORTHING=  107116.6  POP=0 GRO_WAT ! Upland
!
LOCATION HUMAN ID="HL0003" NAME="Human location"
  EASTING =  591771.6  NORTHING=  109366.6  POP=0 GRO_WAT ! Upland
!
LOCATION HUMAN ID="HL0004" NAME="Human location"
  EASTING =  591021.6  NORTHING=  110866.6  POP=0 GRO_WAT ! Upland
!
LOCATION HUMAN ID="HL0005" NAME="Human location"
  EASTING =  588758.3  NORTHING=  111657.5  POP=0 GRO_WAT ! Upland
!
!**************************** Ecological Locations ************************
LOCATION Ecological ID="EL0001" NAME="Ecological location"
  EASTING =  557375.3  NORTHING=  144885.2
  GRO_WAT SOIL SEEP ! Riparian
  APSD    = 4.00E-02  COXYGEN = 1.10E-02  POROSITY= 3.50E-01  FOC  = 1.0E+00
  VEGCOV  = 5.00E-01  NECF    = 1.00E+00  RHOS    = 1.50E+00  SRH  = 1.8E-02
  TEMP    = 2.85E+02  MSWIND  = 3.44E+00  MZWIND  = 3.44E+00  AREA = 2.5E+03
!
LOCATION Ecological ID="EL0002" NAME="Ecological location"
  EASTING =  557363.4  NORTHING=  144938.9
  SUR_WAT POR_WAT SEDIME ! River
  APSD    = 4.00E-02  COXYGEN = 1.10E-02  POROSITY= 3.50E-01  FOC  = 1.0E+00
  VEGCOV  = 5.00E-01  NECF    = 1.00E+00  RHOS    = 1.50E+00  SRH  = 1.8E-02
  TEMP    = 2.85E+02  MSWIND  = 3.44E+00  MZWIND  = 3.44E+00  AREA = 2.5E+03
  LENGTH  = 5.00E+01
!
LOCATION Ecological ID="EL0003" NAME="Ecological location"
```

```
  EASTING =  557542.1  NORTHING=  145266.9
  GRO_WAT SOIL SEEP ! Riparian
  APSD    = 4.00E-02  COXYGEN = 1.10E-02  POROSITY= 3.50E-01  FOC  = 1.0E+00
  VEGCOV  = 5.00E-01  NECF    = 1.00E+00  RHOS    = 1.50E+00  SRH  = 1.8E-02
  TEMP    = 2.85E+02  MSWIND  = 3.44E+00  MZWIND  = 3.44E+00  AREA = 2.5E+03
  LENGTH  = 5.00E+01
!
LOCATION Ecological ID="EL0004" NAME="Ecological location"
  EASTING =  557554.5  NORTHING=  145213.3
  SUR_WAT POR_WAT SEDIME ! River
  APSD    = 4.00E-02  COXYGEN = 1.10E-02  POROSITY= 3.50E-01  FOC  = 1.0E+00
  VEGCOV  = 5.00E-01  NECF    = 1.00E+00  RHOS    = 1.50E+00  SRH  = 1.8E-02
  TEMP    = 2.85E+02  MSWIND  = 3.44E+00  MZWIND  = 3.44E+00  AREA = 2.5E+03
  LENGTH  = 5.00E+01
!
LOCATION Ecological ID="EL0005" NAME="Ecological location"
  EASTING =  558082.5  NORTHING=  145113.2
  GRO_WAT SOIL SEEP ! Riparian
  APSD    = 4.00E-02  COXYGEN = 1.10E-02  POROSITY= 3.50E-01  FOC  = 1.0E+00
  VEGCOV  = 5.00E-01  NECF    = 1.00E+00  RHOS    = 1.50E+00  SRH  = 1.8E-02
  TEMP    = 2.85E+02  MSWIND  = 3.44E+00  MZWIND  = 3.44E+00  AREA = 2.5E+03
  LENGTH  = 5.00E+01
!
!***************************** Economic Locations ***************************
LOCATION ECONOMIC ID="EN0011" NAME="ECONOMIC location"
  EASTING =  595445.1  NORTHING=  109753.5 SUR_WAT ! River
!
LOCATION ECONOMIC ID="EN0021" NAME="ECONOMIC location"
  EASTING =  607747.5  NORTHING=   99528.3 SUR_WAT ! River
!
LOCATION ECONOMIC ID="EN0031" NAME="ECONOMIC location"
  EASTING =  607700.8  NORTHING=   98962.8 SUR_WAT ! River
!
LOCATION ECONOMIC ID="EN0041" NAME="ECONOMIC location"
  EASTING =  621382.6  NORTHING=   82352.9 SUR_WAT ! River
!
LOCATION ECONOMIC ID="EN0012" NAME="ECONOMIC location"
  EASTING =  578758.9  NORTHING=  152518.1 SUR_WAT ! River
!
!***************************** Cultural Locations ***************************
LOCATION Cultural ID="CL0001" NAME="Cultural location"
  EASTING =  574521.6  NORTHING=  154366.6  AREA= 8.86290E+05 GRO_WAT !Upland
!
LOCATION Cultural ID="CL0002" NAME="Cultural location"
  EASTING =  576537.2  NORTHING=  154141.3  AREA= 8.59242E+05 GRO_WAT !Upland
!
LOCATION Cultural ID="CL0003" NAME="Cultural location"
  EASTING =  575271.6  NORTHING=  153616.6  AREA= 8.25346E+05 GRO_WAT !Upland
!
LOCATION Cultural ID="CL0004" NAME="Cultural location"
  EASTING =  576021.6  NORTHING=  153616.6  AREA= 6.30578E+05 GRO_WAT !Upland
!
LOCATION Cultural ID="CL0005" NAME="Cultural location"
  EASTING =  574276.0  NORTHING=  153966.8  AREA= 1.85106E+05 GRO_WAT !Upland
!
! Model parameters for biotic plant and animal growth and respiration rates
GROWTH   DELTA=0.002 BETA=0.25
RESPIRE  GAMMA=0.2   PHI=0.032
!
SPECIE ID="PERPHY" TYPE="QP" NAME="Periphyton"
  AE  = 2.30000E-01  AWD     = 1.00000E+01  FLIPID  = 4.18000E-02
  FOC = 3.50000E-01  GE  = 5.10000E+02  RADIUS  = 1.40000E+00
  WBMASS  = 3.50000E-05
!
SPECIE ID="PHYPLK" TYPE="QP" NAME="Phytoplankton"
  AE  = 2.30000E-01  AWD     = 1.00000E+01  FLIPID  = 4.18000E-02
  FOC = 3.50000E-01  GE  = 5.10000E+02  RADIUS  = 1.40000E+00
  WBMASS  = 3.50000E-05
!
END
```

3.4 Data Files and Simulation Control

ESP is used to build or modify input files for several deterministic codes. The following sections describe these files and provide some examples.

3.4.1 Input Files and Simulation Control

Each code in the SAC suite of codes relies on either the ESD keyword file or their own input files. This section contains descriptions of input files for each module. Several utility codes used to generate keywords for the main simulation codes take information from a master waste site definition spreadsheet. This spreadsheet is used to maintain a single source of site information. The spreadsheet is too complex to include in printed form.

3.4.1.1 Input Files for ECDA and HEADER

The creation of the environmental concentration data files is controlled by ESP. Two programs have been created (ecda.exe and header.exe), which rely solely on the ESD keyword file for their input. The program ecda.exe creates the binary concentration data files for all impact types (see Section 2.2). The program header.exe creates a header file used by the SACView program when viewing the contents of the concentration data files.

3.4.1.2 Input Files for INVENTORY

The inventory module typically will be modeled external to ESP control. Section 4.0 discusses the input files and the process required to model the inventory.

3.4.1.3 Input Files for VADER

ESP creates a VADER input file (vader.key) for each execution of VADER (vader.exe). See Table 5.12 for an example file. VADER will be executed once for every combination of sites, analytes, and realizations. ESP creates these files, all named vader.key but located in different directories, using a template file named template_vader.key and a stochastic definition file named stochastic_vader.key that are located in each release site subdirectory.

The VADER template file contains information on the analytes and release models, including parameters, for all release models being used at the specific waste site. Table 3.5 shows an example of this file.

Table 3.5 Example File …/assessment/vadose/600-148/template_vader.key

```
! VADER template keyword file (template.key)
!
TITLE "VADER template keyword file for the 600-148 site"
!
! Define models for each analyte
MODEL ANALYTE="H3" STARTREL=1944 STOPREL=3050 SOIL KD=0.0
  CS=9.7000E+07 TW=5.9400E-02
  B=1.5350E+00 A=9.8942E+08 H=5.3000E+02 COMPUTE
MODEL ANALYTE="Sr90" STARTREL=1944 STOPREL=3050 SOIL KD=0.0
  CS=1.3700E+06 TW=5.9400E-02
  B=1.5350E+00 A=9.8942E+08 H=5.3000E+02 COMPUTE
!
END
```

The stochastic definition file stochastic_vader.key contains information for all analytes, all release models being used for this waste site, and the simulation definitions. In the example of the VADER stochastic definition file (…/assessment/vadose/600-148/stochastic_vader.key) shown in Table 3.6, the soil release model is being used for the 600-148 site and for the two analytes H3 and Sr90.

Table 3.6 Example File …/assessment/vadose/600-148/stochastic_vader.key

```
! VADER stochastic keyword file (stochastic.key)
TITLE "VADER stochastic keyword file for the 600-148 site"
SEED 2323.23
DEBUG STOCHASTIC="stoch/vader_stoch.out"
!
! Define stochastic values for each analyte
STOCHASTIC ANALYTE="H3" "SOIL-KD" "6H"
STOCHASTIC ANALYTE="H3" "SOIL-CS" 1 9.7000E+07
STOCHASTIC ANALYTE="H3" "SOIL-TW" 11 3.2828E+00 5.2817E+01
  1.000E-03 9.9900E-01
STOCHASTIC ANALYTE="H3" "SOIL-B"  7 1.535E+00 1.085E-02
  TRUNCATE 1.000E-02 9.9000E-01
STOCHASTIC ANALYTE="H3" "SOIL-A"  1 9.8942E+08
STOCHASTIC ANALYTE="H3" "SOIL-H"  1 5.3000E+02
!
STOCHASTIC ANALYTE="Sr90" "SOIL-KD" "6H"
STOCHASTIC ANALYTE="Sr90" "SOIL-CS" 1 1.3700E+06
STOCHASTIC ANALYTE="Sr90" "SOIL-TW" 11 3.2828E+00 5.28171E+01
  1.000E-03 9.9900E-01
STOCHASTIC ANALYTE="Sr90" "SOIL-B"  7 1.535E+00 1.085E-02
  TRUNCATE 1.000E-02 9.9000E-01
STOCHASTIC ANALYTE="Sr90" "SOIL-A"  1 9.8942E+08
STOCHASTIC ANALYTE="Sr90" "SOIL-H"  1 5.3000E+02
!
END
```

For this simulation, using the file shown in Table 3.6, the sorption value (SOIL-KD) will be simulated using the 6H sorption distribution as defined in the STOMP stochastic definition file stochastic_stomp.key. The solubility (SOIL-CS) will be constant (same for all realizations) for both H3 and Sr90 (9.7000E+07 Ci/cm^3 and 1.3700E+06 Ci/cm^3, respectively). The fractional volumetric water content (SOIL-TW) will be simulated for H3 and Sr90 using the same Beta distribution (distribution type 11, with parameters 3.2828E+00, 5.2817E+01, 1.000E-03, and 9.9900E-01). The bulk density (SOIL-B) will be simulated for H3 and Sr90 using the same normal distribution (distribution type 7, with parameters 1.535E+00, 1.085E-02, TRUNC 1.000E-02, and 9.9000E-01). The cross-sectional area (SOIL-A) and thickness (SOIL-H) will be constant for both H3 and Sr90 (9.8942E+08 cm^2 for both analytes and 5.3000E+02 cm for both analytes, respectively).

3.4.1.4 Preparation of VADER Input Files Using the Utility Code vadertemplate.exe

A program named vadertemplate.exe creates the VADER template and stochastic files used in creating the vader.key files (by running ESP with the IOONLY keyword enabled in the ESD keyword file). To create VADER stochastic files containing median values rather a full set of stochastic generated values, run the vadertempmedian.exe program, instead of vadertemplate.exe. Use the following procedure to run vadertemplate.exe.

1. Access the master waste site definition spreadsheet (GVLsSiteList-2002-01-15-DWE.xls, for example).

2. Save the main data (the Full SAC Rev. 0 List) as a csv-format file (named vadose.csv in this example). Several of the comment columns need to be deleted, including

    - F – Other Site Names
    - J – Location Description / UPR-source-Code
    - Y – Waste Type (from HSWMUR)
    - AD – Comments / References
    - AI – Description from HSWMR.

3. Save the release table of the master spreadsheet (containing analytes, solubility, fractional release rate, and diffusion coefficient) as a csv-format file (call it solubility.dat). The following is an example of the contents of this file:

```
Analyte     Csat        Fraction Release   Diffusion Coeff
H3    9.70E+07    3.65E-04    1.58
Sr90  1.37E+06    1.10E-02    1.58E-03
```

4. Create a control file (call it vadertemplate.dat). The following is an example of the contents of this file:

```
1944 3050         -- start date, end date
2323.23                -- seed
2                 -- number of analytes
H3
Sr90
```

5. Create a file that contains the sites to process (call it sites.dat). The following is an example of the contents of this file:

```
Site Template
116-B-5
116-B-7
118-B-8
216-B-17
241-A-101
600-148
```

6. Create the STOMP stochastic definition file (called stochastic_stomp.key). Table 3.9 provides an example of the contents of this file.

7. Make sure that these five files are located in the directory to which all the sites will be created – typically, this will be in the vadose subdirectory of the assessment. If the site directories do not exist, then the program will create them. It is easier to do this step after the creation of the directory structure, which can be accomplished by running ESP with the CREATEDIR keyword enabled. The following five data files are needed for this program:

   - vadose.csv ! user supplies the name
   - vadertemplate.dat ! name is set
   - sites.dat ! name is set
   - solubility.dat ! name is set
   - stochastic_stomp.key ! name is set

8. Open a command window in Linux and change to the directory where the five files reside, typically …/assessment/vadose.

9. Run the program provided to create the ESP/VADER input files. The data file can be placed on the command line. If it is missing, then the program will prompt for it.

10. The program should create a number of files. A file named template _vader.key should be located in each waste site subdirectory. Table 3.5 provides an example of this file. A file named stochastic_vader.key should be located in each waste site subdirectory. Table 3.6 provides an example of this file. In addition, a file called models.csv should be generated. This file contains lists of sites, release models, and release parameters. The following is an example of the contents of this file:

```
116-B-5,Liquid,1.0
116-B-7,River,1.0
118-B-8,Reactor Block,3.6500E-04
216-B-17,Liquid,1.0
241-A-101,Liquid,1.0
241-A-101,Salt Cake,0.36,7(1.58;0.2;0.01;0.99),4104330
600-148,Soil Debris,6H,7(1.5;0.1;0.01;0.99),11(3.28;52.81;0.001;0.999),9.8942E+08,5.3000E+02
```

3.4.1.5 Input Files for STOMP

For every execution of VADER, STOMP will also be run. In fact, VADER and the STOMP transport are always run together. ESP creates a different STOMP input file (input-esp) for each execution of STOMP (stomp.exe), and places it in the appropriate subdirectory. Table 3.7 provides an example input-esp file. STOMP will be executed once for every combination of site, analyte, and realization. ESP uses a template file (named template_stomp.key) for each site and a stochastic definition file (named stochastic_stomp.key) when creating the STOMP input files.

Table 3.7 Example STOMP Control File input-esp

```
#SAC STOMP input created by ESP   05/06/2002 - 14:45:07
#SAC Case ID           : SAC Rev. 0 Initial2 Assessment
#SAC Template          : vadose/618-11/template_stomp.key
#SAC Site ID           : 618-11
#SAC Site NW Easting   :   588884.0
#SAC Site NW Northing  :   127356.7
#SAC Site SE Easting   :   589070.6
#SAC Site SE Northing  :   127170.0
#SAC Analyte           : Tc99
#SAC Realization       :     1

~Simulation Title Card
1,
316R-6,
W E Nichols,
Pacific Northwest National Laboratory,
January 9 2002,
3 PM PST,
2,
Template 316R - For shallow disposal sites
300 Area, Shallow Burial, Waste Chemistry Designation 6

~Solution Control Card
Restart,
Water Transport Courant,
1,
0.0,yr,1110.0,yr,1.0,s,1100.0,yr,1.5,8,1.0e-6,
10000,
0,

~Grid Card
Cartesian,
1,1,495,
0.00000E+00,m, 1.86641E+02,m,
0.00000E+00,m, 1.86641E+02,m,
0.0,ft,305@0.2,ft,220@0.5,ft,

~Rock/Soil Zonation Card
3,
Hg(6H),1,1,1,1,448,495,
Rg(6I2),1,1,1,1,306,447,
PPlz(6I2),1,1,1,1,1,305,

~Mechanical Properties Card
```

```
hg(6h),2650,kg/m^3, 1.75030E-01, 1.75030E-01,,,millington and quirk,
rg(6i2),2650,kg/m^3, 6.80613E-02, 6.80613E-02,,,millington and quirk,
pplz(6i2),2666,kg/m^3, 4.54146E-01, 4.54146E-01,,,millington and quirk,

~Hydraulic Properties Card
hg(6h), 5.42278E-04,hc cm/s, 5.42278E-04,hc cm/s, 5.42278E-04,hc cm/s,,
rg(6i2), 1.46331E-03,hc cm/s, 1.46331E-03,hc cm/s, 1.46331E-03,hc cm/s,,
pplz(6i2), 7.60036E-07,hc cm/s, 7.60036E-07,hc cm/s, 7.60036E-07,hc cm/s,,

~Saturation Function Card
hg(6h),Nonhysteretic van Genuchten, 5.53759E-02,1/cm ,1.38727E+0,4.34833E-02,,
rg(6i2),Nonhysteretic van Genuchten, 1.17821E-2,1/cm ,1.66846E+0,2.70291E-02,,
pplz(6i2),Nonhysteretic van Genuchten,6.07363E-3,1/cm ,1.98132E+0,1.00728E-1,,

~Aqueous Relative Permeability Card
Hg(6H),Mualem,,
Rg(6I2),Mualem,,
PPlz(6I2),Mualem,,

~Solute/Fluid Interaction Card
1,
Tc99,Conventional, 1.00000E-09,cm^2/s,, 2.11100E+05,yr,
0,

~Solute/Porous Media Interaction Card
hg(6h),0.09,m,,,
Tc99, 0.00000E+00,mL/g,1.0,
rg(6i2),0.09,m,,,
Tc99, 0.00000E+00,mL/g,1.0,
pplz(6i2),0.031,m,,,
Tc99, 0.00000E+00,mL/g,1.0,

~Initial Conditions Card
Gas Pressure,Aqueous Saturation,
4,
Gas Pressure,101325.0,Pa,0.0,1/m,0.0,1/m,0.0,1/m,1,1,1,1,1,495,
Aqueous Saturation ROCK,0.2230,,Hg(6H),
Aqueous Saturation ROCK,0.2259,,Rg(6I2),
Aqueous Saturation ROCK,0.2178,,PPlz(6I2),

~Boundary Conditions Card
2,
#Top,Neumann,Zero Flux,
#1,1,1,1,495,495,1,
#0.0,yr,-1.0,mm/yr,,,
Bottom,Dirichlet,Outflow,
1,1,1,1,1,1,1,
0.0,yr,101325.0,Pa,,,
Top,Neumann Aqueous,Zero Flux,
1,1,1,1,495,495,8,
0.000000000000E+00,s,-4.00000E+00,mm/yr,,,
5.680800000000E+08,s,-5.54000E+01,mm/yr,,,
2.303769600000E+09,s,-4.00000E+00,mm/yr,,,
3.281990400000E+09,s,-4.00000E+00,mm/yr,,,
3.534451200000E+09,s,-4.00000E+00,mm/yr,,,
3.755376000000E+09,s,-4.00000E+00,mm/yr,,,
4.007836800000E+09,s,-4.00000E+00,mm/yr,,,
4.228761600000E+09,s,-4.00000E+00,mm/yr,,,

~Output Control Card
1,
1,1,495,
1,1,yr,m,6,6,6,
6,
aqueous pressure,pa,
aqueous saturation,null,
z aqueous vol,mm/yr,
aqueous courant number,null,
solute volumetric conc.,Tc99,1/m^3,
solute source integral,Tc99,null,
12,
1.609459200000E+09,s,
1.767225600000E+09,s,
3.345148800000E+09,s,
4.922985600000E+09,s,
8.078659200000E+09,s,
1.123433280000E+10,s,
1.439000640000E+10,s,
1.754576640000E+10,s,
2.070144000000E+10,s,
2.701278720000E+10,s,
3.332422080000E+10,s,
3.490205760000E+10,s,
1,
solute volumetric conc.,Tc99,1/m^3,

~SAC Release Plane Card
1,
0.0,yr,1,

~Balance Card
12,
1.609459200000E+09,s,
1.767225600000E+09,s,
3.345148800000E+09,s,
4.922985600000E+09,s,
8.078659200000E+09,s,
1.123433280000E+10,s,
1.439000640000E+10,s,
1.754576640000E+10,s,
2.070144000000E+10,s,
2.701278720000E+10,s,
3.332422080000E+10,s,
3.490205760000E+10,s,

~SAC Remediation Card
  1,
2.303769600000E+09,s,Tc99, 4.500000000000E+00,m, 1.000000000000E+00,600-148,
```

The template file needs to appear under each site subdirectory (e.g., …/assessment/vadose/600-148). The template file contains all the information needed for the STOMP analysis. Table 3.8 shows a sample template file.

Table 3.8 Example File …/assessment/vadose/600-148/template_stomp.key

```
~Simulation Title Card
1,
216S-6,
name,
Pacific Northwest National Laboratory,
February 26 2001,
1 PM PST,
2,
Template 216S - For shallow disposal sites (e.g., Cribs, Burial Grounds)
200W Area (South), Cribs/Burial Grounds, Waste Chemistry Designation 6

~Solution Control Card
Restart,
Water Transport Courant,
1,
0.0,yr,1110.0,yr,1.0,s,1100.0,yr,1.5,8,1.0e-6,
4,hr,4,hr,10000,
0,

~Grid Card
Cartesian,
1,1,654,
0.0,m,39.74,m,
0.0,m,39.74,m,
0.0,ft,204@0.5,ft,200@0.2,ft,250@0.5,ft,

~Rock/Soil Zonation Card
4,
Hg(6H),1,1,1,1,525,654,
Hss(6H),1,1,1,1,405,524,

~Mechanical Properties Card
Hg(6H),2070,kg/m^3,0.166,0.166,,,Millington and Quirk,
Hss(6H),1760,kg/m^3,0.346,0.346,,,Millington and Quirk,

~Hydraulic Properties Card
Hg(6H),,,,,5.0e-3,hc cm/s,
Hss(6H),,,,,6.0e-3,hc cm/s,

~Saturation Function Card
Hg(6H),Nonhysteretic van Genuchten,0.083,1/cm,1.660,0.131,,
Hss(6H),Nonhysteretic van Genuchten,0.108,1/cm,2.111,0.086,,

~Aqueous Relative Permeability Card
Hg(6H),Mualem,,
Hss(6H),Mualem,,

~Solute/Fluid Interaction Card
1,
Tritium,Conventional Tortuosity Model,1.e-09,m^2/s,Continuous,1.0e20,yr,
0,

~Solute/Porous Media Interaction Card
Hg(6H),0.09,m,,,
Tritium,0.0,ml/g,1.0,
Hss(6H),0.203,m,,,
Tritium,0.0,ml/g,1.0,

~Initial Conditions Card
Gas Pressure,Aqueous Saturation,
5,
Gas Pressure,101325.0,Pa,0.0,1/m,0.0,1/m,0.0,1/m,1,1,1,1,1,654,
Aqueous Saturation ROCK,0.2230,,Hg(6H),
Aqueous Saturation ROCK,0.1288,,Hss(6H),

~Boundary Conditions Card
2,
Top,Neumann,Zero Flux,
1,1,1,1,654,654,1,
0.0,yr,-1.0,mm/yr,,,
Bottom,Dirichlet,Outflow,
1,1,1,1,1,1,1,
0.0,yr,101325.0,Pa,,,

~Output Control Card
1,
1,1,654,
1,1,yr,m,6,6,6,
6,
Aqueous Pressure,Pa,
Aqueous Saturation,,
Z Aqueous Vol,mm/yr,
Aqueous Courant Number,,
Solute Volumetric Conc.,Tritium,1/m^3,
Solute Source Integral,Tritium,,
0,
1,
Solute Volumetric Conc.,Tritium,1/m^3,

~SAC Release Plane Card
1,
0.0,yr,1,
```

The stochastic definition file (stochastic_stomp.key) needs to appear only under the vadose subdirectory. This file contains distribution information for each rock type for porosity, saturation, and the hydraulic properties. It also contains distribution information for each sorption class. A sample stochastic definition file is shown below:

Table 3.9 Example File …/assessment/vadose/stochastic_stomp.key

```
!=======================================================================
! Title and Random Number Generator Seed.
!=======================================================================
TITLE "STOMP stochastic definitions for SAC Rev. 0 Initial2 Assessment"
SEED 6749.24
DEBUG STOCHASTIC=”stoch/stomp_stoch.out”

!-----------------------------------------------------------------------
! Hfs (Hanford fine sand) HYDRAULIC PROPERTIES
!-----------------------------------------------------------------------
POROSITY   ROCK="Hfs"   "TOTAL"            7   3.65E-01   5.00E-02
   TRUNCATE   2.39E-02   9.40E-01
POROSITY   ROCK="Hfs"   "DIFFUSIVE"        7   3.65E-01   5.00E-02
   TRUNCATE   3.13E-02   9.40E-01
```

```
SATURATION ROCK="Hfs"   "RESIDUAL"         7   8.57E-02   4.64E-02
   TRUNCATE   3.24E-02   9.83E-01
SATURATION ROCK="Hfs"   "AIR"  UNIT="1/cm" 9  -3.79E+00   1.37E+00
   TRUNCATE   1.03E-01   9.90E-01
SATURATION ROCK="Hfs"   "EXPONENT"         7   2.36E+00   9.81E-01
   TRUNCATE   1.18E-01   9.90E-01
HYDRAULIC  ROCK="Hfs"   UNIT="hc cm/s"     9  -7.45E+00   2.05E+00
   TRUNCATE   1.00E-01   9.82E-01

!-----------------------------------------------------------------------
! Hcs (Hanford coarse sand) HYDRAULIC PROPERTIES
!-----------------------------------------------------------------------
POROSITY   ROCK="Hcs"   "TOTAL"            7   3.36E-01   8.06E-02
   TRUNCATE   4.22E-02   9.88E-01
POROSITY   ROCK="Hcs"   "DIFFUSIVE"        7   3.36E-01   8.06E-02
   TRUNCATE   7.93E-02   9.88E-01
SATURATION ROCK="Hcs"   "RESIDUAL"         7   7.62E-02   4.81E-02
   TRUNCATE   5.66E-02   9.90E-01
SATURATION ROCK="Hcs"   "AIR"  UNIT="1/cm" 9  -2.53E+00   1.02E+00
   TRUNCATE   1.00E-01   9.90E-01
SATURATION ROCK="Hcs"   "EXPONENT"         7   2.05E+00   7.15E-01
   TRUNCATE   1.37E-01   9.90E-01
HYDRAULIC  ROCK="Hcs"   UNIT="hc cm/s"     9  -6.28E+00   1.96E+00
   TRUNCATE   1.00E-01   9.60E-01

!-----------------------------------------------------------------------
! Sorption Class (1H) [Kd (mL/g)]
!-----------------------------------------------------------------------
SORPTION ANALYTE="H3"    CLASS="6H"   1   0.000
SORPTION ANALYTE="Sr90"  CLASS="6H"   9    2.303   0.297
   TRUNCATE  0.010  0.914
!-----------------------------------------------------------------------
! Sorption Class (1I1) [Kd (mL/g)]
!-----------------------------------------------------------------------
SORPTION ANALYTE="H3"    CLASS="1I1"  1    0.000
SORPTION ANALYTE="Sr90"  CLASS="1I1"  9    3.091   0.338
   TRUNCATE  0.010  0.990

!-----------------------------------------------------------------------
EFFDIF - Defines STOMP effective diffusion option.
!-----------------------------------------------------------------------
EFFDIF ANALYTE="H3"       "Conventional"
EFFDIF ANALYTE="Sr90"     "Conventional"

!-----------------------------------------------------------------------
! SOLPAR - Defines STOMP solute partition option.
!-----------------------------------------------------------------------
SOLPAR ANALYTE="H3"       "Continuous"
SOLPAR ANALYTE="Sr90"     "Continuous"

!-----------------------------------------------------------------------
! End of stochastic input for the STOMP model in SAC.
!-----------------------------------------------------------------------
END
```

3.4.1.6 Typical Steps to Prepare STOMP Input for SAC

The user who prepares the vadose zone flow and transport inputs for SAC must be a competent user of the STOMP simulator (White and Oostrom, 2000). To prepare for vadose zone flow and transport calculations in SAC, the user must take the following steps:

1. Ensure that the REALIZATION, ANALYTE, AGGREGATE, RECHARGE, and REMEDIATE keywords, at minimum, are completed in the ESD file. These collectively define the essential inputs to be used in creating STOMP input files for the vadose zone flow and transport simulations.

2. Prepare the /*assessment*/vadose subdirectory structure. To do this, modify the ESD file to include the following active keywords:
   - CREATEDIR compute
   - MODULE STOMP FLOW TRANSPORT compute

3. Run the ESP program with the modified ESD file; it will create the necessary /assessment/vadose subdirectory structure. (Note that ESP need only be run once with the CREATEDIR for the STOMP module.)

4. Prepare the following template files:
   - stochastic_stomp.key – a single file containing stochastic parameters for all STOMP simulations in the assessment
   - template_stomp.key – one vadose zone *site template* for each site identified by and AGGREGATE keyword in the ESD file for the assessment

5. Copy the template files to the appropriate location in the assessment directory:
   - /*assessment*/vadose/stochastic_stomp.key
   - /*assessment*/vadose/*site*/template_stomp.key

6. Create all STOMP input files for the assessment. To do this in advance of a production run, modify the ESD file to include the following active keywords:
   - IOONLY compute
   - MODULE STOMP FLOW TRANSPORT compute

7. Run the ESP program with the modified ESD file; it will create all of the stochastic inputs for STOMP simulations and generate all the input files for the assessment.

8. Run either the full assessment or the vadose zone portion of it. Note that all VADER inputs (Section 5.5.1) must also be ready before the vadose zone transport calculations.

3.4.1.7 Preparing STOMP Templates

A STOMP input file template is a fully functional STOMP input file used to direct a single STOMP simulation for a given vadose zone site. SAC will use the template as a base from which to prepare an individual realization of STOMP input files for a site. The process is illustrated in Figure 3-3. It is good practice to prepare the STOMP input file templates using mean values of those parameters that will be treated stochastically in a SAC assessment. This way, testing the template in advance will give you a reasonably good indication of how the simulation will perform for any vadose zone site or sites using a given template.

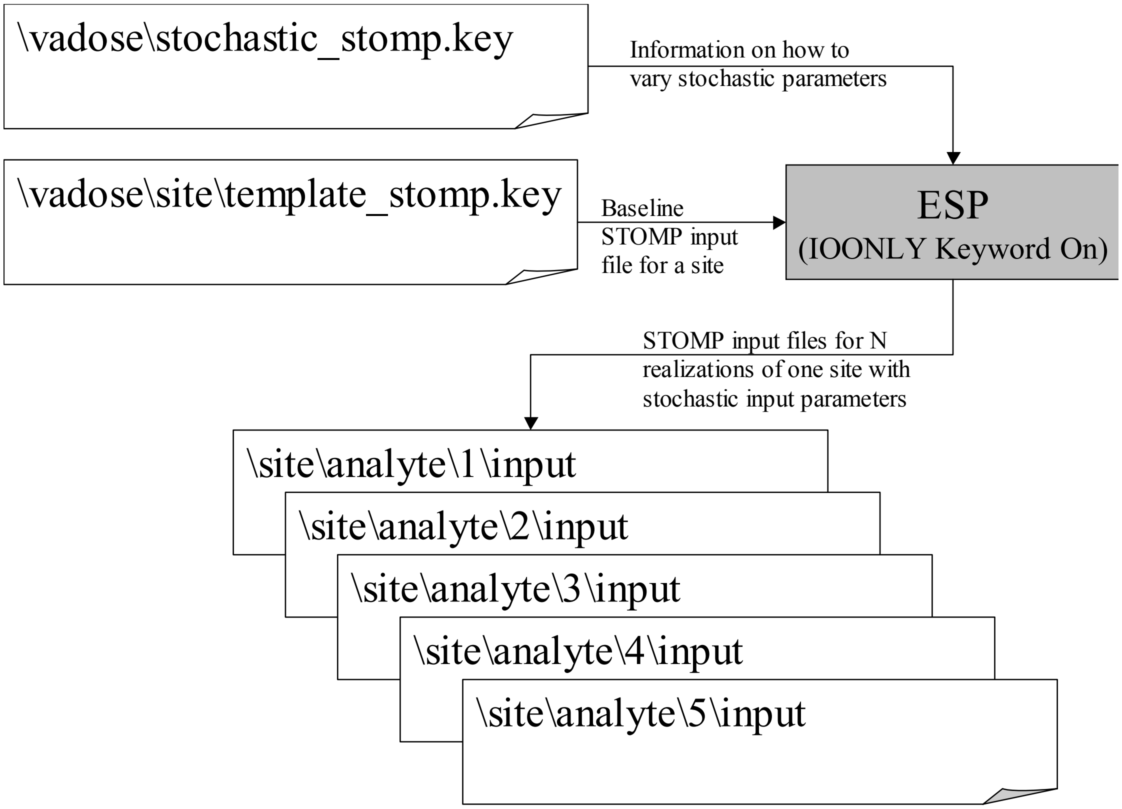

\vadose\stochastic_stomp.key
Information on how to vary stochastic parameters
\vadose\site\template_stomp.key
Baseline STOMP input file for a site
ESP
(IOONLY Keyword On)
STOMP input files for N realizations of one site with stochastic input parameters
\site\analyte\1\input
\site\analyte\2\input
\site\analyte\3\input
\site\analyte\4\input
\site\analyte\5\input

Figure 3-3 ESP Combines Information from the stochastic_stomp.key and template_stomp.key Input Files to Generate Individual STOMP Input Files for Each Stochastic Realization

There are certain necessary restrictions placed on the preparation of the STOMP input file template to make it suitable for inclusion in SAC:

- The input file ~Solution Control card must specify a “Water Transport” or “Water Transport Courant” mode
- The input file ~Grid card must be one-dimensional (or two-dimensional axi-symmetric) and use the Cartesian coordinate system
- The input file must include one dilute species (no more, no less); the actual species included and its properties are not important, as these will be overwritten in a SAC assessment.

These restrictions are imposed by limiting assumptions of the SAC, consistent with the SAC software requirements. The STOMP simulator itself is capable of simulating much more sophisticated problems, but the additional capability is not supported within the SAC framework at present.

Ideally, a STOMP template can be simulated in a standalone fashion. In fact, it is highly recommended that a fake liquid volume and dilute species ~Source Card be used to test any template before it is actually included in a SAC assessment. Such testing will ensure that the template is fully functional (does not include input formatting errors) and will considerably reduce the problems encountered in conducting a SAC assessment.

3.4.1.8 Suggestions for Managing Vadose Zone Site Templates

A typical SAC assessment will involve hundreds of vadose zone sites. While an individual template could be prepared for each of these, it is more common that you will take advantage of similarities between vadose zone sites in the same general area and of the same general nature to prepare *base templates* that apply to more than one vadose zone site.

It is recommended that a collection of base templates be prepared and tested in a separate disk location from any assessment and that a batch file be used to copy the base templates into site template locations in a SAC assessment subdirectory structure. This approach will protect the base templates from any activity in the assessment itself and will make standalone testing of the base templates easier to manage.

A simple example will be used to illustrate the general recommended approach. Consider an assessment with five vadose zone sites, of which the first three (X, Y, and Z) are very alike (same general stratigraphy and hydraulic properties with waste disposed at same general depth), while the remaining two (A and B) are also alike but different from X, Y, and Z. Thus, two base templates can be used to represent these five individual sites. Templates would then be prepared in the following directory structure:

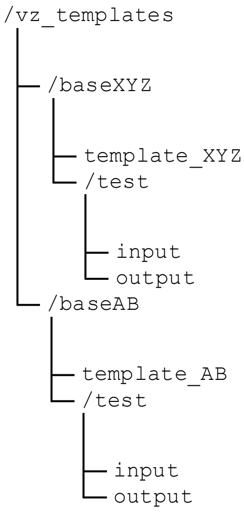


Figure 3-4 Sample Template Directory Structure

In this case, make test versions of the templates under /test subdirectories, and when satisfied, place a final version of the each base template in the locations …./baseXYZ/input_baseXYZ and

…/baseAB/input_baseAB. Next, a site template must be placed in each location for this "simple" assessment. To do this, prepare a batch file containing commands similar to the following:

```
cp /vz_templates/baseXYZ/template_XYZ /simple/vadose/X/template_stomp.key
cp /vz_templates/baseXYZ/template_XYZ /simple/vadose/Y/template_stomp.key
cp /vz_templates/baseXYZ/template_XYZ /simple/vadose/Z/template_stomp.key
cp /vz_templates/baseAB/template_XYZ  /simple/vadose/A/template_stomp.key
cp /vz_templates/baseAB/template_XYZ  /simple/vadose/B/template_stomp.key
```

This batch file, when executed, will place a copy of the desired base templates for sites X, Y, and Z in the appropriate location in the assessment subdirectory structure with the required site template file name "template_stomp.key'. Similarly, a copy of the base template file to be used for both sites A and B will be placed in the appropriate locations for those sites. This method efficiently propagates two templates into five vadose zone sites in our assessment.

The advantages to managing templates in the manner described above include the following:

- Easy to track testing of base templates
- Base templates are preserved apart from any assessments and can be "pegged" to the data rather than the assessments
- The base templates can easily be reused for several assessments by adapting the batch file used to propagate base templates to site templates
- If a correction is needed for a base template, it is easy to rerun the batch file after making the correction to ensure that site templates are updated for all sites that use a corrected base template.

3.4.1.9 Input Files for VZDROP

Vadose zone transport results computed by STOMP are used as boundary conditions for the groundwater transport, modeled using the CFEST code. The VZDROP code takes results from STOMP and inserts them into the proper input files for CFEST. ESP will create the needed VZDROP input file (vzdrop.key) to run VZDROP. Section 6.0 further describes VZDROP.

3.4.1.10 Input and Control for Groundwater Transport Using CFEST

Once VADER and STOMP have completed their modeling of the vadose zone release and transport for all sites, analytes, and realizations, the groundwater transport module (CFEST) can be run. CFEST requires many input files, which need to be placed in the appropriate subdirectories. The user who prepares the groundwater flow and transport inputs for SAC must be a competent user of the CFEST simulator (CFEST 2000). ESP will simulate CFEST stochastic variables and place them in the appropriate files. Table 3.10 provides a sample CFEST control file.

Table 3.10 Sample CFEST Control File …/assessment/cfest/H3/cfest.key

```
'cfest_001',  Important this version number should match code version
0,nprint_ctl
'hanford.qa'
'bincf/cfest'                  Line 1 - path and name for binary & ascii output files
input.lp1                      Line 2 - LPROG1 input file name
input.l3i                          Line 3 - LPROG3I input filename
input.hbc                          Line 4 - time dependent head B.C. file name
concbc.dat                         Line 5 - time dependent head B.C. file name
MKD                                Line 6 - Model units (valid options are "FPS" or "MKS")
0, 0, 0, 1, 0, 0, 1, 1,            Line 7 - print options:
1,0,                               Line 8 - NTSUBD,INTERM Time step sub-division
0,0,                               Line 9 - ITT (initial or restart time step),ITSTOP(0=ALL)
0, 1, 1, 0,                        Line 10 - NHEAD_NEW,IOPT_HEAD,NCYES,NDYES(1=y,0=n)
2, 4,                              Line 11 - NDIGH,NDIGC ** digits in head & conc output
1.00,                              Line 12 - Implicit (=1.0) CRANK-NICOLSON (0.5-1.0)
1.00,                              Line 13 - UPSTRM (0.0 to 1.0) Upstream factor
1.0,4492.47,                       Line 14 - RETARD,HALF-LIFE in simulation units
1,                                 Line 15 - NCROSS (0=norm,1=enhanced,2=same as 1 except exit,
3=cross deriv. to 0
2,0.00100,                         Line 16 - ITRHO (Max. rho iter.), RHO convergence (0.01-.0001)
1.E-06,1.E-05,                     Line 17 - RPARM1_H, RPARM1_C iterative convergence factors
1,                                 Line 18 - ITRANS (0=confined, 1=unconfined, 2=both)
15,0.750000E-01,0.090000,2,        Line 19 - IWT_TIMES,ERR_TRANS,THICK_MIN,IWRITE_DRY
0,                                 Line 20 - IOPT_K (0=normal,1=unsat_k.tab)
1,0,0.100000E-05,0.100000E-05,     Line 21 - IOPT_M,IM_ERR,FL_NODAL_ERRMIN, SOL_NODAL_ERRMIN
1.00000,                           Line 22 - FACT_VOL factor for printing fluid volume
cu. meters                         Line 23 - FUNITS_VOL text string for fluid volume output
Kg                                 Line 24 - Mass units
day                                Line 25 - time units(e.g. sec, days)
1.00000,                           Line 26 - FACT_AREA used for printing only
sq. meters                         Line 27 - area Units used for printing only
*----------------------------------------------------------------------------------------
* NSPGC ACCELRATOR (BCGS is most tested. LANMIN=used if BCGS not converged, others see manual)
* IACCEL_C=0(BCGS),1(LANMIN),2(CGCR), 3(LANDIR), 4(OMIN), 5(ODIR)
* ITMAX_Factor=1-5 (5 used as default - user can include lower values if system locks in)
*----------------------------------------------------------------------------------------
1,5,
```

The only variable being simulated for CFEST is the retardation coefficient for each analyte. Table 3.11 provides a sample stochastic definition file.

Table 3.11 Sample Stochastic Definition File for CFEST …/assessment/cfest/stochastic.key

```
! CFEST stochastic keyword file (stochastic.key)
!
TITLE "CFEST Stochastic Definition Keyword File"
SEED 3434.34
DEBUG STOCHASTIC="stoch/cfest_stoch.out"
!
!--- Sorption Class F1 (Groundwater)
RETARDATION ANALYTE="H3"     1   1.0              compute
RETARDATION ANALYTE="Tc99"   1   1.0              compute
RETARDATION ANALYTE="I129"   9   1.6054  1.0306   compute
RETARDATION ANALYTE="U238"   9   3.2149  1.6418   compute
RETARDATION ANALYTE="Sr90"   9   5.0752  1.3345   compute
RETARDATION ANALYTE="Cs137"  9   7.7774  0.6512   compute
RETARDATION ANALYTE="Pu239"  9   7.3721  0.7417   compute
RETARDATION ANALYTE="CCl4"   9   0.9517  0.2609   compute
RETARDATION ANALYTE="CrVI"   1   1.0              compute
!
END
```

3.4.1.11 Location of Groundwater Flow and Transport Files in SAC

In SAC, the groundwater is modeled separately for each analyte and realization. SAC defines a special subdirectory, always named /cfest, to manage groundwater calculations. The structure of the /cfest subdirectory is depicted in Figure 3-5.

```
.../assessment/cfest
   │
   ├stochastic.key
   └.../cfest/analyte
        │
        ├cfest_runs.out
        ├cfest.key
        ├cfest.l3i
        ├cfest.lp1
        ├input.hbc
        ├ZTOP.DAT
        ├/cfest/analyte/bincf
        ├/cfest/analyte/cf_tmpbinary
        └/cfest/analyte/realization
             │
             ├cfest_runs.out
             ├cfest.ctl
             ├input.hbc
             ├input.l3i
             ├input.lp1
             ├ZTOP.DAT
             ├/cfest/analyte/realization/bincf
             ├/cfest/analyte/realization/cf_tmpascii
             └/cfest/analyte/realization/cf_tmpbinary
```

Figure 3-5 Structure of the /cfest Subdirectory

In this illustration, the /cfest subdirectory structure shows fixed (hard-coded) names in bold and variable names in plain text. For example, "**/cfest**" is literal, but "/analyte" could be /CCl4 or /nasty, where the names of analytes are defined by the ANALYTE keyword in the ESD keyword input file. Similarly, realizations are numbered, and the number of digits used depends on the total number of realizations. Thus, for a 25-realization run /realization could be /01, /02, or /25, but for a five-realization run it could be /1, /2, or /5. The /analyte and /realization directories are repeated for as many sites, analytes, and realizations as are specified in the ESD keyword input file.

Figure 3-5 also indicates the appropriate location of critical input files for the groundwater transport module. The first of these is the stochastic.key file for CFEST runs that defines the values of retardation, treated stochastically in the groundwater in a SAC assessment. When ESP is run to create groundwater input files, the actual values for each realization of retardation are reported to the stochastic.rpt file in the same directory level. The user must prepare the stochastic.key file and place it at the /assessment/cfest subdirectory level before using ESD to create groundwater input files.

Other input files for groundwater transport are the templates used to provide the pre-computed CFEST flow solution and baseline input files. A full template set is placed at each /assessment/cfest/analyte subdirectory level and consists of the following:

- cfest_runs.out – A history file created by CFEST during the flow solution; this will be the starting file for additional history output by CFEST during the transport solutions.
- cfest.key – A template for the cfest.ctl file used to control a CFEST simulation. This file must provide the specifics for controlling a transport-only simulation of CFEST. The actual

values of retardation and half-life are not important, as ESD will replace these when preparing cfest.ctl files for each realization for the analyte, but all other inputs will be used as provided in the cfest.key file.

- cfest.lp1 – A cfest .LP1 file designed for this analyte.
- cfest.l3i – A CFEST .L3I file designed for this analyte. Values of nodal concentration or dry mass injection source/sink strengths are unimportant, as these will be overwritten in a SAC assessment by the VZDROP code. However, all other values in the .L3I file will be used as provided.
- input.hbc – A file created by CFEST during the flow solution; this will be used "as-is" by the transport solution.
- ZTOP.DAT – A file only needed if CFEST is used in 2D mode; however, ESP at present still expects this file to be present (even though it is not used in a 3D simulation), so a placeholder file with this name must still be provided for 3D simulations.
- /bincf – A copy of the /bincf subdirectory and all binary files therein created by CFEST in the flow solution.
- /cf_tmpbinary – A copy of the /cf_tmpbinary subdirectory and all binary files therein created by CFEST in the flow solution.

Groundwater flow is not treated stochastically in STOMP (only groundwater transport has a stochastic aspect). Hence, it is unnecessary and inefficient to simulate groundwater flow inside the SAC framework. Moreover, CFEST flow simulations for a Hanford Site problem take a substantial length of computer time to solve (on the order of days) and would be identical for each analyte (lacking a stochastic aspect). Therefore, the SAC software presumes groundwater flow will be pre-computed and the flow solution provided as an input to SAC. Typically, the user should set up and simulate the flow solution outside the /assessment directory structure and then copy the needed files from the completed flow solution into the /assessment/cfest directory structure. A script or batch file can be extremely useful to manage this process, particularly when a single flow solution will be used for two or more analytes. For example, if a flow solution is prepared for unretarded analytes Tc99 and H3 in directory /home/DATA/Database.2/groundwater/Flow.Fast, and for retarded analyte Sr90 in /home/DATA/Database.2/groundwater/Flow.Slow, one could use the script shown in Table 3.12 to populate the Median2 Assessment directory (/home/ANALYSIS/Median2) with CFEST flow solutions. The file names on the copy commands in the table are so long that they wrap to the second line. The syntax for the copy command is actually "cp file1 file2", all on the same line.

Table 3.12 Sample script file to populate an assessment with CFEST flow solutions

```
#
# H3
rm -R /home/ANALYSIS/Median2/cfest/H3/bincf/*
rm -R /home/ANALYSIS/Median2/cfest/H3/cf_tmpbinary/*
cp /home/DATA/Database.2/groundwater/Template.Fast/cfest.key
/home/ANALYSIS/Median2/cfest/H3/cfest.key
cp /home/DATA/Database.2/groundwater/Template.Fast/cf_runs.out
/home/ANALYSIS/Median2/cfest/H3/cf_runs.out
cp /home/DATA/Database.2/groundwater/Template.Fast/cfest.lp1
/home/ANALYSIS/Median2/cfest/H3/cfest.lp1
cp /home/DATA/Database.2/groundwater/Template.Fast/cfest.l3i
```

```
/home/ANALYSIS/Median2/cfest/H3/cfest.l3i
cp /home/DATA/Database.2/groundwater/Template.Fast/input.hbc
/home/ANALYSIS/Median2/cfest/H3/input.hbc
cp /home/DATA/Database.2/groundwater/Template.Fast/ZTOP.DAT
/home/ANALYSIS/Median2/cfest/H3/ZTOP.DAT
cp /home/DATA/Database.2/groundwater/Template.Fast/bincf/*
/home/ANALYSIS/Median2/cfest/H3/bincf/
cp /home/DATA/Database.2/groundwater/Template.Fast/cf_tmpbinary/*
/home/ANALYSIS/Median2/cfest/H3/cf_tmpbinary/
#
# Tc99
rm -R /home/ANALYSIS/Median2/cfest/Tc99/bincf/*
rm -R /home/ANALYSIS/Median2/cfest/Tc99/cf_tmpbinary/*
cp /home/DATA/Database.2/groundwater/Template.Fast/cfest.key
/home/ANALYSIS/Median2/cfest/Tc99/cfest.key
cp /home/DATA/Database.2/groundwater/Template.Fast/cf_runs.out
/home/ANALYSIS/Median2/cfest/Tc99/cf_runs.out
cp /home/DATA/Database.2/groundwater/Template.Fast/cfest.lp1
/home/ANALYSIS/Median2/cfest/Tc99/cfest.lp1
cp /home/DATA/Database.2/groundwater/Template.Fast/cfest.l3i
/home/ANALYSIS/Median2/cfest/Tc99/cfest.l3i
cp /home/DATA/Database.2/groundwater/Template.Fast/input.hbc
/home/ANALYSIS/Median2/cfest/Tc99/input.hbc
cp /home/DATA/Database.2/groundwater/Template.Fast/ZTOP.DAT
/home/ANALYSIS/Median2/cfest/Tc99/ZTOP.DAT
cp /home/DATA/Database.2/groundwater/Template.Fast/bincf/*
/home/ANALYSIS/Median2/cfest/Tc99/bincf/
cp /home/DATA/Database.2/groundwater/Template.Fast/cf_tmpbinary/*
/home/ANALYSIS/Median2/cfest/Tc99/cf_tmpbinary/
#
# Sr90
rm -R /home/ANALYSIS/Median2/cfest/Sr90/bincf/*
rm -R /home/ANALYSIS/Median2/cfest/Sr90/cf_tmpbinary/*
cp /home/DATA/Database.2/groundwater/Template.Slow/cfest.key
/home/ANALYSIS/Median2/cfest/Sr90/cfest.key
cp /home/DATA/Database.2/groundwater/Template.Slow/cf_runs.out
/home/ANALYSIS/Median2/cfest/Sr90/cf_runs.out
cp /home/DATA/Database.2/groundwater/Template.Slow/cfest.lp1
/home/ANALYSIS/Median2/cfest/Sr90/cfest.lp1
cp /home/DATA/Database.2/groundwater/Template.Slow/cfest.l3i
/home/ANALYSIS/Median2/cfest/Sr90/cfest.l3i
cp /home/DATA/Database.2/groundwater/Template.Slow/input.hbc
/home/ANALYSIS/Median2/cfest/Sr90/input.hbc
cp /home/DATA/Database.2/groundwater/Template.Slow/ZTOP.DAT
/home/ANALYSIS/Median2/cfest/Sr90/ZTOP.DAT
cp /home/DATA/Database.2/groundwater/Template.Slow/bincf/*
/home/ANALYSIS/Median2/cfest/Sr90/bincf/
cp /home/DATA/Database.2/groundwater/Template.Slow/cf_tmpbinary/*
/home/ANALYSIS/Median2/cfest/Sr90/cf_tmpbinary/
```

To illustrate the /cfest subdirectory structure further, consider an example SAC application, the “big” assessment, involving two analytes (nitrate and molybdenum) and two realizations. The defining keywords in the ESD file for this assessment would be

```
REALIZAT 2 compute
ANALYTE ID=”Ni” NAME=”Nitrate” … compute
ANALYTE ID=”Mo” NAME=”Molybdenum” … compute
```

In this example assessment, the /cfest subdirectory would appear as shown in Figure 3-6.

```
.../big/cfest
 │
 ├stochastic.key
 ├/Mo
 │ │
 │ ├cfest_runs.out
 │ ├cfest.key
 │ ├cfest.l3i
 │ ├cfest.lp1
 │ ├input.hbc
 │ ├ZTOP.DAT
 │ ├/bincf
 │ ├/cf_tmpbinary
 │ ├/1
 │ │ │
 │ │ ├cfest_runs.out
 │ │ ├cfest.ctl
 │ │ ├input.hbc
 │ │ ├input.l3i
 │ │ ├input.lp1
 │ │ ├ZTOP.DAT
 │ │ ├/bincf
 │ │ ├/cf_tmpascii
 │ │ └/cf_tmpbinary
 │ └/2
 │   │
 │   ├cfest_runs.out
 │   ├cfest.ctl
 │   ├input.hbc
 │   ├input.l3i
 │   ├input.lp1
 │   ├ZTOP.DAT
 │   ├/bincf
 │   ├/cf_tmpascii
 │   └/cf_tmpbinary
 └/Ni
   │
   ├cfest_runs.out
   ├cfest.key
   ├cfest.l3i
   ├cfest.lp1
   ├input.hbc
   ├ZTOP.DAT
   ├/bincf
   ├/cf_tmpbinary
   ├/1
   └/2
```

Figure 3-6 Another Example /cfest Subdirectory Structure

The files under …/cfest/Ni/1 and …/cfest/Ni/2 are not shown but would be the same file names as for …./cfest/Mo/1 and …/cfest/Mo/2. Even this simple example demonstrates that SAC manages a large directory structure for groundwater transport calculations. The actual CFEST input and output files used to simulate groundwater transport reside at the lowest level, the /realization level.

3.4.1.12 Typical Steps to Prepare CFEST Input for SAC

To prepare for groundwater transport calculations in SAC, the user must take the following steps:

- Ensure that, at a minimum, the REALIZATION and ANALYTE keywords are completed in the ESD file. These collectively define the essential inputs to be used in creating CFEST input files for the groundwater transport simulations.
- Prepare the /*assessment*/cfest subdirectory structure. To do this, modify the ESD file to include the following active keywords:
  - CREATEDIR compute
  - MODULE CFEST compute
- Run the ESP program with the modified ESD file; it will create the necessary /assessment/cfest subdirectory structure. (Note that ESP need only be run once with the CREATEDIR for the CFEST module.)
- Prepare the following template file: stochastic.key – a single file containing stochastic parameters for retardation for all CFEST simulations in the assessment.
- Prepare flow solutions appropriate to each analyte to be simulated. Note that one flow solution may be used for more than one analyte, if appropriate (see script example in Table 3.12, above).
- Copy the completed flow solution files to the appropriate location in the assessment directory (a script or batch file is highly recommended for this step; see example in Table 3.12, above).
- Create all CFEST input files for the assessment. To do this in advance of a production run, modify the ESD file to include the following active keywords:
  - IOONLY compute
  - MODULE CFEST compute
- Run the ESP program with the modified ESD file; it will create all of the stochastic inputs for CFEST simulations and generate all the ‘cfest.ctl’ files for the assessment.
- Run either the full assessment or the groundwater transport of it.

3.4.1.13 Preparing CFEST Templates

CFEST input file templates are merely the completed flow solution files but with a new cfest.key file that specifies what the CFEST control file (‘cfest.ctl’) will be for a transport-only solution. ESP will use the cfest.key template as a base from which to prepare individual realizations of CFEST cfest.ctl control files for each analyte and realization. Figure 3-7 illustrates the process. It is good practice to prepare the CFEST input file templates using the mean value of retardation, which will be treated stochastically in a SAC assessment. This way, testing the template in advance will give you a reasonably good indication of how the simulation will perform for any groundwater case using a given template.

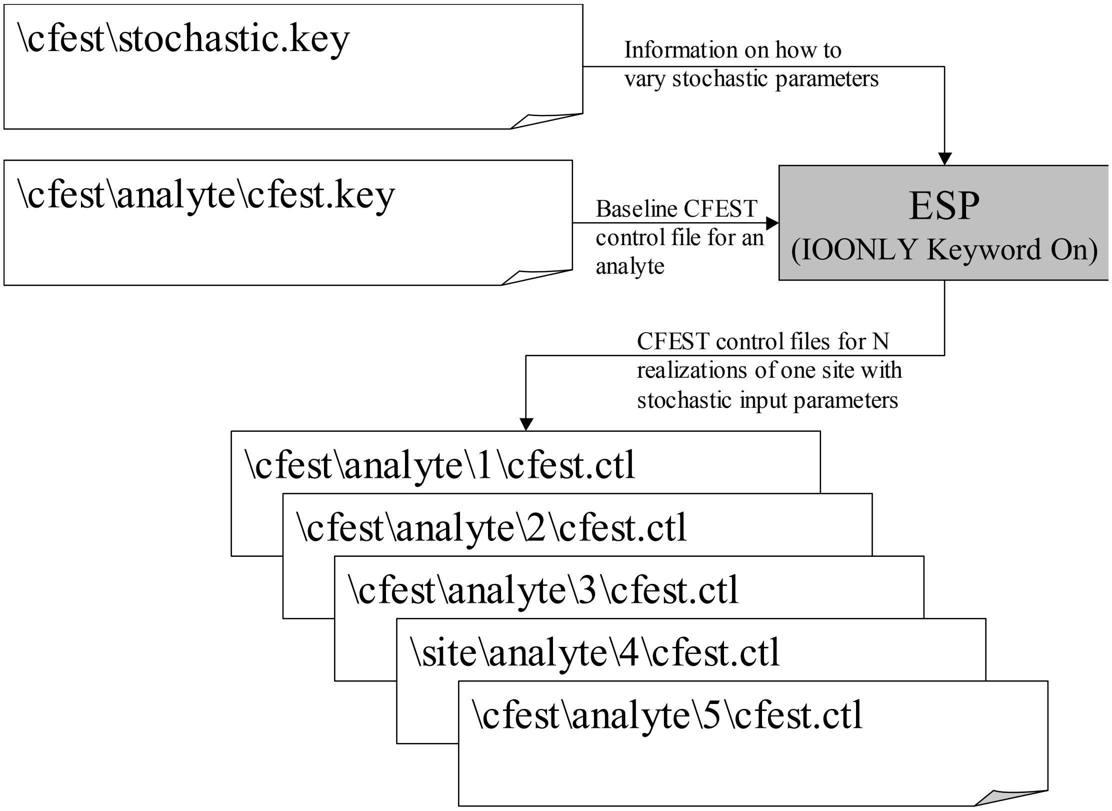

\cfest\stochastic.key
Information on how to vary stochastic parameters
\cfest\analyte\cfest.key
Baseline CFEST control file for an analyte
ESP
(IOONLY Keyword On)
CFEST control files for N realizations of one site with stochastic input parameters
\cfest\analyte\1\cfest.ctl
\cfest\analyte\2\cfest.ctl
\cfest\analyte\3\cfest.ctl
\site\analyte\4\cfest.ctl
\cfest\analyte\5\cfest.ctl

Figure 3-7 ESP Combines Information from the /cfest/stochastic.key and /cfest/analyte/cfest.key Input Files to Generate Individual CFEST Control Files for Each Stochastic Realization

Table 3.13 provides recommended parameter values for the cfest.ctl file for pre-computed flow solutions and for SAC transport solutions.

Table 3.13 Recommended Parameter Values for the cfest.ctl File for Flow Solutions and the cfest.key Template File for SAC Transport Solutions

Parameter	*Line* number	*cfest.ctl (Flow)*	*cfest.key (Transport)*
Path and file name	1	bincf/cfest	bincf/cfest
LPROG1 filename	2	input.lp1	input.lp1
LPROG3I filename	3	input.l3i	input.l3i
Time dependent head B.C. file	4	input.hbc	input.hbc
Time dependent conc B.C. file	5	concbc.dat	concbc.dat
Model Units	6	MKD	MKD
Print option: lprog1	7	-	0
Print option: lband	7	-	0
Print option: lprog3i	7	-	0
Print option: lprog3	7	-	0
Print option: limitp	7	-	0
Print option: limitt	7	-	0

Print option: nsubp	7	-	0
Print option: npmass	7	-	1
NHEAD_NEW	10	1	0
IOPT_HEAD	10	1	1
NCYES	10	0	1
NDYES	10	0	0
NDIGH	11	2	2
NDIGC	11	0	4
RPARM1_H	17	1.0E-6	1.0E-6
RPARM1_C	17	1.0E-5	1.0E-5
Mass Units	24	kg	kg
Time Units	25	day	day
NSPGC Accelerator	27	0	1
IACCEL_C	27	5	5

A typical SAC assessment will involve up to ten analytes. While an individual template could be prepared for each of these, it is more common to take advantage of similarities between analyte mobility to prepare *base flow solutions* that apply to more than one analyte.

It is recommended that a collection of base flow solutions be prepared and tested in a separate disk location from any assessment and that a batch or script file be used to copy the base flow solutions into site template locations in a SAC …/assessment/cfest subdirectory structure. This approach will protect the base templates from any activity in the assessment itself and will make standalone testing of the base templates easier to manage.

3.4.1.14 Input Files and Control of GWDROP

Data from the groundwater transport module (CFEST) to be used in the river transport module (MASS2) are compiled using the GWDROP code. Section 7.0 describes the GWDROP code and several input files needed to run GWDROP. The template file used by ESP for control of GWDROP defines the location of all the input files needed to run GWDROP. Table 3.14 provides an example template file.

Table 3.14 Example gwdrop.key file for Use in ESP

```
# GWDROP input data
TITLE "GWDROP input data"

! River Elements File
RIVERELEM "mass2/col_river_elem.dat" compute

! River Nodes File
RIVERNODE "mass2/col_river_node.dat" compute

! Interpolation type -- use POLYINT for course grid, POLYCEN for fine grid
POLYTYPE "POLYINT" compute

! transport thickness
TTHICK 20.0 compute

! CFEST output location
CFDIRECT "cfest" compute
```

```
! CFEST formats
CFFORMAT compute
   "(A,T13,0500I15)"
   "(A,T13,0500G15.7)"
   "(A,T13,0500I15)"
   "(A,T13,0500G15.7)"

! River Cell files
RIVERCELL "mass2/hanfnad83m-pt.000" compute
RIVERCELL "mass2/hanfnad83m-pt.001" compute
RIVERCELL "mass2/hanfnad83m-pt.002" compute

END
```

The gwdrop.key file must reside under the mass2 subdirectory. The files that are defined in the file include the following:

• mass2/col_river_elem.dat	the river elements file
• mass2/col_river_node.dat	the river nodes file
• mass2/hanfnad83m-pt.000	a river cell file
• mass2/hanfnad83m-pt.001	a river cell file
• mass2/hanfnad83m-pt.002	a river cell file

Running ESP using the template file in Table 3.14 with H3 and realization 1 will produce the input file for GWDROP shown in Table 3.15.

Table 3.15 Example gwdrop.inp file prepared by ESP

```
mass2/H3/01/cfest/gwdrop.log
cfest/H3/01/cfest.ctl
cfest/H3/01/input.lp1
cfest/H3/01/input.l3i
mass2/col_river_elem.dat
mass2/col_river_node.dat
mass2/H3/01/cfest/polygons.dat
mass2/H3/01/cfest/TMS
mass2/H3/01/cfest/COV
01-01-1944 00:00:00
POLYINT : TTHICK   20.0000000 : SKIP 1 : RIVER R
 1
(A,T13,0500I15)
(A,T13,0500G15.7)
(A,T13,0500I15)
(A,T13,0500G15.7)
K:ESD.key
Q:cfest/H3/01/hheldq001.tab
A:H3,RADIOACTIVE, 9724.50000, -1.0
M:cfest/H3/01/hheldm001.tab
B:cfest/H3/01
R: mass2/hanfnad83m-pt.000
R: mass2/hanfnad83m-pt.001
R: mass2/hanfnad83m-pt.002
V:/home/ANALYSIS/Initial2/vadose/116-B-7/H3/01/vader.river,   565257.438,  145324.625
```

For modeling the background-only assessment, the CFEST input for the river transport module (MASS2) will be set to zero (remember to enable the BACKGROUND keyword of the ESD file). The CFEST output location keyword (CFDIRECT) of the gwdrop.key file must point to a directory containing previously created CFEST output (hheldm001.tab and hheldq001.tab) files. ESP will read both of these files and only save the first time step (which will have zero flow and mass). A sample CFDIRECT keyword for the background-only case is shown below:

```
CFDIRECT "…/assessment/cfest/H3/01" compute
```

A sample CFDIRECT keyword for the Hanford plus background case is shown below:

```
CFDIRECT "cfest" compute
```

For the background-only case, the CFDIRECT keyword needs to contain a specific location of a results file, while this keyword in the Hanford plus background transport case points to the main CFEST subdirectory. The two keyword examples also show the difference with an absolute path (…assessment/cfest/H3/01) and a relative path (cfest). The absolute path contains the full path name (it must begin with a /), where the relative path (which does not begin with a /) contains the path below the current subdirectory (…/assessment). During an ESP run, this relative path would point to the …/assessment/cfest subdirectory.

3.4.1.15 Input Files and Control of CRDROP_INDEX

The CRDROP code (see Section 8.0) extracts data from files written by MASS2 (river flow and transport) and writes it to the environmental concentration data files (see Section 2.2). MASS2 executes once for every combination of analyte and realization. However, all MASS2 runs use the same transport grid. Therefore, a cross-index program (cdrop_index.exe) is executed once to develop a grid indexing file for later use by CRDROP. The following input file (Table 3.16) is needed by CRDROP_INDEX and must be located in the mass2 subdirectory.

Table 3.16 Example index.key File Used by ESP for CRDROP_INDEX

```
!  Input file for INDEX_CRDROP program to generate location cross index values
!
REPORT "INDEXCR.OUT"
!
TITLE "Test keyword file for INDEX_CRDROP"
USER "name"
!
! Execute
EXECUTE
!
!  Maximum allowed value for the minimum distance between locations
DISTANCE 10000000.
!
!  File name for MASS2 grid file
FILE MASS2GRID "CRDROP_grid.dat" compute
!
!  Output file for cross-index values generated by INDEX_CRDROP
FILE GRID "CrossIndex.grd" compute
!
end
```

The MASS2 grid file (CRDROP_grid.dat, as specified on the FILE MASS2GRID keyword) is needed as an input file to this module.

3.4.1.16 Input Files for MASS2

The sorption value (Kd) is the only stochastic variable used in the main module of MASS2. The Kd is specified as a function of the sediment type. The file in Table 3.17 shows the index of the Kd value (as simulated earlier when creating the ECDA files (see Section 2.1.14)) and the sediment type for both the suspended sediment and the bed sediment. This file is used by the ESP to prepare input files for MASS2.

Table 3.17 Example File mass2.key Used by ESP in Preparing MASS2 Input Files

```
# MASS2 input data

TITLE "MASS2 input data"

! Seed
SEED 54321.54321

! Kd's for suspended sediments
KDSUS ANALYTE="H3"   CLAY="KDH"     SILT="KDH"     LSAND="KDH"     SAND="KDH"
      COBBLE="KDH"     COMPUTE
KDSUS ANALYTE="Sr90" CLAY="KDSUSSr" SILT="KDSUSSr" LSAND="KDSUSSr" SAND="KDSUSSr"
      COBBLE="KDSUSSr" COMPUTE

! Kd's for bed sediments
KDBED ANALYTE="H3"   CLAY="KDH"  SILT="KDH"  LSAND="KDH"  SAND="KDH"  COBBLE="KDH"  COMPUTE
KDBED ANALYTE="Sr90" CLAY="KDSr" SILT="KDSr" LSAND="KDSr" SAND="KDSr" COBBLE="KDSr" COMPUTE

END
```

The biotic transport module of the MASS2 code requires its own stochastic definition file. At the time of publication of this document, the biotic transport portion of MASS2 has been disabled. Table 3.18 provides a sample input file for the ESP for this function. This file must be named biota.stoch and reside in the mass2 subdirectory.

Table 3.18 Example biota.stoch File for Use in ESP

```
!       Stochastic Keyword file for the MASS2 & Back_MASS2 codes

!  Purpose:
!    This keyword file is for use in the ESP program.

TITLE "Stochastic Keyword Data for MASS2 (Biotic Transport) and Back_MASS2"

SEED 12345.67
DEBUG STOCHASTIC="stoch/biota_stoch.out"
!
! Species definition list (to get data from the ESD file)
SPECIE ID="PERPHY" SHIFT=75 FLAT=30
SPECIE ID="PHYPLK" SHIFT=0  FLAT=0
!
! Botton type names for use in MASS2
! Entry order becomes the index for referencing
BOTTOM NAME="Silt and sand"
BOTTOM NAME="Small cobble"
BOTTOM NAME="Medium cobble"
BOTTOM NAME="Large cobble"
!
! Oxygen content in surface water (same at all locations)
OXYGEN VALUE=0.011
!
! KOW (octonol-water partition) values are needed for every analyte
```

```
STOCHASTIC ANALYTE="Sr90  " VARIABLE="KOW"   1       1 BOMB
STOCHASTIC ANALYTE="H3    " VARIABLE="KOW"   1       1 BOMB
!
! Fraction of lifetime spent in pore water is needed for every species
STOCHASTIC  SPECIES="PERPHY"  VARIABLE="BPORE" 1 1.0
STOCHASTIC  SPECIES="PHYPLK"  VARIABLE="BPORE" 1 0.0
!
! Bioconcentation factors are needed for analye and species combinations
STOCHASTIC ANALYTE="Sr90  " SPECIES="PERPHY" VARIABLE="BCF" 6  7.650E+03  8.500E+03 9.350E+03
STOCHASTIC ANALYTE="H3    " SPECIES="PERPHY" VARIABLE="BCF" 6  9.000E-01  1.000E+00 1.100E+00
!
STOCHASTIC ANALYTE="Sr90  " SPECIES="PHYPLK" VARIABLE="BCF" 6  4.050E+03  4.500E+03 4.950E+03
STOCHASTIC ANALYTE="H3    " SPECIES="PHYPLK" VARIABLE="BCF" 6  9.000E-01  1.000E+00 1.100E+00
!
! Minimum Biomass by bottom type and species (kg/ft^2)
STOCHASTIC VARIABLE="BMIN" SPECIES="PERPHY" BOTTOM="1" 1 0.0
STOCHASTIC VARIABLE="BMIN" SPECIES="PERPHY" BOTTOM="2" 6 6.488E-2 8.611E-2 1.033E-1
STOCHASTIC VARIABLE="BMIN" SPECIES="PERPHY" BOTTOM="3" 6 6.488E-2 8.611E-2 1.033E-1
STOCHASTIC VARIABLE="BMIN" SPECIES="PERPHY" BOTTOM="4" 6 6.488E-2 8.611E-2 1.033E-1
!
STOCHASTIC VARIABLE="BMIN" SPECIES="PHYPLK" BOTTOM="1" 6 1.722E-3 2.153E-3 2.583E-3
STOCHASTIC VARIABLE="BMIN" SPECIES="PHYPLK" BOTTOM="2" 6 1.722E-3 2.153E-3 2.583E-3
STOCHASTIC VARIABLE="BMIN" SPECIES="PHYPLK" BOTTOM="3" 6 1.722E-3 2.153E-3 2.583E-3
STOCHASTIC VARIABLE="BMIN" SPECIES="PHYPLK" BOTTOM="4" 6 1.722E-3 2.153E-3 2.583E-3
!
! Maximum Biomass by bottom type and species (kg/ft^2)
STOCHASTIC VARIABLE="BMAX" SPECIES="PERPHY" BOTTOM="1" 1 0.0
STOCHASTIC VARIABLE="BMAX" SPECIES="PERPHY" BOTTOM="2" 6 0.6028 0.7535 0.9042
STOCHASTIC VARIABLE="BMAX" SPECIES="PERPHY" BOTTOM="3" 6 0.6028 0.7535 0.9042
STOCHASTIC VARIABLE="BMAX" SPECIES="PERPHY" BOTTOM="4" 6 0.6028 0.7535 0.9042
!
STOCHASTIC VARIABLE="BMAX" SPECIES="PHYPLK" BOTTOM="1" 6 5.6E-2 7.0E-2 8.4E-2
STOCHASTIC VARIABLE="BMAX" SPECIES="PHYPLK" BOTTOM="2" 6 5.6E-2 7.0E-2 8.4E-2
STOCHASTIC VARIABLE="BMAX" SPECIES="PHYPLK" BOTTOM="3" 6 5.6E-2 7.0E-2 8.4E-2
STOCHASTIC VARIABLE="BMAX" SPECIES="PHYPLK" BOTTOM="4" 6 5.6E-2 7.0E-2 8.4E-2
!
! Spring growth rate parameter by species (1/day)
STOCHASTIC VARIABLE="KG" SPECIES="PERPHY" 6 0.112 0.140 0.168
STOCHASTIC VARIABLE="KG" SPECIES="PHYPLK" 6 0.224 0.280 0.336
!
! Senescence growth rate parameter by species (1/day)
STOCHASTIC VARIABLE="KS" SPECIES="PERPHY" 6 0.06 0.075 0.09
STOCHASTIC VARIABLE="KS" SPECIES="PHYPLK" 6 0.16 0.20  0.24
!
! Spring replenishment rate parameter by species (1/day)
STOCHASTIC VARIABLE="FG" SPECIES="PERPHY" 6 0.0096 0.012 0.0144
STOCHASTIC VARIABLE="FG" SPECIES="PHYPLK" 6 0.0464 0.058 0.0696
!
! Fall (senescence) replenishment rate parameter by species (1/day)
STOCHASTIC VARIABLE="FS" SPECIES="PERPHY" 6 0.0048 0.006 0.072
STOCHASTIC VARIABLE="FS" SPECIES="PHYPLK" 6 0.024  0.03  0.036
!
! Year information for background data
! REFERENCE = Year the input concentration data were collected
! BOMB      = Year when initial bomb pulse data came through
BACKYEAR REFERENCE=1995 BOMB=1950
!
! River IDs and names for use in the MASS2 background data
RIVER ID="PRD"     NAME="Columbia River at Priest Rapids Dam"
RIVER ID="Yakima"  NAME="Yakima River at entrance to Columbia River"
RIVER ID="IHR"     NAME="Snake River at entrance to Columbia River"
!
STOCHASTIC BACKGROUND ANALYTE="Sr90"  SOLUTION="diss"  RIVER="PRD"     9 -23.184 0.6206
STOCHASTIC BACKGROUND ANALYTE="Sr90"  SOLUTION="diss"  RIVER="Yakima"  9 -23.184 0.6206 !(PRD)
STOCHASTIC BACKGROUND ANALYTE="Sr90"  SOLUTION="diss"       RIVER="IHR"     9 -20.633 0.7360
STOCHASTIC BACKGROUND ANALYTE="Sr90"  SOLUTION="diss_floor" RIVER="PRD"     1 0.0
STOCHASTIC BACKGROUND ANALYTE="Sr90"  SOLUTION="diss_floor" RIVER="Yakima"  1 0.0
STOCHASTIC BACKGROUND ANALYTE="Sr90"  SOLUTION="diss_floor" RIVER="IHR"     1 0.0
!
STOCHASTIC BACKGROUND ANALYTE="H3" SOLUTION="diss"       RIVER="PRD"     9 -17.042 0.5933
STOCHASTIC BACKGROUND ANALYTE="H3" SOLUTION="diss"       RIVER="Yakima"  9 -17.042 0.5933 !(PRD)
STOCHASTIC BACKGROUND ANALYTE="H3" SOLUTION="diss"       RIVER="IHR"     9 -17.042 0.5933 !(PRD)
STOCHASTIC BACKGROUND ANALYTE="H3" SOLUTION="diss_floor" RIVER="PRD" 6 5.405E-9 1.081E-8 2.432E-8
STOCHASTIC BACKGROUND ANALYTE="H3" SOLUTION="diss_floor" RIVER="Yakima" 6 5.40E-9 1.08E-8 2.43E-8
STOCHASTIC BACKGROUND ANALYTE="H3" SOLUTION="diss_floor" RIVER="IHR" 6 5.405E-9 1.081E-8 2.432E-8
!
! Suspended sediment loading (kg/m^3) in each river
STOCHASTIC BACKGROUND SOLUTION="sediment" RIVER="PRD"    1 0.00375
STOCHASTIC BACKGROUND SOLUTION="sediment" RIVER="Yakima" 1 0.06
STOCHASTIC BACKGROUND SOLUTION="sediment" RIVER="IHR"    1 0.016
!
END
```

The ESP used the biota.stoch file and produces a file named biota.key for every run of MASS2. Table 3.19 provides an example file.

Table 3.19 Example biota.key File for Realization 1 of Tritium for Use by MASS2

```
$ ESP TEMPLATE File  :MASS2/Biota.Stoch
$ ESP Date Generated : 05-14-2002
$ ESP Program Name   : ESP-Unix
$ ESP Version Number : 0.7
$ ESP Program Date   : 15 Apr 2002
$ Current Run ID     : 20020514155914
TITLE "SAC Rev. 0 Initial2 Assessment"
USER "Engel, Eslinger,"
! Realization
REALIZATION CURRENT=    1 TOTAL=     25
! Analyte
ANALYTE ID="H3" TYPE="NR" NAME="Tritium" HALFLIFE=   12.33000 BOMB
! Specie
SPECIE ID="PERPHY" TYPE="QP" NAME="Periphyton" SHIFT=75 Flat=30
   AWD    =    10.00000
   FLIPID =    4.180000E-02
   FOC    =   0.3500000
   WBMASS =    3.500000E-05
SPECIE ID="PHYPLK" TYPE="QP" NAME="Phytoplankton" SHIFT=0 Flat=0
   AWD    =    10.00000
   FLIPID =    4.180000E-02
   FOC    =   0.3500000
   WBMASS =    3.500000E-05
! Growth and Respiration
GROWTH DELTA=   2.000000E-03 Beta=  0.2500000
RESPIRE GAMMA=  0.2000000 PHI=   3.200000E-02
! Bottoms
BOTTOM NAME="Silt and sand"
BOTTOM NAME="Small cobble"
BOTTOM NAME="Medium cobble"
BOTTOM NAME="Large cobble"
! Oxygen
OXYGEN    1.100000E-02
! KOW -- Stochastic (#Analytes)
KOW ANALYTE="H3" VALUE=   1.000000
! BPORE -- Stochastic (#Species)
BPORE SPECIES="PERPHY" VALUE=   1.000000
BPORE SPECIES="PHYPLK" VALUE=   0.000000E+00
! DEPRATE -- Stochastic (#Analytes x #AnimalSpecies)
! BCF -- Stochastic (#Analytes x #Species)
```

```
BCF SPECIES="PERPHY" ANALYTE="H3" VALUE=   1.067686
BCF SPECIES="PHYPLK" ANALYTE="H3" VALUE=   1.051988
! BMIN -- Stochastic (#Bottom x #Species)
BMIN SPECIES="PERPHY" BOTTOM=1 VALUE=   0.000000E+00
BMIN SPECIES="PERPHY" BOTTOM=2 VALUE=   8.632066E-02
BMIN SPECIES="PERPHY" BOTTOM=3 VALUE=   8.201657E-02
BMIN SPECIES="PERPHY" BOTTOM=4 VALUE=   7.860494E-02
BMIN SPECIES="PHYPLK" BOTTOM=1 VALUE=   2.176263E-03
BMIN SPECIES="PHYPLK" BOTTOM=2 VALUE=   2.299463E-03
BMIN SPECIES="PHYPLK" BOTTOM=3 VALUE=   2.160100E-03
BMIN SPECIES="PHYPLK" BOTTOM=4 VALUE=   2.209394E-03
! BMAX -- Stochastic (#Bottom x #Species)
BMAX SPECIES="PERPHY" BOTTOM=1 VALUE=   0.000000E+00
BMAX SPECIES="PERPHY" BOTTOM=2 VALUE=  0.7242640
BMAX SPECIES="PERPHY" BOTTOM=3 VALUE=  0.7438709
BMAX SPECIES="PERPHY" BOTTOM=4 VALUE=  0.7806304
BMAX SPECIES="PHYPLK" BOTTOM=1 VALUE=   6.488729E-02
BMAX SPECIES="PHYPLK" BOTTOM=2 VALUE=   7.773458E-02
BMAX SPECIES="PHYPLK" BOTTOM=3 VALUE=   6.275638E-02
BMAX SPECIES="PHYPLK" BOTTOM=4 VALUE=   6.783128E-02
! KG -- Stochastic (#Bottom x #Species)
KG SPECIES="PERPHY" VALUE=  0.1506909
KG SPECIES="PHYPLK" VALUE=  0.2538987
! KS -- Stochastic (#Bottom x #Species)
KS SPECIES="PERPHY" VALUE=   8.410124E-02
KS SPECIES="PHYPLK" VALUE=  0.2038334
! FG -- Stochastic (#Bottom x #Species)
FG SPECIES="PERPHY" VALUE=   1.225651E-02
FG SPECIES="PHYPLK" VALUE=   5.544397E-02
! FS -- Stochastic (#Bottom x #Species)
FS SPECIES="PERPHY" VALUE=   1.682856E-02
FS SPECIES="PHYPLK" VALUE=   3.137906E-02
! Year info for background data
BACKYEAR REFERENCE=        1995 BOMB=        1950
! Time period for the analysis
PERIOD START=        1944 STOP=        3050 CLOSURE=        2050
! River IDs and Names
RIVER ID="PRD" NAME="Columbia River at Priest Rapids Dam"
RIVER ID="Yakima" NAME="Yakima River at entrance to Columbia River"
RIVER ID="IHR" NAME="Snake River at entrance to Columbia River"
! Dissolved -- Stochastic (#Analytes x #Rivers)
DISSOLVED ANALYTE="H3" RIVER="PRD" VALUE=   2.089807E-08 FLOOR=   2.390706E-08
DISSOLVED ANALYTE="H3" RIVER="Yakima" VALUE=   2.457327E-08 FLOOR=   1.865544E-08
DISSOLVED ANALYTE="H3" RIVER="IHR" VALUE=   7.597653E-08 FLOOR=   1.014443E-08
! SEDIMENT -- Stochastic (#Rivers)
SEDIMENT RIVER="PRD" VALUE=   3.750000E-03
SEDIMENT RIVER="Yakima" VALUE=   6.000000E-02
SEDIMENT RIVER="IHR" VALUE=   1.600000E-02
!
END
```

3.4.1.17 Input Files and Control of CRDROP

CRDROP will read the result file from a single run of the MASS2 code, for example, the file …/assessment/mass2/H3/01/crdrop/H3_0001.RIV, and extract sediment, pore water and surface water concentrations that are then placed into the environmental concentration data files (see Section 2.2.1) for all specified impact types. CRDROP reads the ESD keyword file and a crdrop.key file (created by ESP) to determine where data are located. Section 8.4.1.1 describes the entries for the crdrop.key file. Table 3.20 provides an example file.

Table 3.20 Example CRDROP Keyword File Written by ESP

```
! CRDROP keyword file
!
REPORT "crdrop.out"
TITLE "SAC Rev. 0 Initial2 Assessment"
USER "Engel, Eslinger,"
!
! Do full execution
EXECUTE
!
! Cross index file to MASS2 grid locations
FILE INDEX_CR "CrossIndex.grd"
!
! MASS2 output file path
FILE MASS2 ""
!
END
```

3.4.1.18 Inputs and Control of RIPSAC

The RIPSAC code is used to model the riparian zone (river shore) concentrations in seep water and the associated soil. RIPSAC reads the ESD keyword file and a control keyword file (…/assessment/ripsac/ripsac.key, which is a required file name) for its inputs. The ESD keyword file name must be present on the FILE ESD keyword of the ripsac.key file, as illustrated in Table 3.21. Section 9.0 further describes RIPSAC. RIPSAC can be run once (all analytes, all impacts types, and all realizations), or it can be run multiple times where each run processes a subset of analytes, impact types, and realizations.

Table 3.21 Example File ripsac.key Used in the RIPSAC Code

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Purpose:
!   This file is the RIPSAC Keyword File
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! Report File (first keyword)
REPORT "ripsac/riprac.rpt"
!
! Environmental simulation control keyword file
FILE ESD "ESD.key"
!
! Title line
TITLE "Riparian Zone Model for the SAC Rev. 0 Initial Assessment"
!
! User name
USER "name"
!
! Realizations to process
REALIZATION ALL
! REALIZATION LIST 1
! REALIZATION RANGE 1 - 11
!
! Define the impact type solutions to compute
IMPACTS HUMAN
IMPACTS ECOLOGICAL
IMPACTS ECONOMIC
IMPACTS CULTURAL
!
! Define the analytes to be processed
ANALYTE ID="H3"
ANALYTE ID="Sr90"
!
! Debug outputs are not requested
!DEBUG SOIL SEEP KD DF
!
! Actually execute the problem setup
EXECUTE
!
! Human Impacts: Riparian zone location matching
LOCATION HUMAN PRIMARY="HL0417" SECOND="HL0418" DF="DF5m  "
LOCATION HUMAN PRIMARY="HL0419" SECOND="HL0420" DF="DF5m  "
!
! Human Impacts: Soil-Water partition coefficients
KDSOIL LOCATION="HL0417" ANALYTE="H3    " KDSOIL="KDH   " HUMAN
KDSOIL LOCATION="HL0417" ANALYTE="Sr90  " KDSOIL="KDSr  " HUMAN

! Ecological Impacts: Riparian zone location matching
LOCATION ECOLOGIC PRIMARY="EL0001" SECOND="EL0002" DF="DF5m  "
LOCATION ECOLOGIC PRIMARY="EL0003" SECOND="EL0004" DF="DF5m  "
LOCATION ECOLOGIC PRIMARY="EL0005" SECOND="EL0006" DF="DF5m  "
!
! Ecological Impacts: Soil-Water partition coefficients
KDSOIL LOCATION="EL0001" ANALYTE="H3    " KDSOIL="KDH   " ECOLOGICAL
KDSOIL LOCATION="EL0001" ANALYTE="Sr90  " KDSOIL="KDSr  " ECOLOGICAL
!
KDSOIL LOCATION="EL0003" ANALYTE="H3    " KDSOIL="KDH   " ECOLOGICAL
KDSOIL LOCATION="EL0003" ANALYTE="Sr90  " KDSOIL="KDSr  " ECOLOGICAL
!
KDSOIL LOCATION="EL0005" ANALYTE="H3    " KDSOIL="KDH   " ECOLOGICAL
KDSOIL LOCATION="EL0005" ANALYTE="Sr90  " KDSOIL="KDSr  " ECOLOGICAL
!
! End of keywords
END
```

3.4.2 Output Files and Diagnostics

ESP creates potentially thousands of files that are used as input to the release and transport modules identified on the left side of Figure 1-2. Many of these files contain generated values for stochastic variables. The simulated input values are written to files to be used later during uncertainty analyses. The resulting file names are specified in the individual stochastic definition files (using the DEBUG STOCHASTIC keyword).

The output from each release or transport module is used for input into another module, for diagnostics for the assessment, or for analyzing intermediate assessment results. This section describes the outputs used by ESP.

3.4.2.1 Results Used by ESP from the INVENTORY Code

The INVENTORY module creates a file for each realization that contains the releasable inventory for every simulated year until site closure, for all analytes, and for all release modes. These files (e.g.,

…/assessment/inventory/inv01.res) are read by VADER to get the amount to release for each analyte. Because INVENTORY is typically run in a standalone mode, the ESP does not directly read any outputs from the INVENTORY module.

3.4.2.2 Results Used by ESP from the VADER Code

VADER reads a STOMP input template file created by ESP (input-esp). It then inserts release information and creates the STOMP input file (e.g., …/assessment/600-148/H3/01/input). If the site is a remediation site, VADER will calculate the amount being transferred out and write it to the vader.rem file. This is done for all combinations of sites, analytes, and realizations. ESP checks for the existence of any requested remediation files at the completion of a VADER run.

If VADER completes its analysis without any problems, it creates a vader.done file. However, if VADER fails for any reason, it will create a vader.fail file. These files are used by ESP to make sure all of the modeling is done error free or to log any errors that occur.

3.4.2.3 Results Used by ESP from the STOMP Code

STOMP reads the file created by VADER (input). STOMP will write the time-dependent analyte mass (chemicals) or activity (radionuclides) being released into the groundwater (aquifer) into the release file. If the site is a remediation site, STOMP will calculate the analyte mass (chemicals) or activity (radionuclides) being transferred out and write it to the remediate file. ESP checks for the existence of any requested remediation files at the completion of a STOMP run.

If STOMP completes its analysis without any problems, it creates a stomp.done file. However, if STOMP fails for any reason, STOMP will create a stomp.fail file. These files are used by ESP to make sure all of the modeling is done error-free or to log any errors that occur.

3.4.2.4 Results Used by ESP from the VZDROP Code

VZDROP reads all of the STOMP release files (for each site) and inserts anything being released into the aquifer into the CFEST input file (e.g., …/assessment/cfest/H3/01/input.l3i). This is done separately for each analyte and realization. Before VZDROP can be executed, ESP checks for the existence of all STOMP result files (release) – there should be one release file for each site. All files must exist or ESP will not continue.

If VZDROP completes its analysis without any problems, it creates a vzdrop.done file. However, if VZDROP fails for any reason, VZDROP will create a vzdrop.fail file. These files are used by ESP to make sure all of the modeling is done error free.

3.4.2.5 Results Used by ESP from the CFEST Code

CFEST reads and modifies several binary files. The main result files, which are used by GWDROP to create MASS2 input files, are the tabular data for the flow and mass transfer into the river. Two such tab files are

```
…/assessment/cfest/H3/01/hheldq001.tab
…/assessment/cfest/H3/01/hheldm001.tab
```

These files must exist or ESP will not continue.

3.4.2.6 Results Used by ESP from the GWDROP Code

GWDROP reads the result files from CFEST, creates input files for MASS2 (TMS files), and places the groundwater concentration data into the ECDA. If GWDROP completes its analysis without any problems, it creates a gwdrop.done file. However, if GWDROP fails for any reason, GWDROP will create a gwdrop.fail file. These files are used by ESP to make sure all of the modeling is done error free.

3.4.2.7 Results Used by ESP from the MASS2 Code

For each MASS2 run (one for each analyte and realization combination), a results file is created. This file contains the annual average concentrations in the river for each node for every modeled year. The file name and location of this file are illustrated by the following file name:

```
…/assessment/mass2/H3/01/crdrop/H3_0001.RIV
```

The ESP checks for existence of the specified river file at the completion of each MASS2 run.

3.4.2.8 Results Used by ESP from the CRDROP Code

CRDROP reads the MASS2 result river file and inserts concentration results into the environmental concentration data files for each impact type. The only output from this module is a report file (which appears in the …/crdrop subdirectory). If CRDROP completes its analysis without any problems, it creates a crdrop.done file. However, if CRDROP fails for any reason, it will create a crdrop.fail file. These files are used by ESP to make sure all of the modeling is done error free.

3.4.2.9 Results Used by ESP from the RIPSAC Code

RIPSAC is the last module to run under control of the ESP. It runs only after the groundwater and river modules have completed. It reads the ECDA, performs its analyses, and inserts results back into the ECDA. The only output from this module is a report file (which typically appears in the ripsac subdirectory – which is specified in the ripsac.key file). When RIPSAC is finished, it writes a ripsac.done file. This file will contain the string “Normal termination” if the run was successful and the string “Error termination” if the run did not complete properly. This file is used by ESP to make sure all of the modeling is done error free.

4.0 INVENTORY – Inventory Tracking and Disposition

This section describes the operation of the inventory model.

4.1 Purpose

The purpose of the inventory code is to provide annual movement of contaminants into the environment, thus becoming available for liquid release or waste form dissolution and transport. The information is provided annually for every contaminant in possibly several different waste forms at every waste site. All disposal actions are treated in a stochastic manner and separate result files are generated for each realization.

4.2 Algorithms and Assumptions

The primary analyses performed by the Inventory program are evaluation of stochastic realizations, aggregation of the disposal actions, accounting for radioactive decay, and optionally, normalization of results to expected total amounts. The aggregation and normalization actions involve simple summation of values and normalization to an expected value and are not described further.

The inputs to the program describe a series of disposal actions. Each disposal action is identified by a site name, a year, the volume disposed, and the concentration of each radionuclide in the volume. The volume and concentration parameters are defined as stochastic variables. The program generates sample values for each parameter with the number of values being the number of realizations requested on the REALIZATION keyword (see Section 2.1.20) in the ESD keyword input file.

Radioactive decay is performed using a general decay algorithm to describe chain decay with branching (Strenge 1997). This algorithm, presented below, is written with quantities defined in units of mass rather than activity. The analysis of each disposal action first involves calculation of the activity disposed, by multiplying the generated realization values for concentration and volume. To adjust the disposed activity to a future year, the activity is first converted to a quantity proportional to the total mass present (in order to implement the decay algorithm). This is accomplished by dividing the activity by the radiological decay constant for the radionuclide, as follows:

$$A_c(t) = k\, Q_c(t) \;/\lambda_c$$

where:

$A_c(t)$ = quantity of radionuclide c present at time t in mass units (mass units)
k = proportionality constant between mass units and activity units (mass units / activity units)
$Q_c(t)$ = activity of radionuclide c present at time t (activity units)
λ_c = decay constant for radionuclide c.

The value of the proportionality constant is irrelevant because the conversion is reversed after the decay calculation has been completed and the effects of the constant are cancelled. A value of K = 1 is used in the code.

The decay calculation starts with the initial quantity ($A_c(0)$ from the above equation) for each radionuclide in the decay chain being evaluated. The decay algorithm has four equations, as follows.

$$A_c(t) = \sum_{i=1}^{c} K_{ci}\, e^{-\lambda_i t}$$

$$K_{11} = A_1(0)$$

$$K_{cn}\,(n = 1 \to c - 1) = \frac{\sum_{i=n}^{c-1} d_{ic}\, \lambda_i\, K_{in}}{\lambda_c - \lambda_n}$$

$$K_{cc} = A_c(0) - \sum_{n=1}^{c-1} K_{cn}$$

where:

$A_c(t)$ = quantity of chain member c at time t (atoms)
λ_c = radioactive transition rate constant for chain member c (d^{-1})
d_{ic} = fraction of precursor radionuclide transitions (chain member i) that result in production of the chain member c (dimensionless)
K_{cn} = decay coefficient for term n of chain member c (atoms).

The chain decay data are read from the radionuclide master data library file described in Section 4.5.1.5. This data library provides values for the number of chain members, d_{nc} and λ_c.

4.3 Code Execution Environment

This section describes how to run INVENTORY.

4.3.1 Location in Processing Sequence

INVENTORY is executed one time, before the execution of all the other codes in the processing sequence (see Figure 1-2). Although INVENTORY can be run under the control of the ESP, in practice the code is executed outside the control of the ESP and its results checked prior to performing the subsequent analysis steps. If necessary, results are then placed in the appropriate directory for access by other codes.

4.3.2 How INVENTORY Is Invoked

INVENTORY can run either under the Windows operating system or the Linux operating system. Under Windows (98, NT, or 2000), INVENTORY executes in a DOS box. To start INVENTORY, enter the following command line:

```
INVENTORY ‘ESDfilename’ runid
```

Under Linux, INVENTORY is executed through the following Bourne Shell or C Shell command:

```
   inventory.exe ‘ESDfilename’ runid
```

Under Linux, file names are case sensitive; under Windows, they are not.

For these command lines, INVENTORY or inventory.exe is the name of the executable program; ESDfilename is the name of the ESD keyword file, and runid is the date and time stamp for the run. The ESD keyword file name may contain path information in addition to the file name. The runid parameter is composed of year, month, day, hour (using a 24-hour clock), minute, and second for the run. For example, a run on May 21, 2001 at 20 seconds after 1:05 PM would have a runid entry of 20010521130520. The INVENTORY input files must be located in, and result files are written to, a subdirectory named *inventory* under the directory where the ESD file resides.

4.4 Keyword Descriptions for INVENTORY

Code control input and most input data are provided in three keyword files: the file inventory.key, the Master Waste Stream Disposal Action file, and the ESD keyword file (file name is provided on the command line). Section 2.1 describes the ESD file keywords. Table 4.1 summarizes the keywords used in INVENTORY. The following sections describe the keywords used in the inventory.key and the Master Waste Stream Disposal Action files.

Table 4.1 Summary of Keywords Used by INVENTORY

Keyword	**Source**	**Information Used by INVENTORY**
AGGREGAT	ESD keyword file	Selection of aggregation sites to include in the analysis.
ANALYTE	ESD keyword file	Analyte selection using modifiers ID, TYPE, and COMPUTE.
BALANCE	ESD keyword file	Selection of analytes for which a mass balance analysis will be performed.
DEBUG	inventory.key	Selection of options for writing of intermediate stochastic analysis information to an output file.
DECAY	inventory.key	Specification of handling of radioactive decay in the analysis.
DISPOSAL	Master Waste Stream Disposal Action file	Identifies the volume and concentration of all disposal actions.
END	ESD keyword file, Disposal Action file, and inventory.key	Signifies end of a keyword file.
EXECUTE	inventory.key	Causes complete execution of INVENTORY.
FILE	inventory.key	Specification of file names for input and output.
MASSBALANCE	inventory.key	Specifies that a mass balance analysis will be performed.

Keyword	Source	Information Used by INVENTORY
NORMALIZE	inventory.key	Specification of expected total quantities for analytes.
PERIOD	ESD keyword file	Specification of start and end time for the analysis.
REALIZATION	ESD keyword file	Specification of number of realizations in analysis.
SEED	inventory.key	Specification of random number seeds for generation of stochastic values.
SELECT	inventory.key	Selection of waste types and waste streams to include.
TIMER	inventory.key	Causes execution time information to be generated and written to the output file.
TITLE	inventory.key	Title of inventory.key keyword input file.
USER	inventory.key	User responsible for inventory.key keyword input file.
WASTEMAP	inventory.key	Specification of mapping of waste streams to aggregation sites.

4.4.1 DEBUG Keyword for INVENTORY

The DEBUG keyword is used to specify generation of summary information related to stochastic variables used in the analysis. Three modifiers are used to select specific aspects of data to be written to the stochastic output file (file name specified using the FILE keyword, Section 4.4.6). The modifier DEFINITI causes the variable stochastic definitions to be written to the stochastic output file. The modifier STOCHAST causes the variable stochastic values generated for use in the analysis to be written to the stochastic output file. The modifier STATISTI causes the statistical representation of the generated values to be written to the stochastic output file. The statistical representation information can be compared to the input distribution to determine if the representation adequately simulates the desired distribution of values. The syntax for the keyword is as follows:

```
DEBUG [DEFINITI | STOCHAST | STATISTI]
```

All desired modifiers can be placed on one keyword record or multiple keyword records can be provided with one modifier per record. The following example DEBUG keyword record set specifies all debug output options:

```
DEBUG STOCHAST
DEBUG DEFINITI STATISTI
```

There are no quote strings associated with the DEBUG keyword.

4.4.2 DECAY Keyword for INVENTORY

The DECAY keyword is used to specify the method for accounting for radioactive decay for the input radionuclide activities defined for each disposal action. When the modifier NONE is present on the DECAY keyword record, then the activities disposed are assumed to be defined for the time at which the disposal action occurs, as defined on the disposal action record. In this case, no decay correction is made for the disposal action. However, for purposes of the normalization analysis, a decay correction is applied to the time at which the normalization is to be made so the summation of activities is performed on records with same decay date.

When the modifier NONE is absent, a numerical value must be provided giving the year when all radionuclide activity quantities (e.g., concentrations) are defined. The value must be greater than 1943 or an error message is written and execution is terminated.

The syntax for this keyword record is as follows:

```
   DECAY [NONE | nnnn]
```

Where nnnn is a year greater than 1943. Decimal years are allowed so 1955.5 would represent July 1, 1955 as follows:

```
   DECAY 1955.5
```

There are no quote strings associated with the DECAY keyword.

4.4.3 DISPOSAL Keyword for INVENTORY

The DISPOSAL keyword is used in the Master Waste Disposal Action file to define the volume and concentration of each disposal action. Each DISPOSAL keyword record provides information for one disposal action. A disposal action is associated with one waste site, one waste type, and one year. When waste is disposed to the same site for more than one year, additional records must be provide for each year of disposal. The same is true for disposal of more than one waste type to the same location. The syntax for the DISPOSAL keyword record is the following:

```
   DISPOSAL [WASTEID “waste site name”] [TYPE “waste type”] [YEAR nnnn]
      [VOLUME N1 P1 {P2} {P3}…] [CONTAM1 “analyte name1” N1 P1 {P2} {P3}
      {CONTAM2 “analyte name2” N1 P1 {P2} {P3}…}]
```

Modifiers are used to provide the specific information on the record as described in Table 4.2. Each of the modifiers other than the CONTAMi modifier must appear only once on the keyword record. At least 1 CONTAMi modifier must be present for each DISPOSAL keyword. Section 11.0 provides the stochastic parameter representation.

Table 4.2 Modifiers Used with the DISPOSAL Keyword Record

Modifier	**Description**
CONTAMi	This modifier is composed of the letters *CONTAM* and an integer between 1 and 99 (no spaces). The actual value of the integer is not significant except that it must be unique for each CONTAMi modifier present on one DISPOSAL keyword record. Following the modifier is the name of the analyte (six characters in double quotation marks) and numbers defining the concentration of the analyte (in stochastic parameter representation) for the current disposal action.
TYPE	This modifier is followed by the waste type (six characters in double quotation marks) for the current disposal action.

Modifier	Description
VOLUME	This modifier is followed by the volume (in stochastic parameter representation) for the current disposal action.
WASTEID	This modifier is followed by the waste site name (15 characters in double quotation marks) for the current disposal action.
YEAR	This modifier is followed by a single number (four characters) giving the year the current disposal action occurs.

Table 4.3 provides example DISPOSAL keyword records. These example records define analyte concentrations using the lognormal (base e) stochastic distribution model (number 9) and volumes using a triangular distribution (number 6).

Table 4.3 Example DISPOSAL Keyword Records

```
DISPOSAL WASTEID “100-K-42” TYPE “SF” YEAR 1998
  VOLUME  6  5.6  7.0  8.4
  CONTAM1  “Sr90“   9  1.21E+01 2.46E-01
  CONTAM3  “Cs137“  9  1.32E+01 2.46E-01
  CONTAM6  “H3“  9  3.16E+00 2.46E-01
  CONTAM13 “U“  9  7.08  0.246
DISPOSAL WASTEID “100-K-42” TYPE “SF” YEAR 1999
  VOLUME  6  3.6  5.0  7.4
  CONTAM1  “Sr90“   9  1.21E+01 2.46E-01
  CONTAM3  “Cs137“  9  1.32E+01 2.46E-01
  CONTAM6  “H3“  9  3.16E+00 2.46E-01
  CONTAM99 “U“  9  7.08  0.246
DISPOSAL WASTEID “216-W-16” TYPE “liquid” YEAR 1956
  VOLUME  6  700.  900.  1200.
  CONTAM2  “Sr90“   9  -2.01E+01 1.27
  CONTAM3  “Cs137“  9  -4.59E+01 1.27
  CONTAM4  “H3“  9  -9.02E+00 1.27
```

4.4.4 END Keyword for INVENTORY

The END keyword signifies the end of all keyword data. It should be the last keyword in the keyword file. Any data in the keyword file after the END keyword will be ignored. The following is the syntax for this keyword record:

```
END
```

There are no modifiers, numerical values, or quote strings associated with the END keyword.

4.4.5 EXECUTE Keyword for INVENTORY

The EXECUTE keyword (if present) causes a full analysis to be attempted. If the keyword is not present, then the analysis will terminate after reading the input data and performing consistency checks on the data. Running the code without the EXECUTE record is normally performed only to test the input data set. Once the data have been checked, the EXECUTE record should be present. When INVENTORY is to be exercised under the control of the ESP, the EXECUTE record must be present or the analysis will not proceed correctly. The following is an example EXECUTE keyword record:

```
EXECUTE
```

There are no modifiers, numerical values, or quote strings associated with the EXECUTE keyword.

4.4.6 FILE Keyword for INVENTORY

The FILE keyword is used to specify file names for selected input and output files. Modifiers are used to specify the file being defined and quote strings (up to 72 characters) are used to specify the file name. The syntax for this keyword record is as follows:

```
FILE [DECAY=“Decay_File”| MASS=“Mass_File” | STOCHOUT=“Stoch_File” |
   WASTE=“Waste_File”]
```

Table 4.4 indicates the files that can be specified. File names may include path information.

Table 4.4 Modifiers Associated with the FILE Keyword in the INVENTORY Keyword File

Modifier	**File**	**Description**
DECAY	Decay data file	This input file (see Section 4.5.1.5) contains radionuclide chain decay data. It is required even if the run does not include radioactive analytes.
MASS	Mass balance file	This is a required output file (see Section 4.5.2.3) containing the total release by radionuclide for each year for all realizations.
STOCHOUT	Stochastic data file	This is an optional output file (see the DEBUG keyword in Section 4.4.1 and the file description in Section 4.5.2.2) containing summary information on generated stochastic variables.
WASTE	Master waste stream disposal action file	This is a required input file containing waste stream disposal actions (see Section 4.5.1.3).

The following is an example FILE keyword record for defining a stochastic output file:

```
FILE STOCHOUT “stochvar.out”
```

Multiple FILE keywords must be entered. However, only one modifier and associated file name may be entered on a single FILE keyword record. There are no numerical values associated with the FILE keyword.

4.4.7 MASSBALANCE Keyword for INVENTORY

The MASSBALANCE keyword record is used to specify that a mass balance analysis will be performed. The details of the mass balance analysis are specified on the ESD keyword record BALANCE described in Section 2.1.4. When a mass balance calculation is to be performed, an output file name must be provided on a file keyword record (see Section 4.4.6). If the MASSBALANCE keyword is not entered, then no mass balance calculations will be performed. The syntax for this keyword record is as follows:

```
MASSBALANCE
```

There are no modifiers, numerical values, or quote strings associated with the MASSBALANCE keyword record.

4.4.8 NORMALIZE Keyword for INVENTORY

The NORMALIZE keyword is used to specify the expected amount of an analyte to be disposed from all disposal actions included in the scope of the current analysis. The amount is specified as a stochastic distribution. The syntax for this record is as follows:

```
NORMALIZE [“analyte name” N1 P1 {P2} {P3} … {Pn}]
```

The “analyte name” is the name of an analyte as specified on the ANALYTE keyword record in the ESD file (see Section 2.1.2), N1 is the index of the stochastic distribution model to be used to define the expected distribution of the total amount of the analyte. The parameters P1, P2, P3, etc. are the parameters necessary to define the distribution model. See Section 11.0 for details of the stochastic distribution options. The total amount for radionuclides is defined at the time of closure as given on the PERIOD keyword record in the ESD file (see Section 2.1.18). The following is an example NORMALIZE keyword using a triangular distribution to define the total activity of Sr90 in units of curies at the time of closure:

```
NORMALIZE “Sr90” 6  1.6E+8  2.0E+8  2.4E+8
```

Multiple NORMALIZE keywords can be entered. Data for only one analyte may be specified on each NORMALIZE keyword record. There are no modifiers associated with the NORMALIZE keyword record.

4.4.9 SEED Keyword for INVENTORY

The SEED keyword is used to provide initial seed values for the random number generator used to generate the stochastic values for each realization. One seed value is provided for use in generation of volume and fraction values, and additional seeds are provided for each analyte (one per analyte). The following is the syntax for this keyword record:

SEED [V_1] [“$analyte_1$” S_1] {“$analyte_2$” S_2} … {“$analyte_n$” S_n}

The first numerical value, V_1, provides the seed for volume and fraction realization value generation. The remainder of the entries are presented in data pairs with an analyte name (in double quotation marks) followed by the random seed value to be used for that analyte. Values for all analytes must be provided on the SEED keyword record, or the program will error terminate. Only one SEED keyword record should be included in the keyword file.

An example SEED keyword is as follows:

```
SEED 199044 “Pu239” 23923 “Sr90” 9090
```

This SEED keyword record specifies a seed of 199044 for use in generation of volume and fraction realization values, a seed of 23923 for generation of concentration values for Pu239, and a seed of 9090 for generation of concentration values for Sr90. The entries for each seed should be in the range 1 to 999999. The entries are read using single precision and then moved to a double-precision variable. There are no modifiers associated with the SEED keyword record

4.4.10 SELECT Keyword for INVENTORY

The SELECT keyword record is used to specify waste types and waste streams to be included in the analysis. Specification of waste streams is optional; if it is not provided, then all waste streams listed in the input files are included in the analysis. The syntax for this keyword record is as follows:

```
SELECT [TYPE N “type1” “type2” … “typeN” | WASTE  “filename”]
```

The waste types must be specified on a SELECT TYPE keyword record. If the TYPE modifier is present, then the number of types to be specified “N” is also given and N type names are also provided. The TYPE and WASTE modifiers cannot be specified on the same SELECT keyword record. The following example SELECT keyword selects four waste types:

```
SELECT TYPE 4 “liquid” “glass” “cement” “soil”
```

If the WASTE modifier is present, then a file name is provided that indicates the source of waste stream names to be included in the analysis. Section 4.5.1.4 describes the content of this file. The following example SELECT keyword identifies a file:

```
SELECT WASTE “waste.inp”
```

4.4.11 TIMER Keyword for INVENTORY

This keyword is used to cause run timing information to be written to the run output file (inventory.out). If the TIMER keyword record is present, then timing information is generated and printed; otherwise, it is not. The following is the syntax for this keyword record:

```
TIMER
```

There are no modifiers, numerical values, or quote strings associated with the TIMER keyword record.

4.4.12 TITLE Keyword for INVENTORY

The TITLE keyword is used to define a single-line problem title. Titles are provided in the ESD file, inventory.key file, and the master waste stream disposal action file. The problem title is written to the inventory.out output file. If the title is not supplied, the program will error terminate. The following is the syntax for this keyword record:

```
TITLE [“quote”]
```

The title is entered in a quote string, which must be enclosed in double quotation marks. Titles up to 72 characters long are supported. The following example defines a title for a run of the code.

```
TITLE “Example title line for the inventory keyword control file.”
```

There are no modifiers or numerical values associated with the TITLE keyword.

4.4.13 USER Keyword for INVENTORY

This keyword is used to identify the individual responsible for preparing input in the keyword file. USER records are provided in the inventory.key file and the master waste stream disposal action file. If the user name is not supplied, the program will error terminate. Syntax of the USER keyword record is the following:

```
USER “user name”
```

Where the “user name” is a string up to 20 characters long. The following is an example USER keyword:

```
USER “John Doe”
```

There are no modifiers associated with the USER keyword.

4.4.14 WASTEMAP Keyword for INVENTORY

The WASTEMAP keyword is used to specify aggregation rules for each of the disposal actions. The keyword specifies the waste stream names and the aggregation site that is to be associated with each waste stream. An aggregation site name must be included for each disposal action composed of a waste stream name (e.g., 216-A-11), a waste type (e.g., liquid), and a year of disposal (e.g., 1956). The WASTEMAP keyword record must also specify the fraction of the disposal action quantity that goes to the named aggregation site. This allows a disposal action to be divided between more than one aggregation site. A maximum of three aggregation sites may be used for each disposal action. The following is the syntax of the WASTEMAP keyword record:

```
WASTEMAP [AGGREGATE “aggregation site name”] [FRACTION N1 P1 {P2} {P3} …]
   [STREAMS “stream-type-year-string1” {“stream-type-year-string2”} … ]
```

The modifier AGGREGATE is always required and is followed by the aggregation site name, declared as a string of up to 15 characters. The modifier FRACTION is always required and is followed by a stochastic parameter definition, where N1 is the index of the stochastic distribution type as defined in Section 11.1. This value is followed by parameters necessary to define the distribution for the FRACTION. To define a constant value of 1.0 (indicating all of the disposal action goes to the current aggregation site), set N1 = 1 and P1 = 1.0. The modifier STREAMS is always required and must be followed by at least one string entry identifying a waste disposal action. The construction of the string must be performed as follows: the waste stream name is in character positions 1-15 (left justified); the waste type name is in character positions 16-21 (left justified); and the year is in character positions

22-25. The string is enclosed in double quotation marks. Table 4.5 provides example WASTEMAP keyword records.

Table 4.5 Example WASTEMAP Keyword Records

```
WASTEMAP AGGREGAT “218-E$B6” FRACTION 1 1.0 STREAMS
  “218-E-12A      soil  1998”
  “218-E-12A      soil  1999”
WASTEMAP AGGREGAT “218-E$B6” FRACTION 1 0.4 STREAMS
  “218-E-12B      soil  1998”
  “218-E-12B      soil  1999”
WASTEMAP AGGREGAT “218-E$B5” FRACTION 1 0.6 STREAMS
  “218-E-12B      soil  1998”
  “218-E-12B      soil  1999”
```

This example shows disposal of soil wastes from site 218-E-12A to aggregation site 218-E$B6 in years 1998 and 1999. Also disposed to the same aggregation site is 40% of waste stream 218-E-12B during years 1998 and 1999. The remainder of this waste stream is disposed to aggregation site 218-E$B5. Note that when fractional disposal is employed, the fraction totals should add to 1.0, and the same waste sites must be defined on both WASTEMAP records when the FRACTION option is employed (i.e., the fraction is different than 1.0). If the fractions do not add to 1.0, the program will adjust the values proportionately so that they do add to 1.0 for each realization.

4.5 Data Files

The inventory program reads from up to five input files depending on the selections made in the inventory.key input file, generates seven types of output files, and one temporary file. Table 4.6 summarizes these files. The following sections describe the details of these files.

Table 4.6 Files Used by INVENTORY Code

File	Use	Description
Inventory keyword file	Input	Primary source of run control information, scope of analysis, and disposal site aggregation rules.
ESD file	Input	Source of general information for the overall analysis.
Disposal actions	Input	Source of detailed disposal action data by site, year, and analyte.
Waste site list	Input	Optional input file to specify disposal sites that will be included in the analysis.
Radionuclide decay data file	Input	File containing data for evaluation of radionuclide chain decay with branching.
Stochastic data	Output	File for writing output information on stochastic variables. Content dependent on user selections in inventory.key file.
Analysis summary	Output	File named inventory.out that contains a summary of input option selections, files accessed, and error messages (if any).

File	Use	Description
Mass balance results	Output	If mass balance was selected to be performed (MASSBAL in inventory.key file), then an output file is generated for saving the results of the mass balance. The file name is provided in the inventory.key file.
Result files	Output	One file for each realization is generated giving the quantity of analytes disposed at each disposal aggregation site for each year, by waste type.
Total result file	Output	This output file contains all of the results and is used only by the data extractor INGRAB to analyze output from the INVENTORY run. The file is not used by other SAC codes. The file name is INVENTORY.ALL.
Normalization file	Output	Results of the normalization analysis are written to this file, which contains the normalization factor for each analyte for each realization. The file is always written, even if normalization was not requested in the inventory.key file. The file name is INVENTORY.NRM.
Status file	Output	At the completion of an analysis, a file is written to indicate the status of the analysis. If the analysis completed without error, the file inv.done is generated. Otherwise, the file inv.fail is generated.
Active run file	Temporary	This file is opened when the INVENTORY run begins and is closed (and deleted) when the run is ready to be terminated. The presence of the file signals the ESP that the code is still running. The file name is inv.run.

4.5.1 Input Files

This section describes the five input files used by INVENTORY.

4.5.1.1 Description of the INVENTORY Keyword File

The INVENTORY keyword file provides parameters to define the scope of the analysis and rules for aggregation of the waste disposal streams. Section 4.4 defines the keywords for this file. The input is read using the keyword format syntax as described in Section 10.0.

The TITLE and USER keyword records must also be present with non-blank quote strings or the code will error terminate. Table 4.7 contains an example keyword file.

Table 4.7 Example inventory.key File

```
! Example inventory.key file
USER "DL Strenge"
TITLE "History Matching INVENTORY input file"
EXECUTE
DECAY NONE
DEBUG DEFINITI
MASSBALA
SELECT TYPE 7 "Liquid" "glass" "cement" "sludge" "core" "Soil" "cake"
```

```
FILE DECAY "rmdlib.dat"
FILE WASTE "MEDIAN.DAF"
FILE MASS "MASSOUTN.MAS"
FILE STOCHOUT "stochvar.out"
SEED 1990333 "Pu239" 2394234 "Sr90" 9090909. "Cs137" 137137.
 "U" 7774449 "H3" 3333333 "CCl4" 14141414
 "TC99" 9919919 "Cr(VI)" 7744558 "CrVI" 7744558 "I129" 129129
 "U238" 2362362
! The following are WASTEMAP AGGREGATION Keyword records
WASTEMAP AGGREGAT "241-S-112" FRACTION 1 1.0 STREAMS
 "241-S-112      Liquid2003"
WASTEMAP AGGREGAT "241-S-102" FRACTION 1 1.0 STREAMS
 "241-S-102      Liquid2003"
WASTEMAP AGGREGAT "241-C-104" FRACTION 1 1.0 STREAMS
 "241-C-104      Liquid2003"
WASTEMAP AGGREGAT "241-S-105" FRACTION 1 1.0 STREAMS
 "241-S-105      Liquid2007"
WASTEMAP AGGREGAT "UPR-200-E$A6-6" FRACTION 1 1.0 STREAMS
   "UPR-200-E-141  Liquid1984"
WASTEMAP AGGREGAT "UPR-200-E$B3-12" FRACTION 1 1.0 STREAMS
   "UPR-200-E-7    Liquid1954"
WASTEMAP AGGREGAT "UPR-200-E$B3-12" FRACTION 1 1.0 STREAMS
   "UPR-200-E-7    Liquid1954"
WASTEMAP AGGREGAT "atmosphere" FRACTION 1 1.0 STREAMS
   "B_PLANT_FILTER gases 1944"
   "B_PLANT_FILTER gases 1945"
WASTEMAP AGGREGAT "216-B-15" FRACTION 1 1.0 STREAMS
 "216-B-15       Liquid1956"
 "216-B-15       Liquid1957"
WASTEMAP AGGREGAT "216-B-16" FRACTION 1 1.0 STREAMS
 "216-B-16       Liquid1956"
WASTEMAP AGGREGAT "216-Z-1A" FRACTION 1 1.0 STREAMS
   "216-Z-1A       Soil  1964"
   "216-Z-1A       Soil  1965"
   "216-Z-1A       Soil  1966"
   "216-Z-1A       Soil  1967"
   "216-Z-1A       Soil  1968"
   "216-Z-1A       Soil  1969"
END
```

4.5.1.2 Data Used from the ESD Keyword File in INVENTORY

INVENTORY uses some of the information provided in the ESD keyword file. Section 2.1 describes the keyword records for this file. Table 4.8 lists the keywords read from the ESD file by INVENTORY. The keywords may contain more information than listed in the table; however, the table only addresses the information used in INVENTORY. The input is read using the keyword format syntax described in Section 10.0.

Table 4.8 ESD File Keyword Information Used by INVENTORY

Keyword	Modifiers/Parameters Used	Description of Use
AGGREGATION	COMPUTE	Signifies that this aggregation site is to be included in the scope of the current analysis.
	ID	Identification name for the aggregation site, used for comparison with input from other files.

Keyword	Modifiers/Parameters Used	Description of Use
	OFFSITE	Indicates final disposal is off the Hanford Site. Disposal quantities are included in mass balance and normalization but not in disposal to onsite locations.
	TITLE	Descriptive title for the aggregation site.
ANALYTE	COMPUTE	Flag to cause the analyte to be included in the current analysis.
	ID	Analyte name used to identify input from the disposal action file.
	NAME	Descriptive name of analyte, used for information only.
	TYPE	Identifies an analyte as a chemical or radionuclide.
	HALFLIFE	Half-life of a radionuclide (days).
BALANCE	ALL	Presence of the keyword indicates that mass balance information is to be generated and written to a file. If the modifier ALL is present, then the information is written for all years.
	Years1, years2, …	If modifier ALL is not present, then the years for which information is to be written are supplied following the keyword name.
END	None	Signifies the end of the ESD file and reading of records is terminated.
OS	String	Operating System name *windows* or *unix*; default is *unix*.
PERIOD	START	Indicates the beginning year covered by analysis.
	STOP	Indicates the end year covered by the analysis.
	CLOSURE	Indicates year of closure for the site.
REALIZATION	Number	Provides the number of stochastic realizations to be included in the analysis.
TITLE	String	Descriptive title for the ESD file.

The period of analysis is defined using the PERIOD keyword record. Because the analyses performed by INVENTORY are needed only during the time when the site is open and remediation actions are continuing, the end of the period of analysis is limited to the lesser of the *closure* year and the *stop* year as provided on the PERIOD record.

The aggregation sites to be included in the analysis are defined by the AGGREGATE keyword records in the ESD file. The only sites included in the analysis are those included in the ESD file that have the COMPUTE modifier. If data for other aggregation sites are included in the inventory.key file, a warning is written that the site is not in the master list, but the analysis will continue. This allows the scope to be limited by deleting information in the ESD file, without the need to modify the inventory.key keyword file.

4.5.1.3 Description of the Master Waste Stream Disposal Action File

The Master Waste Stream Disposal Action file contains the details of the disposal actions for the Hanford Site and offsite transfers. The file is written in keyword format, with the keywords described in Section 4.4. Only four keywords are used in this file: DISPOSAL, TITLE, USER, and END. The input is read using the keyword format syntax as described in Section 10.0.

Each DISPOSAL keyword record provides the information for one disposal action. The record gives the volume disposed and the concentration of analytes in the disposal. The disposal action is defined by three parameters: waste site name (following modifier WASTEID), waste type name (following modifier TYPE), and year of disposal (following modifier YEAR). Each disposal action defined in this file (i.e., each combination of waste site, waste type, and year) must have a corresponding entry in the inventory.key file in a WASTEMAP keyword record. If there is no corresponding entry, then an error message is written and the analysis will error terminate.

The TITLE and USER keywords are required and will cause an error termination if a title string or user name string are not provided. Section 11.1 describes the syntax for providing the volume and concentration information as stochastic distributions. The file must end with an END keyword record. Table 4.9 provides excerpted records from a disposal action keyword file.

Table 4.9 Example Records from a Disposal Action Keyword File

```
TITLE "Example DAF file"
USER  "John Doe"
DISPOSAL WASTEID "241-S-112" TYPE "Liquid" YEAR 2003
 VOLUME   6 2.406E+01 3.008E+01 3.609E+01 TRUNCATE 0 1
 CONTAM4  "I129  "  9 -8.732E+00    0.246 TRUNCATE 0.01 0.99
 CONTAM5  "Cs137 "  9  4.049E+00    0.246 TRUNCATE 0.01 0.99
 CONTAM6  "U238  "  9 -6.847E+00    0.246 TRUNCATE 0.01 0.99
 CONTAM7  "Pu239 "  9 -4.030E+00    0.246 TRUNCATE 0.01 0.99
 CONTAM1  "H3    "  9 -3.004E+00    0.246 TRUNCATE 0.01 0.99
 CONTAM2  "Sr90  "  9  1.067E+00    0.246 TRUNCATE 0.01 0.99
 CONTAM3  "Tc99  "  9 -2.532E+00    0.246 TRUNCATE 0.01 0.99
 CONTAM8  "CrVI  "  9  7.364E-01    0.246 TRUNCATE 0.01 0.99
 CONTAM9  "U     "  9 -1.074E-01    0.246 TRUNCATE 0.01 0.99
DISPOSAL WASTEID "218-E-MELTER" TYPE "glass " YEAR 2016
 VOLUME   6 7.040E+00 8.800E+00 1.056E+01 TRUNCATE 0 1
 CONTAM5  "Cs137 "  9  5.427E+00    0.246 TRUNCATE 0.01 0.99
 CONTAM6  "U238  "  9 -4.408E+00    0.246 TRUNCATE 0.01 0.99
 CONTAM7  "Pu239 "  9 -3.639E-01    0.246 TRUNCATE 0.01 0.99
 CONTAM2  "Sr90  "  9  5.272E+00    0.246 TRUNCATE 0.01 0.99
 CONTAM3  "Tc99  "  9 -2.184E+00    0.246 TRUNCATE 0.01 0.99
 CONTAM9  "U     "  9  1.646E+00    0.246 TRUNCATE 0.01 0.99
DISPOSAL WASTEID "B_PLANT_FILTER" TYPE "Cement" YEAR 1974
 VOLUME   6 4.000E+08 5.000E+08 6.000E+08 TRUNCATE 0 1
 CONTAM1  "Sr90  "  9 -9.89E+00  2.46E-01 TRUNCATE 0.01 0.99
 CONTAM3  "Cs137 "  9 -1.07E+01  2.46E-01 TRUNCATE 0.01 0.99
 CONTAM5  "Tc99  "  9 -2.00E+01  2.46E-01 TRUNCATE 0.01 0.99
 CONTAM8  "Pu239 "  9 -2.36E+01  2.46E-01 TRUNCATE 0.01 0.99
 CONTAM14 "I129  "  9 -2.64E+01  2.46E-01 TRUNCATE 0.01 0.99
DISPOSAL WASTEID "US_Ecology" TYPE "Soil" YEAR 2050
 VOLUME   6 4.447E+03 5.559E+03 6.671E+03 TRUNCATE 0 1
 CONTAM1  "Sr90  "  9 -3.96E+00  2.46E-01 TRUNCATE 0.01 0.99
 CONTAM3  "Cs137 "  9 -1.72E+00  2.46E-01 TRUNCATE 0.01 0.99
 CONTAM5  "Tc99  "  9 -1.11E+01  2.46E-01 TRUNCATE 0.01 0.99
 CONTAM6  "H3    "  9 -1.56E+00  2.46E-01 TRUNCATE 0.01 0.99
 CONTAM8  "Pu239 "  9 -1.04E+01  2.46E-01 TRUNCATE 0.01 0.99
 CONTAM13 "U238  "  9 -8.38E+00  2.46E-01 TRUNCATE 0.01 0.99
 CONTAM14 "I129  "  9 -1.36E+01  2.46E-01 TRUNCATE 0.01 0.99
 CONTAM17 "U     "  9 -1.97E+00  2.46E-01 TRUNCATE 0.01 0.99
END
```

4.5.1.4 Description of the Waste Stream Selection File

The waste stream selection file is an optional file used to define the waste disposal sites that are to be included in the analysis. The file is read only if the inventory.key file contains a SELECT WASTE keyword record providing the file name for the waste stream selection file. Section 4.4.10 describes the SELECT keyword record.

The first line of the Waste Stream Selection file contains the number of sites to be defined (integer value), followed by a character string giving a descriptive title for the file. The remaining records each give the name of one waste site to be included in the analysis. The names are character strings with a maximum of 15 characters. The names must be unique. This list of waste stream names is compared against the names provided on the DISPOSAL keyword record for waste streams in the master disposal waste stream disposal action file. The names provided in the DISPOSAL record must match a name provided in the Waste Stream Selection file or an error message will be generated. Note that the test for a match of waste stream names is only performed if a Waste Stream Selection file has been identified on a SELECT WASTE keyword record in the inventory.key file. If not specified, then the analysis is performed for all waste streams defined on the DISPOSAL keyword records in the master disposal waste stream disposal action file.

Table 4.10 Example Waste Stream Selection File

```
34,SiteCode
100-B-10,
100-B-3,
100-B-5,
100-B-8,
100-C-3,
100-C-6,
116-B-1,
116-B-10,
118-F-4,
118-F-5,
118-F-6,
118-F-8,
216-A-11,
216-A-12,
216-C-6,
216-C-7,
218-E-15,
241-A-101,
241-A-151,
241-A-302A,
241-A-302B,
241-AN-101,
241-AN-102,
241-AZ-101,
241-AZ-102,
241-C-101,
241-C-102,
US_Ecology,
216-B-49,
216-B-5,
218-E-12B,
218-E-2,
218-W-4C,
218-W-7,
```

4.5.1.5 Description of the Radionuclide Master Decay Data File

The radionuclide master data library provides the decay data used to evaluate radioactive decay as a function of time. This is a sequential data file defining decay chains, radioactive half-lives, and branching fractions. The file is organized by decay chain with the order of data as defined in Table 4.11. Two types of progeny radionuclides are included in the file: explicit and implicit progeny. Explicit progeny are radionuclides of sufficient half-life that they are included explicitly in the decay chain. Implicit progeny are of short half-life and are not included explicitly in the decay chain. The database contains information on both types of progeny.

Table 4.11 Format of Radionuclide Master Decay Data File

Record	**Parameter type**	**Description**
Title	String in double quotes	Descriptive title for the file.
Second record	2 integer values	Integer 1: Maximum number of chain members contained in any decay chain. Integer 2: Maximum number of implicit progeny that follow immediately after the parent chain member.
Parent line	String, 2 integer values	String: Name of parent expressed in the same format as used for the ANALYTE ID on the ESD file keyword definition. Integer 1: Number of explicit chain members (including the parent) in the current decay chain. Integer 2: Number of implicit progeny that immediately follow the parent in the decay sequence.
Progeny lines	String, 6 numbers	String: Name of chain member expressed in the same format as used for the ANALYTE ID on the ESD file keyword definition. Number 1: Floating point value giving the radionuclide half-life in days. Values are provided only for explicit

Record	Parameter type	Description
		chain members; a blank is provided for implicit progeny. Number 2: Position of the chain member in the decay chain. The parent is position 1 and each explicit progeny is listed in sequential order. Implicit progeny have a blank for chain position. Number 3: Precursor chain member that decays to the current chain member (using the chain member position index provided in number 2 of preceding progeny lines). Number 4: Fraction of the precursor chain member decay events that result in generation of the current chain member. Numbers 5 and 6: Same as numbers 3 and 4 for another precursor to the current chain member (if any).

The parent line is immediately followed by the progeny records, with a progeny record provided for all chain members including the parent, implicit progeny, and explicit progeny. Table 4.12 provides an example data set for a parent and progeny.

Table 4.12 Example Data Set for the Radionuclide Master Decay Data File

```
Ra228      4  1
      RA228      2.10E+3 1 1 1.0000 0 0.0000
      AC228                1 1.0000 0 0.0000
      TH228      6.99E+2 2 1 1.0000 0 0.0000
      RA224      3.66E+0 3 2 1.0000 0 0.0000
      PB212      4.43E-1 4 3 1.0000 0 0.0000
```

This data set shows a chain with four explicit chain members (Ra228, Th228, Ra224, and Pb212). There is one implicit progeny (Ac228) that decays rapidly after each decay event for the parent radionuclide. The decay sequence data indicate that all chain members decay from the previous chain member at 100% (all decay fractions are 1.0).

Note: INVENTORY reads the decay database file name on the first reading of the inventory.key file and attempts to open the file before the analyte names are read from the ESD file. As currently programmed, there is no way to know if radionuclides are present when the decay file is opened (using the subroutine OPEN_FILES). If the decay database file name is not given, then an error message is written and the program terminates execution. Thus, this file must be provided even if the simulation run uses only nonradioactive analytes.

4.5.2 Output Files

Up to six files are generated for a run of INVENTORY. Table 4.6 summarizes these files, which are described in the following sections.

4.5.2.1 Analysis Summary File

The name of the analysis summary file is hard-wired to inventory.out. This file is used as the primary record of the analysis and includes summaries of the control information, scope of the analysis, timing information, and any warning or error messages generated. Table 4.13 provides excerpted records from an analysis summary file. Section 4.6 describes major error messages.

Table 4.13 Excerpted Records from the inventory.out File

```
Summary information from the Keyword Control File
   Complete execution of this analysis will be attempted
   Input activities defined at year of disposal
   Mass Balance requested.  Check years supplied on BALANCE record
   Radionuclide chain decay file is: rmdlib.dat
   Master Waste Stream Disposal Action file name is: disposal.daf
   Mass Balance Results Output file name is: massoutn.mas
   Stochastic data output file name is: stochvar.out


 III  N     N VV     VV EEEEE N     N TTTTT   OO   RRRR Y     Y
  I   NN    N  V     V  E     NN    N   T    O  O   R  RR Y   Y
  I   N N   N   V   V   E     N N   N   T   O    O R  RR  Y Y
  I   N  N  N   V   V   EEE   N  N  N   T   O    O RRRR    Y
  I   N   N N    V V    E     N   N N   T   O    O R R     Y
  I   N    NN    V V    E     N    NN   T    O  O  R  R    Y
 III  N     N     V     EEEEE N     N   T     OO   R   R   Y


                         INVENTORY    Version 1.25
                         Last Modified on 10 Dec 2001
                 ---------------------------------------------
                    Generation of Waste Disposal Quantities
                 ---------------------------------------------
                   Developed By Battelle Memorial Institute
                    Pacific Northwest National Laboratories
                             Richland, Washington
 Call BANNER

            Current Run ID = 20020320115700   User Name = DL Strenge

               System Date = 03-20-2002   System Time = 11:57:26.396


                               Review Signatures

Input Prepared By: ________________________________       Date: _______________

Input Reviewed By: ________________________________       Date: _______________
```

```
 Call CLAIM for QA output

           The software used to generate this output is experimental
              and has not been formally tested or peer reviewed.


 Environmental Settings Data file is: ESD_Initial2.key
 Set array dimensions for number of waste types
 Open files
 First reading Keyword Environmental Settings file

 The number of realizations specified for this analysis is:   25

 PERIOD Keyword results. Start time: 1944   Stop time: 2050
                         Closure: 2050      Number of years:    107
Operating System: UNIX,  Delimiter: /
 Second reading Keyword Environmental Settings file

Message originating in routine KEY_WORD
Message:  Warning: BALANCE year, too large for INVENTORY analysis
          Only years from START to CLOSURE on PERIOD record are analyzed
          CLOSURE  is: 2050, Found: 2100

 The following years have been selected for mass balance calculations
     1995     2000     2050

 The following analytes are included in the analysis
       ID     Type    Name
     Tc99     Rad.   Technetium 99
     H3       Rad.   Tritium
     Sr90     Rad.   Strontium 90
     I129     Rad.   Iodine 129
     Cs137    Rad.   Cesium 137
     U238     Rad.   Uranium 238
     Pu239    Rad.   Plutonium 239/240
     CCl4     Chem   Carbon Tetrachloride
     CrVI     Chem   Hexavalent Chromium

 The following aggregation release sites are defined for the analysis
     Name   Location
     atmosphere      Offsite
     store           Offsite
     100-B-10        Onsite
     100-B-5         Onsite
     100-B-8         Onsite
...
     UPR-300-@R6-2   Onsite
     UPR-300-@R6-4   Onsite
     US_Ecology      Onsite
 Second reading Keyword Control file

 Random seed provided for volume and fraction data is:       1990333.

 Random seeds provided for the following analytes
     Analyte       Seed      Analyte       Seed      Analyte       Seed
     Tc99       9919919.     H3         3333333.     Sr90       9090909.
     I129        129129.     Cs137       137137.     U238       2362362.
     Pu239      2394234.     CCl4      14141414.     CrVI       7744558.

 The following waste types are selected for this analysis
     Liquid
     glass
     cement
     sludge
     core
...
     cmnth2
     rxcomp
     TRUSLG
 Read Disposal Actions file

 Waste streams included in the analysis are as follows
   241-S-112         241-S-102         241-C-104         241-S-105         241-S-106
   241-S-107         241-C-107         241-S-101         241-C-102         241-C-112
   241-U-108         241-BY-111        241-U-107         241-S-110         241-S-108
   241-C-103         241-A-106         241-C-105         241-AX-103        241-A-102
   241-BX-104        241-SX-105        241-SX-103        241-TX-118        241-B-101
   241-AX-102        241-U-106         241-C-101         241-U-105         241-T-101
   241-S-104         241-TX-113        241-BY-105        241-A-101         241-BY-109
...
 Third reading Keyword Control file


 Normalization will be performed for the following contaminants
     None
 Read Aggregation data from Environmental Settings file
 Read chain decay data file

Title from Keyword Control File:
     History Matching INVENTORY input file
Title from Environmental Settings File:
     SAC Rev. 0 Initial2 Assessment
Title from Chain Decay Data file:
     MEPAS Chain Decay Database (19-Apr-2000) DL Strenge, MEPAS/HAZ Version
Title from Waste Disposal Action file:
     DAF with 11-Mar-2002 Uranium isotope revisions
```

```
User from Waste Stream Disposal Action file:
     DL Strenge
 Check input data for consistency
 Generate all realization values

Number of contaminants included in the analysis:      9
Number of waste volumes and concentrations:       28398
Number of aggregation data sets in the analysis:   1035
Total number of stochastic variables defined:     29442

Number of table entries for contaminants:       0
Number of table entries for volumes/conc.:   6090
Number of table entries for fractions:          0
Total number of table entries:               6090

 Check disposal fractions for all realizations
 Start analysis
 Starting analysis for realization number:    1

 Normalization factors for realization    1
 Analyte  Factor
 Tc99    1.000E+00
 H3      1.000E+00
 Sr90    1.000E+00
 I129    1.000E+00
 Cs137   1.000E+00
 U238    1.000E+00
 Pu239   1.000E+00
 CCl4    1.000E+00
 CrVI    1.000E+00
...
 Starting analysis for realization number:   25

 Normalization factors for realization   25
 Analyte  Factor
 Tc99    1.000E+00
 H3      1.000E+00
 Sr90    1.000E+00
 I129    1.000E+00
 Cs137   1.000E+00
 U238    1.000E+00
 Pu239   1.000E+00
 CCl4    1.000E+00
 CrVI    1.000E+00
```

4.5.2.2 Stochastic Data Summary File

The content of this optional file is determined by the keyword DEBUG records in the inventory.key file. If no DEBUG records are provided, then the file will not be generated. The file name is provided on keyword FILE STOCHOUT in the inventory.key file. Three types of output information can be included in the stochastic data summary file: stochastic distribution of input parameters, stochastic values generated for the analysis, and statistical analysis information for the generated values. These data sets are self-explanatory and are not described further in this section.

4.5.2.3 Mass Balance Result File

This file contains the result of the mass balance calculation for the years selected in the ESD file under keyword BALANCE. To have the file generated, it is also necessary that the keyword record MASSBALANCE be present in the inventory.key file and that a file name has been provided in the FILE MASS keyword record in the inventory.key file.

The file is composed of data sets. There is a large data set for each realization, and smaller data sets within the large set for each year included in the analysis. The realization data set starts with a line as follows:

```
   MASS BALANCE RESULTS for realization:   n
```

Where n is the realization number for the data set.

Within each realization data set, there is a data set for each year of the analysis. The following is the first record of each of these data sets:

```
   Total disposal amounts for year:  NNNN
```

Where NNNN is the year for the current data set. Following this record is a record for each analyte in the data set. A record is included for each analyte, even if the release is zero. Each of these records contains the analyte name, the amount released, and the units of the release amount (Ci for radionuclides and Kg for chemicals). The total release amounts are decayed to the end of the year indicated. Table 4.14 provides an example of part of a mass balance result file. This example shows two annual data sets in the large data set for realization 3.

Table 4.14 Example Mass Balance Result File

```
MASS BALANCE RESULTS for realization:    3
Total disposal amounts for year: 1956
CrVI   0.00E+00 Kg
H3     1.23E+04 Ci
C14    0.00E+00 Ci
Sr90   2.40E+01 Ci
Tc99   4.39E-01 Ci
Cs137  9.84E+00 Ci
CCl4   0.00E+00 Kg
Pu239  1.43E+02 Ci
Total disposal amounts for year: 1957
```

```
CrVI   2.01E+01 Kg
H3     7.11E+02 Ci
C14    0.00E+00 Ci
Sr90   2.73E+02 Ci
Tc99   4.89E-01 Ci
Cs137  1.27E+02 Ci
CCl4   4.22E+02 Kg
Pu239  4.97E+02 Ci
```

4.5.2.4 Release Result Files

The primary output of the analysis is generation of amounts of analytes disposed to each aggregation site as a function of time and waste type. One file is generated for each realization with the name of the output file coded to the realization number. The format of the file name is invn.res where n is an integer equal to the realization number represented by the file. If the total number of realization is between 1 and 9, then n has 1 character (1 to 9). If the total number of realization is between 10 and 99, then n has 2 characters (01 to 99). If the total number of realizations is between 100 and 999, then n has 3 characters (001 to 999). If the total number of realizations is between 1000 and 9999, then n has 4 characters (0001 to 9999). If the total number of realizations is between 10,000 and 99,999, then n has 5 characters (00001 to 99999). The maximum number of realizations supported is 99,999.

Table 4.15 describes the format of the .RES files.

Table 4.15 Format of Result Output Files (.RES)

Record	**Parameter / type**	**Description**
First record of file	Realization / integer	Number of the current realization for which results are presented in this file.
	Number of sites / integer	Number of sites for which data are provided in this file. There will follow one data set for each site (even if there are no positive disposals for the site).
First record of each data set	Aggregation name / string	Name of aggregation site for the current data set.
	Number of years / integer	Number of years for which data will be provided in this data set. If there are no releases, then this value is zero and no additional lines are provided for the data set.
Year record for data set	Year / integer	Year for which the current disposal data applies.
	Number of waste types / integer	Number of waste types for which releases are provided in this data set for this year.
	Number of analytes / integer	Number of analytes for which releases are provided in this data set for this year.
	Units / string	Units of volume for this data set (e.g., m^3)

Record	Parameter / type	Description
	Waste types names / strings and Volume / number	A waste type name and volume disposed are provided for each waste type to be included in this data set for this year. Values are provided in pairs (name and volume) for each waste type for the waste site and current year.
Analyte records for data set and year	Analyte name / string	Analyte ID for the current analyte. The number of these records that follow a year record is equal to the number of analytes defined on the year record.
	Units / string	Units of amounts for the analyte (e.g., Ci or Kg)
	Amounts / number	A value of total amount disposed is provided for each waste time defined on the year record. The values are written in the same waste type order as specified on the year record.

Table 4.16 provides an example of a result file. The table illustrates the first record of the file and four data sets.

Table 4.16 Example Result Output File Data Set

```
   21,  721  “Realization No., Number of release sites”
“atmosphere”, 0, “Offsite location, no data provided for this location”
“100-B-10”, 0, “Onsite location, number of years of data that follow”
“100-C-3”, 1, “Onsite location, number of years of data that follow”
1960, 1, 1, “m^3”, “Liquid”, 1.13, “ “,,,,,,,,,,,,,
“Cs137”, “Ci”, 2.00E-6,,,,,,,,,,,,,
“RMWFS”, 62, “Onsite location, number of years of data that follow”
1986, 1, 1, “m^3”, “Soilst”, 1.33E+1, “ “,,,,,,,,,,,,,
“Pu239”,”Ci”,4.28E+0,,,,,,,,,,,,,
1987, 2, 8, “m^3”, “Soilss”,7.02E+2,“Soilst”,1.29E+2,“ “,,,,,,,,,,,,,
“Sr90”,”Ci”,2.08E-7,0.00E+0,,,,,,,,,,,,,
“Tc99”,”Ci”,1.45E-1,0.00E+0,,,,,,,,,,,,,
“I129”,”Ci”,9.07E-9,0.00E+0,,,,,,,,,,,,,
“Cs137”,”Ci”,2.40E-7,0.00E+0,,,,,,,,,,,,,
“U238”,”Ci”,3.95E-0,3.15E-3,,,,,,,,,,,,,
“Pu239”,”Ci”,0.00E-0,1.33E+3,,,,,,,,,,,,,
“CrVI”,”Kg”,7.40E-0,0.00E+0,,,,,,,,,,,,,
“CCl4”,”Kg”,9.39E-0,0.00E+0,,,,,,,,,,,,,
...
```

4.5.2.5 Total Result File

This file is generated for use by data extractors to summarize results of the INVENTORY analysis. The file is not currently used by any other SAC codes. The first information in the file defines the scope of the analysis used to generate the data in the file. This header information includes the following:

- Title record with the runid from the INVENTORY command line (see Section 4.3.2)
- Number of realization in the analysis, defined in the ESD file

- Number of years of data, and specific years included in the analysis
- Number of aggregation sites, and aggregation site names included in the analysis
- Number of analytes and analyte names included in the analysis
- Number of waste types and waste type names included in the analysis.

The remainder of the records in the file contain disposal amounts for one combination of the above analysis variables: realization, year, aggregation site, waste type, and analyte. The two numerical values on the record are the disposal volume (m^3) and disposal amount (Ci for radionuclides and Kg for chemicals). Table 4.17 describes the specific format of the records. The data records are written in comma-separated format for easy import into programs such as Microsoft Excel.

Table 4.17 Description of Record Content for the Total Result File

Position on Record	Type	Description
1	Integer	Realization number for values on this record
2	Integer	Year number for values on this record. A value of 5 represents the fifth year in the list of years provided in the heading information data set.
3	Integer	Aggregation site index for values on this record. The index value is referenced to the list of aggregation names in the heading information data set. A value of 1 represents the first aggregation site.
4	Integer	Waste type index for values on this record. The index value is referenced to the list of waste type names in the heading information data set. A value of 3 represents the third waste type.
5	Integer	Analyte index for values on this record. The index value is referenced to the list of analyte names provided in the heading information data set. A value of 7 represents the seventh analyte in the list.
6	Number	Volume of waste disposed in this disposal action.
7	Number	Amount of waste disposed in this disposal action.

Table 4.18 provides an example of the data records for the total result output file. The example shows result records for realization 1 (first integer value), year 2 (second integer value), aggregation sites 415, 416, 417 (third integer value), waste type 1 (fourth integer value), and analytes 2, 3, 4, 5, 6, and 7 (fifth integer value). Each record ends with the volume and quantity values.

Table 4.18 Example Records for a Total Result Output File

```
   1,  2, 415, 1, 4, 8.859E+00, 8.476E-10,
   1,  2, 415, 1, 5, 8.859E+00, 5.569E-02,
   1,  2, 415, 1, 6, 8.859E+00, 6.409E-06,
   1,  2, 415, 1, 7, 8.859E+00, 2.814E-02,
   1,  2, 416, 1, 2, 9.722E+03, 2.859E-01,
   1,  2, 416, 1, 3, 9.722E+03, 1.251E-05,
   1,  2, 416, 1, 4, 9.722E+03, 3.641E-07,
   1,  2, 416, 1, 5, 9.722E+03, 3.696E-02,
   1,  2, 416, 1, 7, 9.722E+03, 2.556E+01,
   1,  2, 417, 1, 2, 1.148E+02, 8.932E-03,
   1,  2, 417, 1, 3, 1.148E+02, 1.392E-06,
   1,  2, 417, 1, 4, 1.148E+02, 3.562E-08,
   1,  2, 417, 1, 5, 1.148E+02, 1.820E-01,
   1,  2, 417, 1, 6, 1.148E+02, 3.262E-05,
   1,  2, 417, 1, 7, 1.148E+02, 7.272E-02,
```

4.5.2.6 Normalization Result File

The normalization analysis is performed by summing the total amounts released as of the time of site closure. The summed value is compared to the expected value (as defined by the input distribution on the NORMALIZE record in the inventory.key keyword file). If normalization has been selected for the analyte, then a normalization factor is calculated as the ratio of the expected value to the summed value. All disposal quantities in the current realization analysis for the analyte are multiplied by this normalization factor before being written to the result output file. The normalization result file is a compilation of the normalization factors generated for the analysis.

The first record of the file has the following information:

```
    “Normalization Factor Data File RUNID:  yyyymmddhhmmss”
```

Where yyyymmddhhmmss is the runid parameter from the INVENTORY command line. The second record of the file contains the number of analytes for which information is provided. The third record contains headings for columns that follow with the first column heading being *Realization*, followed by the analyte ID values for each analyte included in the analysis. The remainder of the file is one record for each realization, with the realization number followed by data for all analytes on each record. Table 4.19 provides an example normalization result file for a case with five analytes and eight realizations. For this example case, no normalization analyses were requested, so all normalization factors are 1.0. The file is written in comma-separated format for easy editing using spreadsheet software such as Microsoft Excel.

Table 4.19 Example Normalization Result File

```
“Normalization Factor Data File RUNID: 20010829121715”
    5,”Number of contaminants”,
“Realization”,”H3”,”Sr90”,”Tc99”,”I129”,”Cs137”
     1, 1.000E+00, 1.000E+00, 1.000E+00, 1.000E+00, 1.000E+00,
     2, 1.000E+00, 1.000E+00, 1.000E+00, 1.000E+00, 1.000E+00,
     3, 1.000E+00, 1.000E+00, 1.000E+00, 1.000E+00, 1.000E+00,
     4, 1.000E+00, 1.000E+00, 1.000E+00, 1.000E+00, 1.000E+00,
     5, 1.000E+00, 1.000E+00, 1.000E+00, 1.000E+00, 1.000E+00,
     6, 1.000E+00, 1.000E+00, 1.000E+00, 1.000E+00, 1.000E+00,
     7, 1.000E+00, 1.000E+00, 1.000E+00, 1.000E+00, 1.000E+00,
     8, 1.000E+00, 1.000E+00, 1.000E+00, 1.000E+00, 1.000E+00,
```

4.5.3 Temporary Active Run Status File

A temporary file is opened at the start of the analysis to indicate to the ESP that INVENTORY is running. When the INVENTORY analysis is complete (whether successful or not), the temporary file is deleted.

The file name is fixed at inv.run and is opened in subdirectory \inventory under the directory where the ESD keyword file resides.

4.6 Error Messages and Error Recovery Guidance

There are over 330 different error messages programmed into INVENTORY; most are to provide warnings of errors in input values. This section describes the error messages related to inconsistent data input and provides guidance on correcting the errors. Most error messages are self-explanatory and are not described in this section. Error messages are written to the run summary output file inventory.out.

4.6.1 Duplicate Waste Stream Names

Waste stream names are tested to be sure there are no duplicate names defined. The following is the error message indicating duplicate names:

```
Error: duplicate waste stream names: “duplicate waste stream name”
Stream numbers for duplicate streams are:  N1, N2
```

When the waste stream selection file is supplied (keyword SELECT WASTE in the inventory.key keyword file), names are read from that file. The entry numbers in that file given by N1 and N2 should be checked, and one should be modified so the names are unique. Note that when a name change is made to a waste stream, it is also necessary to change the WASTEMAP record for that waste stream (in the inventory.key keyword file).

If the waste stream selection file was not used, then the waste stream names are derived from the waste stream identified in the master waste stream disposal action file. In this case, the error message should not be encountered because the waste stream names master list is generated at the time of input by comparison to previously read names.

4.6.2 Duplicate Aggregation Site Names

The aggregation site names are tested for uniqueness. If duplicates are found, then the following error message is written.

```
Error: duplicate release site names: “duplicate name”
Site numbers for duplicate names are: N1 N2
```

The aggregation site names are read from the ESD keyword file. That file should be checked for duplication of names by searching for the “duplicate name” indicated in the error message.

4.6.3 Duplicate Analyte Names

The analyte names are provided in the ESD keyword file on ANALYTE keyword records. These names are tested for duplicates. If duplicates are found, then the following error message is written.

```
Error: duplicate contaminant names: ”duplicate name”
```

```
   Contaminant numbers for duplicate names are: N1 N2
```

The ESD file should be reviewed to be sure there are no duplicates of analyte names selected for the analysis (on ANALYTE keyword records that contain the COMPUTE modifier).

4.6.4 Duplicate Waste Type Names

The waste type names are provided in the inventory.key keyword file on keyword record SELECT TYPE. These names are tested for duplicates. If duplicates are found, then the following error message is written.

```
   Error: duplicate waste type names: ”duplicate name”
   Waste type numbers for duplicate names are: N1 N2
```

The inventory.key file should be reviewed to be sure there are no duplicates of waste types specified on the SELECT TYPE keyword record.

4.6.5 Aggregation Site (Release Site) Index Not Specified

All waste disposal actions (in the master waste stream disposal action file) must have a corresponding entry in the WASTEMAP keyword records in the inventory.key keyword file. If a corresponding entry is not found, then the following error message is written.

```
   Error in aggregation site index, not specified for'
   Waste stream: “waste stream name”  Waste Type: “waste type name”
   Year:    NNNN, set of three:    C
```

The value for C is one of three possible definition sets (resulting from the allowance for multiple definitions for fractional disposition of waste streams).

A similar error message will indicate the same problem, as follows.

```
   Error in release site index values, none specified for'
   Waste stream: “waste stream name”  Waste Type: “waste type name”
   Year:  NNNN
```

To resolve these errors, both files should be checked to determine discrepancies in definition of the disposal action for the waste stream, waste type, and year of disposal indicated in the error message.

4.6.6 Definition of Disposal Fraction Not Consistent

When a disposal action is split between two aggregation sites, the input records in the inventory.key file that define the two sites must have exactly the same disposal action representations. This is to avoid problems with accounting for distribution of waste. If the set of disposal actions differ between two WASTEMAP records that distribute fractional amounts of one waste disposal action to different aggregation sites, one of the following error messages is written.

```
Disposal fraction definition conflict: incompatible combination (C)'
Between streams “waste stream–waste type 1” and “waste stream–waste type 2”
```

or

```
Disposal fraction definition conflict: secondary combination (C)'
Between streams “waste stream–waste type 1” and “waste stream–waste type 2”
```

In these messages, C is an integer (1, 2, or 3) representing the fractional value index, and the two entries “waste stream–waste type N” are the conflicting entries from the WASTEMAP keyword.

To resolve the problem, entries in both the WASTEMAP keyword records and the DISPOSAL keyword entries from the master waste disposal action file should be reviewed for consistency. When the fractional disposal option is used on the WASTEMAP keyword records, errors can be minimized by specifying a single disposal action for each WASTEMAP record.

4.6.7 Decay Back Calculation Failure

When a decay reference date is specified and a disposal action is defined prior to the decay reference date, the code attempts to back calculate decay to estimate the quantity present at the earlier date. If the calculation fails because there is too much time between the two dates, the following error message is written.

```
Warning: Back calculation failed, too much decay for: “analyte name”
```

To correct the situation, check to be sure all dates specified are correct. The problem may be caused by including a radionuclide with a very short half-life. If this is the case, consider eliminating the radionuclide, or ignoring the earlier disposal actions for that radionuclide.

This problem should not occur when the DECAY NONE option is used for the analysis (see DECAY keyword description, Section 4.4.2).

4.6.8 Release Site Name Not in Master List

The primary source of release site names is the ESD keyword file, through keyword AGGREGATE ID=“name” (see Section 2.1.1). This list is compared against site release names read in the WASTEMAP AGGREGAT keyword from the inventory.key file. If a name in the inventory.key file is not in the ESD file, the following warning message is written:

```
Warning: reading WASTEMAP AGGREGATE     name “name”
Release site name not in master list.
This release site data set will be skipped
```

When this message occurs, check the “name” written against the ESD keyword file master list. This condition does not stop the analysis, but the indicated release site data set will not be included in the output files generated by the run.

4.6.9 Waste Stream Name Not Found in Master List

The primary source of waste stream names is the waste stream selection file (specified on keyword SELECT WASTE [see Section 4.4.10] in the inventory.key file). When the WASTEMAP AGGREGAT keyword records are read from the inventory.key file, the waste stream names are compared to the names in the waste stream selection file. If a name on the WASTEMAP AGGREGAT keyword line is not in the selection file, the following message is written.

```
In WASTEMAP record for STREAMS modifier
Waste stream name not found in master list.  Found “name”
This disposal action will be skipped.
```

When this message occurs, check the “name” written against the waste stream selection file list. If it is not in the selection list, find out why or add it to the list. Also check to see if the entry in the WASTEMAP AGGREGAT keyword is correct. This condition does not stop the analysis, but the disposal action data will not be present in the output files generated for the analysis.

4.7 Approach to Preparation of Input to INVENTORY

Input to INVENTORY can be complicated because it involves several sets of data that must be mutually consistent. This section provides guidance on assembling a complete set of input data for an INVENTORY analysis. The ESD keyword file is used as the starting point for assembling the data set because the ESD keyword file controls the overall scope of the analysis to be performed.

The ESD keyword file should be completed as a starting point for definition of the scope of the analysis. Several keywords provided in the ESD keyword file (see Section 2.1) are read by INVENTORY, which results in specific requirements in other input files. Table 4.20 describes the ESD keyword records read by INVENTORY, along with the corresponding requirements implied by the ESD keyword information.

Table 4.20 ESD Keyword Requirements for INVENTORY Code Input

ESD Keyword	Information Provided in ESD Keyword File	Requirements for Other INVENTORY Input Files
AGGREGAT	Aggregation site names, inclusion flag	Aggregation site names included in the inventory.key file must be defined in the ESD file to be included in the analysis. The inclusion flag (COMPUTE on the AGGREGATE record) must also be present. If the SELECT WASTE option is used in the inventory.key file, then an AGGREGATE record must be present for each waste site provided in the Waste Stream Selection file.
ANALYTE	Analytes to be included in the analysis	The analyte names (following modifier ID) define the analytes that may be included in the analysis. Data in the Master Waste Stream Disposal Action file for other analytes are ignored. A random seed must be provided for each analyte on the SEED keyword record in the

ESD Keyword	**Information Provided in ESD Keyword File**	**Requirements for Other INVENTORY Input Files**
		inventory.key file. If normalization is to be performed for an analyte, then a NORMALIZE record must be provided for the analyte. If the analyte is a radionuclide, then there must be an entry in the Radionuclide Master Decay Data Library.
BALANCE	Years for mass balance analysis	For mass balance analysis to be performed, a MASSBALANCE keyword record must be present in the inventory.key input file. If the MASSBALANCE record is present, then a FILE MASS keyword record must also be provided giving the file name for the mass balance analysis output.
PERIOD	Time period for the analysis	The AGGREGATE records in the INVENTORY keyword file should have all years of disposal within the range specified on the PERIOD record. If out of range, a warning message is written to the inventory.out file, but execution will continue without the out of range record. The Master Waste Stream Disposal Action file does not need the years of disposal within this range because any records for which the year is out of range are ignored.
REALIZATION	Number of realizations to be performed	The number of realizations must be between 1 and 99,999. There are no dependencies on the number of realizations.

The steps in generation of an input set for INVENTORY are summarized below. These steps assume that an ESD keyword file has already been generated.

1. Determine the waste disposal actions to be included by waste stream, waste type, and year of disposal. Generate DISPOSAL, TITLE, and USER keyword records for the Master Waste Stream Disposal Action file.

2. Prepare the TITLE and USER keywords for the inventory.key keyword file.

3. If a waste stream selection file is to be used, generate that file from the list of waste stream names resulting from the previous step. Enter a WASTE keyword in the inventory.key file giving the file name for the waste stream selection file.

4. Determine how each disposal action is to be aggregated into the list of aggregation sites. Generate the WASTEMAP keyword records for the inventory.key file.

5. If normalization is to be performed, determine the expected total amounts of analytes at the time of closure and prepare the NORMALIZE keyword records for the inventory.key file.

6. If a mass balance analysis is to be performed (based on the BALANCE keyword record in the ESD file) then include the MASSBALANCE keyword record in the inventory.key file. Also prepare the FILE MASS keyword record that provides the name of the output file for the mass balance results.

7. Prepare the SEED keyword for the inventory.key file, with entries for each analyte specified in the ESD file.

8. Prepare a SELECT TYPE keyword for the inventory.key file, with entries for each waste type to be included in the analysis.

9. Include a DECAY keyword record in the inventory.key file that indicates the year of decay reference for input values, or specify that no decay will be performed (DECAY NONE).

10. If an attempt to execute the analysis is to be performed, then include the EXECUTE keyword in the inventory.key keyword file.

11. Prepare the FILE WASTE record for the Master Waste Stream Disposal Action file in the inventory.key keyword file.

12. Include a FILE DECAY record providing the name of the Radionuclide Master Decay Data Library in the inventory.key keyword file.

13. If stochastic analysis results are to be saved, prepare the DEBUG keyword record and the FILE STOCHOUT keyword record providing the name of the stochastic results output file.

14. If information on analysis time (CPU time) for the run is to be saved, then include the TIMER keyword record in the inventory.key file.

After all input is prepared, run the analysis and review the inventory.out file for warning and error messages. A successful analysis will have the statement *INVENTORY Normal Termination* written to the screen upon completion of the analysis. Also, the file inv.done will be generated and placed in the \inventory subdirectory.

After all input file errors are corrected, the complete run can be performed. The resulting .RES files should be reviewed to be sure they contain the information expected. One file should be generated for each realization.

5.0 VADER – Vadose Zone Release Module

5.1 Purpose

VADER runs as a standalone program within the context of the SAC system (see Figure 1-2). Its purpose is to calculate quantities of waste material released from containment into the environment at the vadose source zone at regular (annual) time steps based on waste release models. A single VADER run performs these calculations for a given analyte at a given release site, using one or more built-in waste release models. The time series of annual releases forms the boundary conditions or source term for subsequent programs to calculate fate and transport of contaminants through the vadose zone to groundwater.

VADER is keyword-driven in that it reads a file of text commands that directs the processing. VADER reads inventory and remediation data files and writes several output files. The output files support mass-balance calculations, provide source term input to STOMP, support remediation transfers, and provide SAC ESP information for managing processing in the SAC-GW/VZ environment.

A single VADER run generates one set of deterministic waste releases for a given analyte on a given site inventory, using deterministic release model coefficients. Monte Carlo simulations are performed by running VADER many times, each time with a different realization of model coefficients and inventory. This ensemble of time profiles would be examined as a whole to assess the variability in release profiles.

5.2 Concepts, Algorithms, and Assumptions

VADER currently incorporates five solid release models as specified by Appendix D of PNNL-11800 (Kincaid et al. 1998) and two liquid pass-through release models. The solids models are for the waste forms designated as soil-debris, cake, glass, cement, and reactor core waste forms. They are distinguished chiefly by permeability to percolating waters and by chemical release mechanisms. The liquid release model passes liquid releases from the inventory instantaneously into the soil at the top of the vadose zone with no retardation or decay (input into STOMP). The river release model passes liquid releases from inventory directly into the river (body of water) also with no retardation or decay (input to GWDROP).

During any given run, VADER uses the input annual time series plus net remediation as a source term to calculate the quantity (mass or activity) released through simulated release processes accounting for simple radioactive decay. VADER writes annual release quantities to output files in specified formats (STOMP template or GWDROP input file) during each simulation year and also maintains an output file of quantities released and remaining at the site (vader.table file) for mass balance.

5.2.1 Limitations

This version of VADER accounts for simple decay for radioactive contaminants. Moreover, VADER does not attempt to simulate chemical reactions that change the released contaminants into different chemical forms. VADER considers each analyte as being chemically and radiologically independent from all other analytes.

VADER does not generate stochastic realizations of release model coefficients, but rather relies on receiving stochastic realizations of coefficient values generated offline and delivered to VADER through the input keyword file.

VADER calculates released quantities in the release zone (vadose zone or river) accounting for decay and without attempting to account for transport away from the release zone. Once released, transport is the function of the STOMP, CFEST, and MASS2 modules (see Figure 1-2).

5.2.2 Waste Forms in VADER

The waste forms handled by VADER are described in BHI (1999), Appendix D of Kincaid et al. (1998), and Riley and Lo Presti (2001). In the inventory file (see Section 5.5.1.2) quantities of a given substance or analyte are assigned to one or more of several waste forms such as soil-debris, salt cake, cement, glass, reactor core, liquid, or river. Each waste form at a given site has assigned to it unique characteristics such as density, solubility, or moisture content which are specific to or characteristic of that particular analyte at that site. Each site may have one or more waste forms. Each waste form has associated with it a unique release model such as SOIL, CAKE, CMNT, GLASS, REACTOR, LIQUID, or RIVER. For each waste release model, specific characteristics of the waste form are expressed as model coefficients.

An inventory of wastes in terms of quantity (mass or activity depending on the chemical entity) and volume is supplied to VADER in the form of an input INVENTORY file. The INVENTORY file contains annual quantities generated at the site, decayed to the year of generation.

The release model associated with a given waste form depends on what the waste form is called in the INVENTORY file. For example, some sites have carbon tetrachloride in liquid form. If this waste form is called *LIQUID* in the INVENTORY file, it will be released by the LIQUID release model, which is an instantaneous pass-through process. But it may also be released by a different model by assigning it to a different waste form. If the carbon tetrachloride is assigned to the SOIL waste form in the INVENTORY file, it will be released using the SOIL release model.

5.2.3 Release Models

The mathematical derivations for the release models are fully documented in Appendix D of Kincaid et al. (1998) and CCN 0512242 (BHI 1999). The following remarks are intended only to highlight model assumptions and implementation features. Table 5.1 identifies the waste forms processed by VADER.

Table 5.1 Waste Forms Processed by the VADER Code

Waste Form	Characteristics	Source	Release Mechanism	Equations[(a)]
Soil-Debris	High permeability	Soils, cribs, trenches, etc.	Adsorption & Solubility	D.28 - D.44
Glass	Very low permeability/ Immobile	Vitrification	Dissolution of glass	D.52 – D.60

Waste Form	Characteristics	Source	Release Mechanism	Equations[(a)]
Salt Cake/ Sludge	High permeability	Tank waste	Dissolution congruent to salt-cake waste form	D.45 - D.51
Concrete/ cement	Low/medium permeability	Misc.	Diffusion	D.61 - D.63
Reactor Blocks	Other	Irradiated solids	Corrosion	D.64 - D.67
Liquid	Instantaneous	Various	Pass-through	None
River	Instantaneous	Reactor coolant	Pass-through	None

(a) The equations are provided in Appendix D of Kincaid et al. (1998)

Table 5.1 summarizes some of the physical characteristics of the solids release models. All the solids models (for SOIL, CAKE, GLASS, REACTOR/CORE, CEMENT waste forms) invoke a release mechanism that regulates or retards the release of contaminant based on simplified conceptions of how the analyte is leached from, or diffuses out of, the inventory source. Mathematically, these models are simple ordinary differential equations (single-compartment models) without feedback. Some of the models represent inventory depletion with release based on instantaneous current net accumulated inventory. Other models represent depletion with release based on a constant initial inventory.

The release models are expressed in terms of differential equations. In these equations, dM/dt indicates the annual release, M_o indicates the mass of the original inventory, and M_i indicates the mass of the inventory remaining at step i. VADER solves these differential equations numerically for each time step over the release period by calculating dM_i/dt after accounting for all quantities added or subtracted in evaluating the current net accumulated inventory M_i. Details for each release model are provided below. For all models except instantaneous release models, a fourth-order Runge-Kutta solution for dM/dt is implemented annually. The annual solutions allow for time-dependent coefficients, such as the recharge rate Q_w.

The release models are of two types: depleting inventory and nondepleting inventory. In depleting inventory models, the annual releases are based on the current inventory (the SOIL model). In nondepleting inventory models, the releases are based on initial inventories (the CMNT, CAKE, and REACTOR models).

The equations are coded to accommodate annual changes in the source term from which to calculate annual releases. Up until the time of site closure (see Section 2.1.18), the inventory is presented in terms of quantities added to site inventory annually as a result of site operations. In addition, material can be added to, or subtracted from, the inventory each year as a result of remediation transfers. In models where release quantity is based on original quantities (for example, the REACTOR and SALTCAKE models), two quantities are carried forward: (1) net accumulated inventory without accounting for release for calculating release and (2) net accumulated inventory after subtracting release from (1) to calculate inventory remaining.

Liquid releases (for LIQUID and RIVER waste forms) are conceived as instantaneous pass-through models, in which the liquid inventory is immediately passed through the site and on into the vadose zone (LIQUID wastes) or into the river (RIVER wastes). This is expressed as a release fraction of current inventory. For complete pass-through, the release fraction is set to 1.

A useful rule of thumb for numerical solution of differential equations is that the time step is no larger than one-half the smallest time constant in the system being simulated. An example of a system time constant is the half-life of a radioactive analyte. Applying this rule avoids numerical instabilities in the time series results arising from short-lived transients. For SAC Rev. 0 radioisotopes, all the expected radionuclides are long-lived, from a decade to millions of years. However, the coefficients for the release models may take values over several orders of magnitude, based on Table D.2 in Appendix D of Kincaid et al. (1998). In summary, a time step of one year is adequate for analytes having a half-life of a couple of years or longer, when the other model coefficients also result in longer system time constants.

5.2.3.1 Soil Model Release Algorithms

The soil wastes are considered to be unconsolidated porous material with the contaminants distributed uniformly throughout the waste. The following modeling assumptions are made:

- Chemical interaction between contaminants is treated as negligible.
- There is no organic liquid phase immiscible with the aqueous phase present.
- There is no competition between contaminants for sorption sites.
- The only partitions present are aqueous, sorbed, and precipitated.
- No vapor phase is modeled.

The soil release model has two states (dissolution-controlled and desorption-controlled) and switches between model states depending on local conditions. The release quantity at each step is based on current net accumulated inventory quantity remaining after accounting for remediation transfers. The soil release model is applied to soil-debris wastes in cribs, French drains, unconsolidated debris, and trenches. The groundwater recharge rate, Q_w, is time-dependent and the value is updated each year.

The soil release model is based on Equation D.35 of Kincaid et al. (1998). Releases are based on the depleting inventory concept when under desorption control, and on the nondepleting concept when under dissolution control. First, define the unitless phase apportionment factor, R, as follows:

$$R = 1 + \frac{\beta\, K_d}{\theta_w}$$

where:

β = the bulk density of the source zone soil or vadose zone soil (g/cm^3)
K_d = linear equilibrium sorption coefficient for the contaminant (cm^3/g)
θ_w = the volumetric water content of the source zone soil or vadose zone soil (unitless).

Then, define the cutoff between the dissolution and desorption control as

$$M_{max} = R\,\theta_w\;C_{sol}\,A h$$

where:

C_{sol} = the solubility limited concentration for an analyte in water (Ci/cm^3 or kg/cm^3)
A = the cross-sectional area of the waste (cm^2)
h = the height of the waste (cm).

Then, the change in waste mass for a given time step i due to the release mechanism is found from:

$$\frac{dM_i}{dt} = Q_w\,A\,C_w$$

where:

Q_w = the Darcy flux density of water flowing through the source region (cm/yr)
C_w = the waste concentration in the water flowing through the waste form (Ci/cm^3 or kg/cm^3).

The value for C_w depends on whether the release is under dissolution or desorption control. It is set equal to C_{sol} when the waste mass at time step i, denoted by M_i, is greater than M_{max}, otherwise, C_w is found from the following equation in which M_i has been decayed (for radioactive contaminants):

$$C_w = \frac{M_i}{R\,\theta_w\,A\,h}$$

The time-stepping solution for the mass at time step i+1, denoted by M_{i+1}, is found from the equations:

$$\text{Release}_i = \frac{dM_i}{dt} e^{-\lambda t}$$

$$M_{i+1} = \left(M_i - \frac{dM_i}{dt} \right) e^{-\lambda t}$$

in which:

λ is the decay constant for radioactive analytes (1/yr)
t is the length of a time step (typically one year)
$e^{-\lambda t}$ is set to 1 for nonradioactive analytes

In the current implementation of the SOIL model, all the site coefficients remain constant over time except for Q_w. Thus, the release model uses constant site dimensions but allows addition or subtraction of material from the site through remedial actions. The constant site dimensions are maintained by the assumption that any contaminated soil removed is instantaneously replaced with an equal amount of uncontaminated soil.

5.2.3.2 Cake Model Release Algorithms

The cake release model mostly applies to tank wastes, including the residue or hard heel left in the tanks after the wastes have been removed for vitrification. These salt-cake wastes are considered to be a structural matrix of consolidated porous material, with contaminants distributed uniformly throughout the matrix. The distribution of contaminants in the salt cake is assumed to remain constant over time. Currently, the physical mix of salt cake, sludge, and heel is treated as aggregated into a single structural matrix.

The conceptual formulation for this release model considers percolating groundwater from recharge entering the containment region and leaching the salt cake. As the salt cake dissolves, the embedded contaminants are leached congruently with the dissolving salt cake. Annual release quantities are calculated based on net accumulated annual salt-cake inventory; thus the salt-cake model is a nondepleting release model. The model requires the groundwater recharge, Q_w, as a function of time to modulate the quantity of salt cake leached away. The initial contaminant inventory is depleted at a constant rate during periods of constant groundwater recharge rate.

The input inventory data file (Section 5.5.1.2) does not provide salt-cake mass directly. Instead, it provides waste volume, which is taken to be the volume of salt cake added to the site inventory annually. VADER converts the net accumulated volume to mass units using the salt-cake density that is provided as a model parameter. Although tank waste densities may vary widely within each tank, this implementation of the CAKE model uses a single density to represent all the salt-cake wastes at each site. If the model specifications do not set a density, a default value of 1.5 g/cm^3 is used.

The cake model is derived from Equation D.49 of Kincaid et al. (1998) and is given by

$$\frac{dM_i}{dt} = M_0 \left[\frac{Q_w \, A \, C_{sol}}{M_{msco}} \right]$$

where:

M_0 = the original mass or activity of contaminant (Ci or kg)
Q_w = the Darcy flux density of water flowing through the source region (cm/yr)
A = the cross-sectional area of the waste subject to groundwater recharge (cm^2)
C_{sol} = the solubility limited concentration for an analyte in water (Ci/cm^3 or g/cm^3)
M_{msco} = the original salt-cake mass {C_{sol} and M_{msco} have the same mass units} (Ci/cm^3 or g/cm^3).

The time-stepping solution for the mass at time step i+1, denoted by M_{i+1}, is found from the following two equations:

$$\text{Re lease}_i = \frac{dM_i}{dt} e^{-\lambda t}$$

$$M_{i+1} = \left(M_i - \frac{dM_i}{dt} \right) e^{-\lambda t}$$

in which:

λ is the decay constant for radioactive analytes (1/yr)
t is the length of a time step (typically one year)

$e^{-\lambda t}$ is set to 1 for nonradioactive analytes

In VADER, the algorithm is implemented by carrying forward the net accumulated contaminant mass without accounting for release. In effect, the release quantity is based on an adjusted M_0 that accounts for the mass changes when waste is imported to the site or exported to other sites.

In the current implementation, A (cross-sectional area) and C_{sol} (solubility limited concentration) remain constant over time. The numerical algorithm checks the analyte quantity remaining at each time step and stops the release when all the analyte has leached away.

5.2.3.3 Cement Model Release Algorithms

The cement model is applied to cement structures and grouted wastes. Contaminants are assumed to be uniformly distributed throughout the solidified cement waste. The release mechanism is diffusion through the solid material to the outer surface of the waste form, from whence it is carried away by water percolating past the surface.

Releases are based on the initial inventory, M_0 (non-depleting release model), and on the square root of elapsed time from start of release mechanism or the date of entry into the inventory, whichever is closer to the current year. Because of the square root of elapsed time, a special stepping algorithm is supplied for the CEMENT model; each individual inventory entry of cement has to be tracked separately rather than pooled as is done with other non-depleting models.

The cement model is given by Equation D.61 of Kincaid et al. (1998) and has the following form:

$$\frac{dM}{dt} = M_0 \left(\frac{A}{V} \right) \sqrt{\frac{D_c}{\pi \, \Delta t}}$$

where:

M_0 is the original amount of contaminant in the cement (Ci or kg)
A is the cross-sectional area of the waste exposed to the release mechanism (cm^2)
V is the volume of the waste (cm^3)
D_c the effective diffusion coefficient of the contaminant within a cement waste form (cm^2/yr)
π is the mathematical constant
Δt is the elapsed time from effective start of the release mechanism ($\Delta t = t_i - t_0$).

The time-stepping solution for the mass at time step i+1, denoted by M_{i+1}, is found from the following two equations:

$$\mathrm{Re\,lease}_i = \frac{dM_i}{dt} e^{-\lambda t}$$

$$M_{i+1} = \left(M_i - \frac{dM_i}{dt} \right) e^{-\lambda t}$$

in which:

λ is the decay constant for radioactive analytes (1/yr)
t is the length of a time step (typically one year)
$e^{-\lambda t}$ is set to 1 for nonradioactive analytes

This is performed for each unique cement inventory addition, and to get the final cement release, all the release profiles and quantity remaining profiles are summed together. The numerical algorithm checks the analyte quantity remaining at each time step and stops the release when all the analyte has leached away.

5.2.3.4 Reactor (Core) Model Release Algorithms

The reactor core model is applied to graphite reactor core blocks. The reactor core release rate is dependent on the original inventory M_0, so the cumulative inventory without accounting for releases is used to calculate the release each year. The reactor core model from Equation D.65 in Kincaid et al. (1998) is the following:

$$\frac{dM_i}{dt} = M_0\ F_r$$

where M_0 is initial net contaminant quantity (Ci or kg), and Fr is an empirical release rate (1/yr) which does not reflect any particular release mechanism. The inventory at any given time step i is given by the equations:

$$\text{Re lease}_i = \frac{dM_i}{dt} e^{-\lambda \Delta t}$$

$$M_i = \left[M_0 - \Delta t \frac{dM}{dt} \right] e^{-\lambda \Delta t}$$

where:

λ is the decay constant for radioactive analytes (1/yr)
$e^{-\lambda t}$ is set to 1 for nonradioactive analytes
Δt is the elapsed time from the year the reactor core was first exposed to leaching water.

The numerical algorithm checks the analyte quantity remaining at each time step and stops the release when all the analyte has leached away.

5.2.3.5 Liquid Model Release Algorithms

Liquid releases are modeled as being dumped to the ground and instantly available to go into the vadose zone without retardation the year they enter the inventory. This is a pass-through, no decay, instantaneous release and complete depletion release model. The liquid release model is given in the following equations:

$$\mathrm{Release}_i = \frac{dM_i}{dt} = M_i\ F$$

$$M_{i+1} = M_i - \frac{dM_i}{dt}$$

where M_{i+1} denotes the inventory remaining at time step i+1 (Ci or kg) and F denotes a fractional release rate (1/yr). When the fractional release rate F is assigned the value 1, the entire inventory is instantaneously released into the ground.

VADER provides for specifying the fractional release rate F between 0 and 1, so that the dumping of wastes can be retarded. In this case, the liquid release model is considered a depleting-inventory model, in that the release quantity is based on quantity remaining. The inventory remaining at any given time step i is given by the equation:

$$M_{i+1} = \left(M_i - \frac{dM_i}{dt} \right) e^{-\lambda t}$$

in which:

λ is the decay constant for radioactive analytes (1/yr)
t is the length of a time step (typically one year)
$e^{-\lambda t}$ is set to 1 for nonradioactive analytes

The numerical algorithm checks the analyte quantity remaining at each time step and stops the release when all the analyte has moved into the ground.

5.2.3.6 River Model Release Algorithms

The river-release model is a pass-through, no decay, instantaneous and complete depletion release model. River releases are treated as instantly available to go into the river without retardation the year they enter the inventory. This model simulates the practice of some of the early Hanford reactors. The river releases are written to file vader.rem, which is later read by the GWDROP model.

The river release model is given in the following equations:

$$\mathrm{Release}_i = \frac{dM_i}{dt} = M_i\ F$$

$$M_{i+1} = M_i - \frac{dM_i}{dt}$$

where M_{i+1} denotes the inventory remaining at time step i+1 (Ci or kg) and F is a fractional release rate (1/yr). When the fractional release rate F is assigned the value 1, the entire inventory is instantaneously released into the river.

VADER provides for specifying fractional release rate F between 0 and 1, so that the release of wastes can be retarded. In this case, the liquid release model is considered a depleting-inventory model, in that release quantity is based on quantity remaining. Quantities released to the river bypass STOMP (and never enter groundwater) and are read by the GWDROP module. The decayed inventory at any given time step i is calculated using the equation:

$$M_{i+1} = \left(M_i - \frac{dM_i}{dt} \right) e^{-\lambda t}$$

in which:

λ is the decay constant for radioactive analytes (1/yr)
t is the length of a time step (typically one year)
$e^{-\lambda t}$ is set to 1 for nonradioactive analytes

The numerical algorithm checks the analyte quantity remaining at each time step and stops the release when all the analyte has moved into the river.

5.2.4 Waste Site Remediation

VADER allows wastes to be transferred to or from a working site as a result of environmental remediation efforts. From the viewpoint of a VADER run, remedial actions may be of two forms:

- **Transfers of soil wastes from the ground or upper vadose zone of previously computed sites to the working site inventory.** These transfers are generated by STOMP in the form of REMEDIATE files written to the working site directories. It is assumed that all STOMP-generated wastes are SOIL wastes to be placed in the working site inventory as of a given year.
- **Transfers of wastes from previously computed site inventories to the working site inventory.** These wastes may be SOIL, CORE, or any other canonical waste form. These transfers are generated by VADER in the form of vader.rem files written to the previously computed site directories. Wastes in the working site inventory may also be transferred to other yet to be computed site by the same mechanism.

Other assumptions are that remedial actions can occur only once per year and all material is transferred instantaneously on the first day of the year. Radioactive wastes transferred on a given year are assumed to have been decayed to that year.

A list of remediation actions to perform at a working site is defined using REMEDIAT keywords (see Section 5.4.14). The information required is the calendar year of the action, import and export sites, and percentage of net cumulative contaminant (analyte) to transfer, so as to allow calculation of mass or activity to be subtracted from the release and remaining inventory profiles and added to the quantities to be transferred.

Remedial transfers of material can take place in more than one step, but a circular flow of the same waste is not allowed. The flow of actions can be such that some material from site A can be moved to site B, and then later the material can be moved from site B to site C. However, material moving from site B cannot later go back to site A, and material from a site cannot be transferred to itself. Moreover, material from site A cannot go to site B, then to site C, then back to site A. VADER runs for a network of sites must be conducted in an order that reflects the site remediation plan.

5.3 Code Environment

VADER is written in Fortran 95. The code has been run in the Windows operating system (Releases 98, NT, and 2000) and also in a Linux (Red Hat 7.1) environment.

5.3.1 Location in the Processing Sequence

The general processing sequence for SAC environmental runs is outlined in Figure 1-2. The inventory data for input to VADER is generated by the INVENTORY program. The vader.key file is generated by SAC ESP, which also generates realizations of release model parameters and coefficients. Remedial action files for use in VADER are generated by previous VADER and STOMP runs.

In addition to providing input to itself through remedial actions, VADER provides source term inputs to two programs: STOMP and GWDROP. STOMP uses the vadose zone source term in three parts: solid waste annual releases, liquid waste annual releases, and liquid waste volume annually. This data transfer is accomplished by overwriting a STOMP template file that the SAC ESP provided as input to VADER. Thus, the VADER and STOMP codes run in tandem. GWDROP uses the quantity of contaminants released directly into the river, thereby bypassing the vadose zone and groundwater (placed into the vader.rem file). A detailed context diagram for VADER is provided in Figure 5-1.

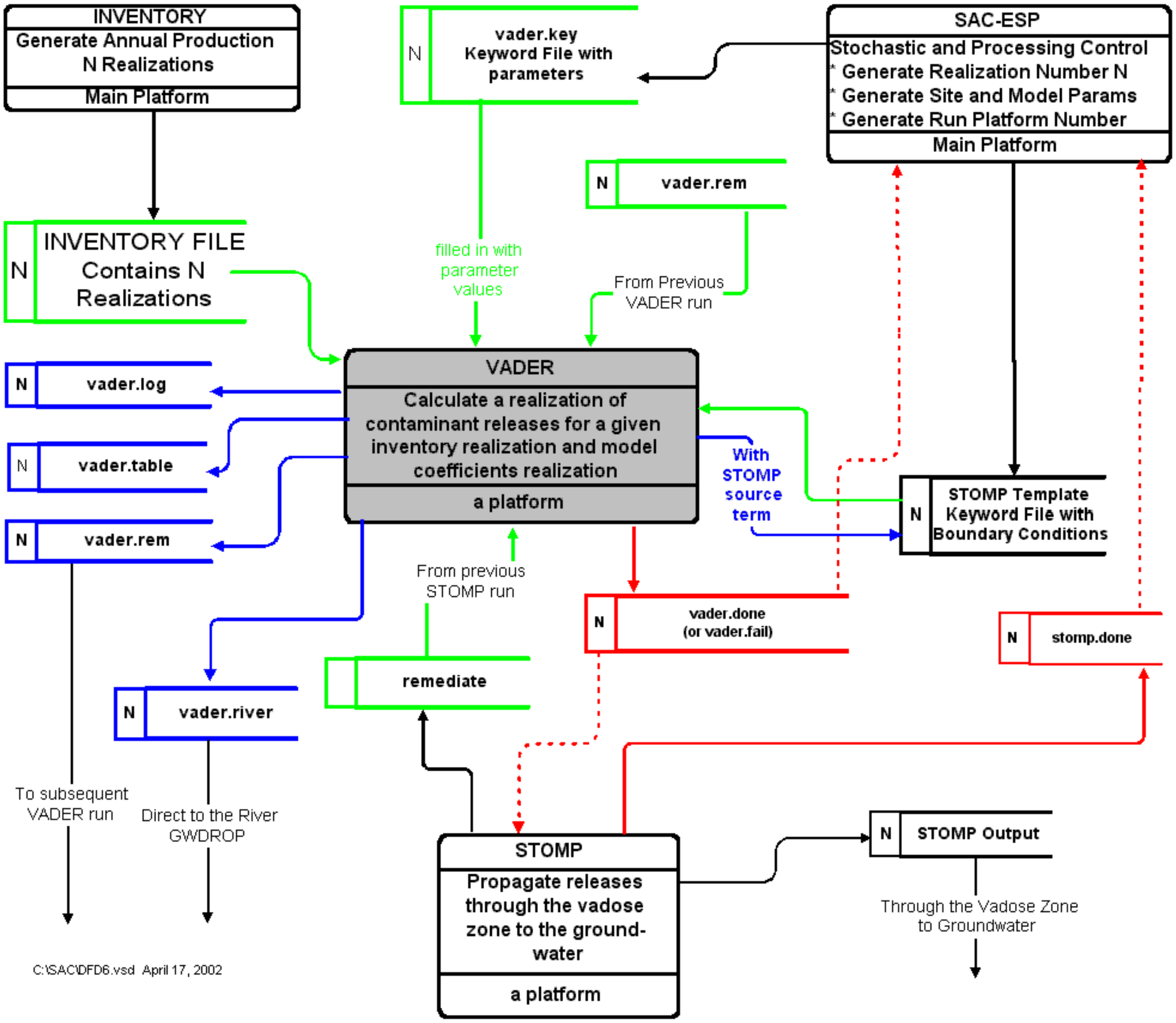


Figure 5-1 VADER Data Flow Diagram

5.3.2 How the VADER Code Is Invoked

VADER is a standalone program with no user interaction beyond preparing the input files. Under a Windows operating system, VADER is executed by entering the following command in a DOS window:

```
VADER path
```

in which *path* is normally the path name to the working directory. The working directory is expected to contain the vader.key file, the inventory file, and the STOMP template file, and will contain all of the output files written by VADER. Under the Linux operating system, VADER is executed through the following Bourne Shell or C Shell command:

```
vader.exe path
```

in which *path* is normally the path name to the working directory. The following rules apply for both the Windows and Linux operating systems:

- If the final character of *path* is the / or \ symbol, treat *path* as a path name, and store the \ or / for later path building. This use assumes that all processing will take place under the same operating system.
- If the final character of *path* is not \ or /, scan for \ or / in the text of *path*. If the \ or / symbol is embedded in the text of *path*, append the same character on the end of *path* and use the text string as a path name. If the \ or / symbol is not embedded in the text of *path*, treat the text as a file name to use instead of the default *vader.key* and look for it in the current working directory.

If application of the above two rules does not identify a VADER keyword file to be read, VADER writes the vader.fail file and terminates execution. If the command line argument is missing, VADER searches for a file named vader.key in the working directory in which VADER was invoked. If vader.key is absent, VADER writes a vader.fail file and terminates execution.

The following is an example command line entry:

```
VADER C:\SAC\Run\VADER\Site_200E\Analyt_H3\01\
```

This command tells VADER to search for the vader.key keyword file in the working directory C:\SAC\Run\VADER\Site_200E\Analyt_H3\01.

5.3.3 Run Times

VADER running in batch mode on a 450-MHz Pentium III PC under Windows 98 can run a typical seven-model release scenario over 1,100 years in under 2 CPU seconds, with no other application program running. For a 11,100-year run, VADER runs the same scenario in 25-30 CPU seconds. Accessing numerous remediation files increases the run time slightly.

5.4 VADER Keyword Descriptions

This section lists VADER keywords and command structures. It is expected that in the operational system, the SAC ESP module will generate keyword files specific to realization, site, and analyte for input to VADER and will also provide information relative to remediation actions.

In the following keyword descriptions, some data are optional and some are required. Data that are required are enclosed in square brackets. For AB to be required, it would be denoted by [AB]. If only one of the three items AB, BC, CD were required, it would be written as [AB|BC|CD]. The vertical bars indicate that the user must select one of the items in the list. Optional items are enclosed in normal brackets. For DE to be an optional entry, it would be denoted by {DE}. These symbols do not need to be entered when the keyword is constructed. The keyword name can contain any number of characters; however, only the first eight characters are used (for example, REALIZAT = REALIZATION = REALIZATxxx). Section 10.0 further describes the general syntax for keywords. Table 5.2 provides a summary of keywords used in VADER.

Table 5.2 Summary of Keywords Used by VADER

Keyword	**Description**
AGGREGAT	The AGGREGAT keyword is used to specify the working aggregate release site to process.
ANALYTE	The ANALYTE keyword specifies the single analyte to be used in the simulation.
CUTOFF	The optional CUTOFF keyword supplies lower cutoff values for reporting annual quantity and annual volume releases for the vadose zone.
DEBUG	The optional DEBUG keyword is used to activate additional output options that are useful for code debugging purposes.
DELTAT	The optional DELTAT keyword supplies the time step for numerical solution of the release model differential equations.
END	The END keyword signifies the end of all keyword data.
FILE	The optional FILE keyword can be used to enter file names that are different from the default file names.
MAP	The optional MAP keyword provides a means for explicitly assigning a release model to an arbitrary waste form present in the INVENTORY.
MODELS	The MODELS keyword specifies which release model simulations to run and supplies model release coefficients.
PATH	The optional PATH keyword overrides the command-line argument path name.
PERIOD	The PERIOD keyword identifies the start and stop years for the release simulation.
REALIZATION	The REALIZATION keyword identifies the single realization to be simulated.
RECHARGE	This optional keyword allows entry of a time-dependent water recharge value at a given site as a lookup table.
REMEDIATE	The optional REMEDIAT indicates the input files that define incoming remediation quantities and provides information for calculating outgoing remediation quantities.
TITLE	The TITLE keyword defines a problem title to be written to output files.

The keywords can generally be entered in any order. However, the END keyword signifies the end of the keywords and must be the last keyword in the file. No BALANCE keyword is used in VADER because a complete mass-balance is always generated and written to the output vader.table file.

5.4.1 AGGREGAT Keyword for VADER

The AGGREGAT keyword is used to specify the working aggregate release site to process. This keyword is mandatory. One and only one working release site is defined for each run; if more than one AGGREGAT keyword is entered, the last one entered will be used. The site specification value is used for identification and to look up site-specific characteristics (constants) not specified explicitly in the keyword file. The following is the general syntax for this keyword:

```
   AGGREGATE [ID= "quote 1"] {TITLE="quote 2"} {NOCOMPUTE}
```

Table 5.3 defines the modifiers associated with the AGGREGAT keyword. These modifiers, together with their associated quote strings or numerical data, can be entered in any order.

Table 5.3 Modifiers and Data Associated with the AGGREGAT Keyword

NOCOMPUTE	Releases for this aggregation site will be calculated unless the NOCOMPUTE modifier is present. The NOCOMPUTE modifier provides a method for error-checking inputs without performing all of the calculations. This modifier must not be present during a production run of the code because no releases will be calculated.
ID	The site identification string is entered in quotes using the ID modifier. This string is limited to a maximum of 15 characters and must be unique. It is used to associate other data with a specific site.
TITLE	A descriptive title for the aggregation site is entered in quotes using the TITLE modifier for labeling contents of output files. The title line is limited to 72 characters.

The following is an example keyword command to specify an aggregation release site:

```
   AGGREGATE  ID= "WIPP" TITLE="Waste Isolation Pilot Plant"
```

5.4.2 ANALYTE Keyword for VADER

The ANALYTE keyword specifies the analyte to be used in the simulation. One and only one analyte is defined for each run. If more than one ANALYTE keyword is entered, the last entered will be used. The following is the general syntax for this keyword.

```
   ANALYTE [ID="Quote1"] [TYPE="Quote2"] {HALFLIFE=N1} {NOCOMPUTE}
```

Table 5.4 defines modifiers associated with the ANALYTE keyword. These modifiers, together with their associated quote strings or numerical data, can be entered in any order. Data associated with the modifiers should match exactly with the data for the same modifiers for the ANALYTE keyword in the ESD keyword file (see Section 2.1.2).

Table 5.4 Modifiers and Data Associated with the ANALYTE Keyword in VADER

NOCOMPUTE	Releases for this analyte will be calculated in the absence of the NOCOMPUTE modifier. The NOCOMPUTE modifier provides the capability to error check inputs without performing calculations.
HALFLIFE	The numerical entry associated with the HALFLIFE modifier is the half-life of a radioactive isotope in years. This modifier should not be entered for nonradioactive isotopes.
ID	The quote string associated with the ID modifier is an analyte identification string up to six characters in length.

TYPE	The quote string associated with the TYPE modifier is a two-character string to specify one of four analyte types: • NR - Inorganic radioactive isotope or compound containing a radioactive isotope • NS - Nonradioactive (stable) isotope or inorganic compound • OR - Organic radioactive isotope or compound containing a radioactive isotope • OS - Organic compound containing only nonradioactive (stable) isotopes

As an example, the following ANALYTE keywords would select tritium (H3) and neptunium-237 for analysis (in different runs).

```
ANALYTE "H3" NOCOMPUTE
ANALYTE ID="Np237" HALFLIFE= 2.14e06   TYPE="NR"
```

The ANALYTE keyword for tritium would invoke only input error-checking options and no calculations would be performed. The Np237 keyword would cause calculations to be performed.

5.4.3 CUTOFF Keyword for VADER

The optional CUTOFF keyword supplies lower cutoff values for reporting annual quantity and annual volume releases for the vadose zone. This keyword applies to writing data for the STOMP INPUT file. The purpose of this option is to allow VADER to write to INPUT only data over the time period when the values are consistently greater than the cutoff. By not having to process values lower than the cutoff, STOMP is able to move into adaptive step-sizes and perform more efficient computations based on release quantities and volumes of significant health or environmental impact. VADER continues to write all results over the simulation time period to vader.table. The following is the general syntax for this keyword.

```
CUTOFF ANALYTE=Na  VOLUME=Nv
```

in which Na is the smallest quantity of analyte released (for both the AGGREGATE solids release and LIQUIDS release) to write to the STOMP INPUT file and Nv is the smallest volume of liquids release water to write to the STOMP input template. Cutoff value units are activity or mass units (Curies per year or kilograms per year), and m^3 per year or liters per year for volumes, as specified in the INVENTORY file. Both Na and Nv default to 0 unless entered using this keyword.

The following is an example where the analyte cutoff is set to 1 picoCurie per year:

```
CUTOFF  ANALYTE=1.0E-12
```

For this example, the volume cutoff defaults to 0. Suppose the release profile starts at 10 Curies in 1990, and then decreases with time, so that by 2020 it drops below 1 picoCurie and never again rises above 1 picoCurie. VADER will write to the STOMP input file all releases over the period 1990 through 2020, stopping at 2021.

Annual release quantities are written to the STOMP INPUT file in time units of seconds. This means that if CUTOFF is set to 10^{-12} Ci per year, the INPUT file will show quantities down to approximately 10^{-20} Ci per second. The same time scale considerations apply to volumes. To provide an idea of the change in scale, there are 3.1536 x 10^{7} seconds per non-leap year.

5.4.4 DEBUG Keyword for VADER

The optional DEBUG keyword is used to activate additional output options that are useful for code debugging purposes. This keyword has 10 optional modifiers, and multiple DEBUG keywords can be entered. Outputs activated by this keyword are written to the vader.debug file. Table 5.5 shows the portions of VADER to be debugged. The following is the syntax for this keyword record:

```
DEBUG {CAKE} {CEMENT} {CORE} {GLASS} {INVENTORY} {LIQUID} {LOADING}
   {REACTOR} {REMEDIATE} {SOIL}
```

Table 5.5 Modifiers Associated with the DEBUG Keyword in VADER

Modifier	**Portion of Code Affected**
CAKE	Salt cake/sludge release model routines.
CEMENT	Cement release model routines.
CORE	Reactor release model routines.
GLASS	Glass release model routines.
INVENTORY	Inventory processing.
LIQUID	Liquid release model routines.
LOADING	Loading the inventory and remediation data into arrays.
REACTOR	Reactor release model routines.
REMEDIATE	Remediation processing routines (input and outgoing).
SOIL	Soil release model routines.

An example use of the DEBUG keyword to activate the CAKE, CEMENT, and SOIL output options is the following:

```
DEBUG CAKE
DEBUG CEMENT SOIL
```

5.4.5 DELTAT Keyword for VADER

The optional DELTAT keyword supplies the time step for numerical solution of the release model differential equations. The default setting if the DELTAT keyword is omitted is 1 for one-year time-steps. The following is the general syntax for the keyword:

```
   DELTAT N
```

The default setting can also be achieved by entering the following keyword:

```
   DELTAT 1.0
```

A time step of one year implies that a release model system would need to have a time constant (minimum decay constant) of at least two years for its time-dependent behavior to be adequately captured. The numerical calculations use the 4th order Runge-Kutta algorithm for all release models for solid wastes.

5.4.6 END Keyword for VADER

The END keyword signifies the end of all keyword data. It should be the last keyword in the keyword file. Any data in the keyword file after the END keyword will be ignored. The syntax for this keyword record is the following:

```
   END {"message"}
```

If a message with at least one character is entered, it is written to the log file upon run completion.

5.4.7 FILE Keyword for VADER

The optional FILE keyword can be used to enter file names that are different from the default file names. Each file definition requires a separate FILE keyword with a type designator and a NAME modifier. If multiple keywords for the same file type appear in the keyword file, the last one encountered is the one used.

The following is the syntax for the FILE keyword record:

```
   FILE  [INVENTORY|SAC_ESD|REMEDIATION|LOG_FILE|OUTPUT_TABLE|
      STOMP_TEMPLATE|DONE] [NAME="quote1"] {TYPE=ASCII|BINARY}
      {OPEN=CREATE|OVERWRITE|READONLY} {NOCOMPUTE}
```

Table 5.6 describes the modifiers for the FILE keyword.

Table 5.6 Modifiers Associated with the FILE Keyword in VADER

Modifier	Description
INVENTORY	The INVENTORY modifier identifies an input inventory file generated by the inventory module.
SAC_ESD	The SAC_ESD modifier identifies the input environmental settings definition keyword file.
REMEDIATION	The REMEDIATION modifier identifies an incoming remediation file from STOMP.

Modifier	**Description**
LOG_FILE	The LOG_FILE modifier identifies the VADER output log file.
OUTPUT_TABLE	The OUTPUT_TABLE modifier identifies the output file containing tabulated results.
DONE	The DONE modifier identifies the name of the *done* file to signal the SAC ESP that VADER has completed execution.
STOMP_TEMPLATE	The STOMP_TEMPLATE modifier identifies the STOMP template file to load releases into.
TYPE	The TYPE=ASCII modifier selection causes the file associated with the NAME modifier to be opened as an ASCII (text) file. The TYPE=BINARY modifier selection causes the file associated with the NAME modifier to be opened as a binary file.
NOCOMPUTE	The NOCOMPUTE modifier applies only when the STOMP_TEMPLATE modifier is used. If present, the NOCOMPUTE modifier causes VADER to skip STOMP template file processing.
NAME	The quote string associated with the NAME modifier contains the name of the file to be opened. File names are entered as text strings in double quotation marks and may contain path names up to 72 characters long. This modifier is required except when the FILE STOMP_TEMPLATE NOCOMPUTE option is used.
OPEN	If present, the OPEN=CREATE option causes the following actions to occur: • Deletion of any existing file by that name • Creation of a new file If the OPEN=CREATE option is not present, the following actions will occur: • Check if the file exists - if not, error terminate • Check for file record length consistency with expected record type – if incorrect, error terminate • Open the specified file in read-only mode If the OPEN=READONLY option is specified, the file is opened in read-only mode. If present, the OPEN=OVERWRITE option causes the following actions for the STOMP template file: • Check for existence of the file – if not present, error exit • Open the specified file

Modifier	Description
	• Read the file into a specified data structure • Fill the release data fields in the new file with the values calculated by VADER • Overwrite the file with new data and close the file

The following is additional information regarding file operations:

- File names for outgoing remedial actions are specified by the REMEDIATION keyword (see Section 5.4.14).
- The vader.run file name is hardwired and cannot be specified by a FILE keyword.
- The vader.debug file is not accessible via the keyword definitions. It captures debugging data generated by VADER via WRITE(*,*) and WRITE(IPRT,*) statements (IPRT is set to UNIT 6 for screen output). This file is written to the directory from which VADER is invoked.
- The vader.done file is written if the run completes successfully. If the run does not complete successfully, a vader.fail file is written instead. In either case, the vader.run file disappears. The analyst may refer to the vader.debug file for details and error messages.
- In many cases, VADER is run as a standalone process with no need to generate a STOMP source term. In that case, it is inconvenient to supply a STOMP_TEMPLATE file. Setting the keyword FILE STOMP_TEMPLATE NOCOMPUTE turns off all STOMP template processing. The NAME modifier is not required in this case.

The following set of example keywords from the VADER keyword file specifies a full set of files for running VADER for Realization 66 at Site 666 for Strontium-90. The name of the VADER keyword file is specified on the command line when VADER is invoked.

```
FILE INVENTORY  NAME="INV\Real66.inv"  OPEN=READONLY
FILE SAC_ESD  NAME="SAC_ESD.1999"  OPEN=READONLY
FILE LOG_FILE  NAME="Site_666_Sr90_Real66.log"  OPEN= CREATE
FILE STOMP_TEMPLATE  NAME="Site_666_Sr90_Real66.stp"  OPEN=OVERWRITE
FILE OUTPUT_TABLE  NAME="Site_666_Sr90_Real66.out"  OPEN=CREATE
TYPE=BINARY
FILE DONE  NAME=" vader.done"  OPEN=CREATE
```

If a FILE keyword line is absent, the default file names are set as shown in Table 5.7. The OPEN and File Type settings shown are the default settings.

Table 5.7 Default File Names for VADER

Default File Name	Where referred to	Use	OPEN	File Type
vader.key	Path name (command line)	Mandatory	READONLY	ASCII
inv.dat	vader.key	Mandatory	READONLY	ASCII
sac_esd	vader.key	Mandatory	READONLY	ASCII
input-esd	vader.key	Optional	OVERWRITE	ASCII

Default File Name	Where referred to	Use	OPEN	File Type
vader.log	vader.key	Mandatory	CREATE	ASCII
vader.table	vader.key	Mandatory	CREATE	ASCII or BINARY
vader.river	vader.key	Optional	CREATE	ASCII
vader.running	Internal	Mandatory	SCRATCH	ASCII
vader.done vader.fail	Internal	Mandatory	CREATE	ASCII
vader.debug	Internal	Mandatory	CREATE	ASCII

5.4.8 MAP Keyword for VADER

The optional MAP keyword provides a means for explicitly assigning a release model to an arbitrary waste form present in the INVENTORY. This method complements the implicit association of INVENTORY waste to a release model through the waste name, as in SOIL waste form mapped to the SOIL release model or the association of INVENTORY waste form to a release model through embedding the model name, as in SOILxx waste associated with the SOIL release model. The analyst can specify multiple MAP keywords. The syntax for this keyword is

```
MAP Waste="Quote₁"  To "Quote₂"
```

Examples (can have all of these in one vader.key file):

```
MAP Waste  “MUCK”  To “SOIL”
MAP Waste “MARL”  To “SOIL”
MAP Waste  “Mole”  To “CMNT”
```

The MAP keyword overrides implicit and embedded release model associations. For example, if the INVENTORY contains CMNTxx wastes, and no MAP keyword is present, the CMNT release model would be invoked. But if a MAP keyword like *MAP Waste “CMNTxx” To “SOIL”* were present, the CMNTxx waste would be released through the SOIL release model.

5.4.9 MODELS Keyword for VADER

The mandatory MODELS keyword specifies which release model simulations to run and supplies model release coefficients. Each release model requires its own MODELS line. Each model has its own unique set of coefficients. Table 5.8 provides the coefficients and expected physical units for each model. In practice, the control program SAC ESP (see Section 5.2.4) supplies a set of release model coefficients for a given realization by overwriting a vader.key template file with a set of model specifications and coefficient values.

The general syntax for the MODELS keyword is the following:

```
MODELS [SOIL | CMNT | CEMENT | CAKE | SLUDGE | REACTOR | CORE |
   LIQUID | RIVER] {STARTREL=N1} {STOPREL=N2} {NOCOMPUT} [Parameters …]
```

```
      [COLLAPSE|NOCOLLAP] COMMENT="Quote"
```

The data identified as [Parameters …] changes as the model changes. Table 5.8 describes the necessary coefficients for each model. The general syntax for each model is given in the following statements. In each model, the general statement includes the syntax "parameter name" = "numerical value for the parameter."

```
   MODELS SOIL [Kd=Nkd]  [Cs=Ncs]  {Qw=Nqw}  {B=Nb}  {Tw=Ntw} {A=Na} {H=Nh}
   MODELS CMNT [Dc=Ndc]  [Ac=Nac]  [Vc=Nva]
   MODELS CEMENT [Dc=Ndc]  [Ac=Nac]  [Vc=Nva]
   MODELS CAKE [Cs=Nds] {Qw=Nqw} [R=Nr] {A=Na}
   MODELS SLUDGE [Cs=Nds] {Qw=Nqw} [R=Nr] {A=Na}
   MODELS REACTOR [Fr=Nfr]
   MODELS CORE [Fr=Nfr]
   MODELS LIQUID [F=Nf]
   MODELS RIVER [F=Nf]
```

In some cases, the program may attempt to retrieve a value from the SAC-ESD file if a coefficient is not declared in the keyword file. An example is the exposed area A and depth h in the SOIL model, which should be the same for all analytes at a given site. Other candidates for retrieval from the SAC-ESD file are Q_w, T_w, and B.

Table 5.8 Modifiers Associated with the MODELS Keyword and Coefficients for Each Release Model

Release Model	**Coefficients[a]**	**Modifier Description**	**Units**

SOIL	Kd=*nnn*
	Cs=*nnn*
	Cs=*nnn*
	Qw=*nnn*[b]
	Tw=*nnn*
	B=*nnn*
	A=*nnn*
	H=*nnn*
		Linear equilibrium sorption coefficient to source zone soil.
		Aqueous solubility of the analyte (radionuclide).
		Aqueous solubility of the analyte (nonradionuclide).
		Darcy flux density of water flowing through source zone (recharge).
		Volumetric water content fraction (0<Tw≤1).
		Bulk density of source zone soil.
		Effective cross-sectional area of contaminant source zone.
		Average vertical thickness of contaminant source zone.
		Note: Cs (solubility) units must reflect the units of inventory. Example: If inventory is in kg, then Cs must be in kg/cm^3.
			cm^3/g
			Ci/cm^3
			kg/cm^3
			cm/yr[b]
			unitless
			g/cm^3
			cm^2
			cm

CMNT or CEMENT (synonyms)	Dc=*nnn*
	Ac=nnn
	Vc=nnn
		Diffusion coefficient of contaminant within cement waste
		External surface area of cement waste form in source zone
		Volume of cement waste form in source zone (Vc>0).
			cm^2/yr
			cm^2
			cm^3

CAKE or SLUDGE (synonyms)	Cs=*nnn*
	Qw=*nnn*[b]
	A=*nnn*
	R=nnn
		Aqueous solubility of major structural component of salt cake (NaOH)
		Darcy flux density of water flowing through source zone (recharge)
		Effective cross-sectional area of cake or sludge waste form
		Density of major structural matrix, for converting waste volume from the inventory file into inventory mass units. Default 1.5 g/cm^3
		Note: the mass of structural matrix (salt cake) must be in same mass units as the solubility of the structural matrix (Cs). Currently, inventory is expected to have structural matrix volume in m^3, and density R is used to convert from cubic meter to grams, so enter Cs in g/cm^3.
			g/cm^3
			cm/yr[b]
			cm^2
			g/cm^3

Release Model	**Coefficients[(a)]**	**Modifier Description**	**Units**
REACTOR Or CORE (synonyms)	Fr=*nnn*	Initial fractional release rate for contaminant from **graphite core** waste form. Fr ranges from 0 to 1.	1/yr
LIQUID	F=*nnn*	Instantaneous Fractional Release Rate (0 to 1) for liquids released to the vadose zone (STOMP). When *nnn*=1, 100% pass-through. **Note: Quantity released is not decayed.**	unitless
RIVER	F=nnn	Instantaneous Fractional Release Rate (0 to 1) for liquids released to the river (GWDROP). When *nnn*=1, 100% pass-through. **Note: Quantity released is not decayed.**	unitless
STARTREL	=[StartYear][(c)]	Year the release mechanism is activated. Each MODEL line may have this keyword. Applies to all waste in this waste form. Defaults to the START entry for the PERIOD keyword (see Section 5.4.11).	yr
STOPREL	=[StopYear][(c)]	Year A.D. release mechanism is deactivated. Each MODEL line may have this keyword. Applies to all waste in this waste form. Defaults to the STOP entry for the PERIOD keyword (see Section 5.4.11).	yr
NOCOMPUT	N/A	If present, do not compute this model (just do error-checking).	N/A
NOCOLLAP	N/A	Do not collapse all variant waste names found in inventory. Use separate MODELS lines for each unique waste form. The default action is COLLAPSE, in which all variant waste forms in the same waste form class are pooled into one super waste form and run with the MODELS entry.	N/A
COMMENT	=“text”	An optional text string used to annotate the model.	N/A

(a) Coefficient definitions are taken from Table D.1 of Appendix D in Kincaid et al. (1998).

(b) A single value for recharge rate to apply over the entire simulation period as provided by STARTREL and STOPREL may be entered as shown here. An entry for recharge on the MODELS keyword overrides any RECHARGE keywords. The SOIL and CAKE release models require recharge rate Q_w in cm/yr. Several values for Q_w over different time periods can be entered via RECHARGE keywords.

(c) There is no provision for intermittent release and nonrelease over several time periods in the same model, but different models can have different release periods. This value defaults to the entry on the PERIOD keyword.

The release mechanism in a release model is switched on at a given start date and continues to a given stop date. These dates can be specified in two ways: through the STARTREL and STOPREL dates in each MODELS keyword line, or if these modifiers are not present, the dates default to the entries in the PERIOD keyword (see Section 5.4.11). Although inventory and remediation actions may take place at dates outside those specified using STARTREL and STOPREL, the actual release mechanism starts up at the date given by STARTREL and ends at the date given by STOPREL. Calculations for release from containment take place only between these dates.

The following are some example MODELS keyword entries (using default start and stop dates):

```
   MODELS SOIL Kd=.0001  Cs=.0003  Qw=25  B=2.2  Tw=.05 COMMENT="Joey's Mud Model"
   MODELS REACTOR Fr=.0045   COMMENT="Good Ship USN Lollipop Reactor Block"
```

The following three MODELS keyword entries specify start and stop dates for two models at the same site. The LIQUID model case pertains to a leak of one year's duration. The first CAKE model implies that the release mechanism was halted after 44 years. The next CAKE model implies that the release mechanism was activated in 1975 and continues to run until the end of the simulation as specified by the PERIOD keyword. If STOPREL is absent, the release mechanism is considered active until the ending date from the PERIOD keyword.

```
MODELS LIQUID F=1 COMMENT="Pass Liquids Through" STARTREL=1956 STOPREL=1956
MODELS CAKE Ds=.06 Qw=45 R=2.1 A= 1.5e6   STARTREL=1956 STOPREL=2000
MODELS CAKE Ds=.06 Qw=45 R=2.1 A= 1.5e6   STARTREL=1975
```

If a table of time-dependent recharge rates is entered, the analyst uses the RECHARGE keyword (see Section 5.4.13) and may skip entry of Q_w in the MODELS keyword for SOIL and for CAKE release models. If both Q_w from either the SOIL or CAKE model and the RECHARGE keywords are present, the RECHARGE keywords supercedes and overwrites all entries of Q_w in the MODELS keywords.

Inputs from the INVENTORY file and from input remediation files can have variant names for their wastes. For example, at site Cairo, the inventory has SOILF and SOILG waste forms, and the remediation file from Alexandria identifies the SOILA waste form. All these wastes are of generic waste model SOIL.

VADER has a provision to optionally collapse all these waste inventories into one pool of soil wastes and run a single SOIL release model on them, as provided by one MODELS SOIL line. This is the default option. The output is pooled under one SOIL section in the vader.table file and is also pooled into the AGGREGAT section of the vader.table file. VADER looks for the generic release model name embedded in the waste name.

Alternatively, VADER can run a SOIL model with specific release parameters on each soil's waste type variant. This requires a MODELS line for each soil inventory with the NOCOLLAPSE option set (see Table 5.8). The following three keywords show an example to invoke the SOIL release model with different parameters for each inventory:

```
MODELS SOILA Kd=.0001  Cs=.0003  Qw=25  B=2.2  Tw=.05 COMMENT="Joey's Sand Model"
MODELS SOILB Kd=.002    Cs=0.003  Qw=25 B=2.2 Tw=0.15  COMMENT="USN Garden Soil"
MODELS SOIL    Kd=.002    Cs=0.05    Qw=25 B=2.2 Tw=0.15  COMMENT="Mississippi Mud"
```

With these variant soil release model specifications, VADER will apply the appropriate one to each inventory SOIL, SOILF, and SOILG as found in the inventory after remedial action inputs are read. The outputs are given their own sections in the vader.table file following the exact waste name, and are also pooled into the AGGREGAT section of the vader.table file because SOIL is embedded in all three waste names.

5.4.10 PATH Keyword for VADER

The optional PATH keyword overrides the command-line argument path name. The PATH keyword can be used to specify the path name to the working directory to which to write output files. One and only one PATH keyword line can be entered. The default setting is to place the output files in the VADER working directory. The syntax for this keyword is the following:

```
PATH “pathname”
```

The following is an example to place the output files in a directory identified by Site, Analyte, and Realization:

```
PATH "C:\SAC\PayRun\VADER\Site001\Sr90\Real001"
```

5.4.11 PERIOD Keyword for VADER

The PERIOD keyword identifies the start and stop years for the release simulation. Only one PERIOD keyword line can be entered. The syntax of the keyword follows:

```
PERIOD [START=Year1]  [STOP=Year2 ]  [CLOSURE=Year3]
```

The start of the simulation period is identified by the modifier START and the value Year1. The end of the simulation period is identified with the modifier STOP and the value Year2 , which must be greater than Year1. Start and stop years should be entered as whole numbers. The year of site closure is identified by the modifier CLOSURE and the value Year3. The closure year does not affect any calculations in VADER and is included for reference only. The year of site closure cannot be smaller than the start year but can be larger than the stop year. The following is an example PERIOD keyword that simulates releases from 1944 through 3050 with site closure occurring at 2050:

```
PERIOD START=1944 STOP=3050 CLOSURE 2050
```

5.4.12 REALIZATION Keyword for VADER

The REALIZATION keyword identifies the realization to be simulated. One or two values may be entered. The general syntax of the keyword is

```
REALIZATIONS N1 {N2}
```

The value N1 is the integer realization number for the current run. The optional argument N2 is the maximum realization number for an ensemble of realizations such that N1 is 0 or larger, and N2, when present, is greater than or equal to N1. An example for the 99th realization out of a planned set of 200 realizations is

```
REALIZAT 99 200
```

The value for N2 is supplied to tell VADER that leading zeroes may be needed in a realization directory name. In a typical SAC run, the path in the command line when VADER is invoked (see Section 5.3.2) contains the realization number. The directory naming convention is for all realization directories to have the same number of characters. This means that a sequence of the form /01, /02, /03, …, /24, /25 is expected in the case where the maximum realization number is 25. When N2 is missing, VADER treats the directory name as N1 without leading zeroes. If N2 < N1, N2 is set to N1. VADER handles realization numbers in the range 0 to 999.

5.4.13 RECHARGE Keyword for VADER

This optional keyword allows entry of time-dependent Q_w recharge value at a given site as a lookup table. Multiple RECHARGE keywords can be entered, but VADER only processes the first 64. Recharge values entered by the RECHARGE keywords are global and apply over all release models for which the Q_w parameter is not entered specifically for a SOIL or CAKE release model (see Section 5.4.9). The general syntax is

```
RECHARGE [START=Year1]  [END=Year2]  [Qw] {UNIT="quote1"}
```

The value Year1 associated with the START modifier is the starting year for the recharge rate. The value Year2 associated with the END modifier is the ending year for which Q_w is effective. The value Year2 must be greater than or equal to the value Year1. The value Q_w is the net recharge rate (infiltration rate) at the site. The default units are cm/yr. The UNIT modifier supplies units such as cm/yr, m/yr, etc. If units are declared to be other than cm/yr, a conversion to cm/yr is performed if the conversion factors are present. Currently, conversion from m/yr, mm/yr and in/yr to cm/yr are implemented. If the START and END modifiers are not present on the RECHARGE keyword, the START and END years from the PERIOD keyword (see Section 5.4.11) apply. If the supplied recharge values do not cover the entire span of years needed, the entered values are extended to earlier or later years as needed to cover all years of simulation.

If a value for Q_w is entered for a specific release model in the MODELS keyword, it overrides RECHARGE keyword entries for calculations pertinent to that release model. When no RECHARGE lines are present, the Q_w values specific to each release model keyword entry (see Section 5.4.9) are used over the entire simulation period. There is no attempt to reconcile SOIL model Q_w against CAKE model Q_w, as it is conceivable that it is desirable to use different recharge rates for the two release models. When no RECHARGE keywords are present and no MODELS keyword entries for models that use recharge rate (SOIL and CAKE) are present, VADER declares an error and terminates.

In the following example, a lookup table is built with two entries for Q_w, one with Q_w=23.45 cm/yr applying over 1944 through 1988, and one with Q_w=12.34 cm/yr applying over 1989 through 3050, for all models including the CAKE model.

```
MODELS CAKE A=1234 Cs=0.2345 R=3.33
RECHARGE START=1944 END=1988 Qw=23.45 UNIT="cm/yr"
RECHARGE START=1989 END=3050 Qw=12.34
```

The same result as above is achieved in the next example, except for the CAKE model. The Q_w=20 cm/yr entry in the MODELS card overrides the recharge table entries. Also illustrated is that Q_w=12.34 cm/yr is used from 2051 to the 3050, the ending simulation period year. To have Q_w be set to a different value, it must be set explicitly.

```
PERIOD START=1944 STOP=3050 CLOSURE=2050
MODELS CAKE A=1234 Cs=0.2345 Qw=20 R=2.25
RECHARGE START=1944 END=1988 Qw=23.45
RECHARGE START=1989 END=2050 Qw=12.34
```

In the following example, a lookup table for the recharge rate is built with one entry Q_w=20 cm/yr covering the entire simulation period as given in the PERIOD keyword.

```
MODELS CAKE A=1234 Cs=0.2345 R=1.11 Qw=20 (no RECHARGE keywords)
```

The CAKE model requires a groundwater recharge. In the following example, where no RECHARGE keywords are entered, VADER prints an error message and terminates the run.

```
MODELS CAKE A=123456 R=1.23 Cs=0.2345     (and no RECHARGE keywords)
```

5.4.14 REMEDIAT Keyword for VADER

The optional REMEDIAT keyword has the following two functions. First, it indicates the input files that define incoming (imported) remediation quantities from previously computed sites. Second, it provides information for calculating outgoing (exported) remediation quantities, specifying sites to be computed later, and writing an output remediation file in the working directory.

The following is the REMEDIAT keyword's syntax:

```
REMEDIAT {YEAR=N1} [FROM="quote1"] [TO="quote2"] {SOIL=N1} {CAKE=N2}
{CEMENT=N3} {CORE=N4} {LIQUID=N5} {RIVER=N6}
```

VADER always remediates waste in the working site inventory. In contrast, the STOMP code remediates from the vadose zone. This creates the possibility of transfers to sites of quantities from previous runs of both VADER and STOMP. Hence, the REMEDIAT keyword has two forms, one for imports and the other for exports. The import form needs to have only enough information to locate the previously computed REMEDIAT and vader.rem files. The export form needs to specify the export site and specify the information necessary to calculate the quantities of wastes to transfer each year.

The radioactive quantities in imported remediation transfers are assumed to be already decayed to the stated year. Each previously computed site specifies the quantity in kg or Ci to be exported to the working site at the specified effective year. Table 5.9 describes the modifiers for the REMEDIAT keyword that apply.

Table 5.9 REMEDIAT Keyword Modifiers in VADER Associated with Imports to the Working Site

Modifier	**Description**
FROM	The quote string associated with the FROM modifier identifies the previously computed site from which remediation data are being transferred to the working site.
TO	The quote string associated with the TO modifier identifies the working site. It is the site to which remedial action waste will be imported. The FROM and TO sites must be different.

As an example, let the working site be named *Memfis*. The following REMEDIAT keyword identifies that waste will be imported from the separate previously computed site named *Dump*.

```
   REMEDIAT FROM="Dump" TO="Memfis"
```

VADER constructs path names to point to two files, one written by VADER and one written by STOMP in previously computed site directories (see Section 3.2.2 for a directory structure discussion). VADER attempts to read both files. When VADER identifies the TO site as the working site, it builds the path name from the FROM site, and then ignores all other data entered in the REMEDIAT keyword line. If the file cannot be opened, VADER indicates an error message and proceeds; it is not a fatal error to not find an incoming remediation file. The input files contain one or more records that provide transfer date, analyte, quantity, and waste form for a given transfer in a given year.

STOMP-built remediation files and VADER-built vader.rem files have essentially the same format but vary slightly in details described below. For example, these files are internally identified as to source and to time units (VADER defines time in calendar years; STOMP defines time in elapsed seconds from the starting of the run).

For export, each REMEDIAT keyword line identifies a separate transfer from the working site to the export site during a given year. Each line also specifies the export fraction of each waste form in current inventory to transfer. Because material might be exported to several sites during a given year, more than one keyword line per year per export site is allowed. See Section 5.2.4 for a discussion on allowable sequences of remedial actions. Export fractions are used to determine how much of the remaining quantity at a site is to be taken away in the remedial action. These fractions are interpreted as the fraction of remaining inventory to export at the given year. The quantity to export in the current year is based on multiplying the net cumulative inventory remaining after the previous year’s release and remedial actions by the export fraction. Outgoing quantities are decayed to the export year. Table 5.10 describes the modifiers for the REMEDIAT keyword for remedial action exports. One or more waste forms may be set for exports. The export fractions for unspecified waste forms are set to 0.

Table 5.10 REMEDIAT Keyword Modifiers in VADER Associated with Exports from the Working Site

Modifier	**Description**
YEAR	The numerical value associated with the YEAR modifier identifies the year in which the remediation action takes place. A separate record must be entered for each year a remediation event occurs. This modifier is required for remediation exports.
FROM	The quote string associated with the FROM modifier must be set to the working site. It identifies the release site from which waste or soil will be exported.
TO	The quote string associated with the TO identifies the downstream site to which material will be exported. The FROM and TO sites must be different.
SOIL	The numerical value associated with the SOIL modifier identifies the fraction of remaining soil-debris waste to export. Valid values are in the range 0 to 1. VADER is currently not programmed to recognize variant names such as SOILxx.
CAKE	The numerical value associated with the CAKE modifier identifies the fraction of remaining salt cake/sludge waste to export. Valid values are in the range 0 to 1. The SLUDGE and SALT modifiers can be used in place of the CAKE modifier.

Modifier	Description
CMNT	The numerical value associated with the CMNT modifier identifies the fraction of remaining cement waste to export. Valid values are in the range 0 to 1. The CEMENT modifier can be used in place of the CMNT modifier. VADER is currently not programmed to recognize variant names such as CMNTxx.
CORE	The numerical value associated with the CORE modifier identifies the fraction of the remaining reactor/component waste to export. Valid values are in the range 0 to 1. The REACTOR modifier can be used in place of the CORE modifier.
LIQUID	The numerical value associated with the LIQUID modifier identifies the fraction of remaining liquid waste to export. Valid values are in the range 0 to 1.
RIVER	The numerical value associated with the RIVER modifier identifies the fraction of remaining liquid waste to export. Valid values are in the range 0 to 1.

The following is an example remediation keyword that exports portions of the SOIL, CAKE, and CMNT wastes from the site MEMFIS to the site Kairo in the year 1960:

```
REMEDIATION YEAR=1960 FROM="MEMFIS" TO="Kairo" SOIL=.11 CAKE=.28 CMNT=.15
```

For exports from the current working site, VADER generates the vader.rem file, with lines for exports to one or more sites each year, and writes it by default to the current VADER working directory. These output files are to be picked up by succeeding VADER runs with the files pointed to in REMEDIAT keyword lines. Because these remediation files are written in the current working directory, subsequent VADER runs rely on the FROM arguments for their own sets of REMEDIATION keywords to find all the files.

The following two REMEDIATE keywords provide an example of outgoing remedial actions. The first line tells VADER to subtract 45% of the remaining Strontium-90 in the soil-debris waste remaining at the year 2022 from the Memfis inventory and to write a vader.rem file in the working directory (to be picked up by a subsequent VADER run for the Thebes site). If the soil-debris inventory is 432 Ci, VADER calculates the quantity transferred out as 194.4 Ci, and the remaining inventory quantity effective the next year, 2022 AD, as 237.6 Ci (not adjusting for decay in this simple example). When fractions of inventory are to be transferred to several sites in one year, the quantities are all calculated from the initial inventory quantity for that year. Therefore, all fractions for a given waste form must add to no more than 1 over a given year. In this example, transferring 45% to Thebes and 55% to Kairo completely eliminates the SOIL waste (moves it all somewhere else) at Memfis and replaces it with clean soil.

```
REMEDIATE YEAR=2022 FROM="Memfis" TO="Thebes"  SOIL=0.45
REMEDIATE YEAR=2022 FROM="Memfis" TO="Kairo"    SOIL=0.55
```

The resulting records written in the vader.rem file (as per Table 5.15) would be

```
2022, yr, Thebes, sr90, 194.4, SOIL
2022, yr, Kairo,   sr90, 237.6, SOIL
```

The VADER run at working site Thebes will have been told to look for a vader.rem file in the Memfis working site directory (for this realization and analyte). VADER would scan the records, find the Thebes entry or entries, and add them, in this case, to the SOIL inventory, under the canonical SOIL waste form name.

Remediation files read by VADER typically were built by previous VADER or STOMP runs. Remediation files built by VADER are intended to be read by subsequent VADER runs. VADER never reads VADER-built remediation files residing in the working directory.

5.4.15 TITLE Keyword for VADER

The TITLE keyword defines a problem title to be written to output files. Titles up to 132 characters long are supported. There are no modifiers associated with the TITLE keyword. This keyword is optional and the default title is VADER RUN. The following is the syntax for this keyword record:

```
   TITLE ["quote"]
```

The title is entered in a double-quote string. The following example defines a title for a run of the code.

```
   TITLE "Scenario: Release Sr90 at Site 666 Realization 99"
```

5.5 Data Files

For a given site, analyte, and realization, all input files for both VADER and STOMP are placed in one working directory. All files generated by VADER and STOMP are written to this working directory. The path name to the working directory is provided to VADER as the command-line argument (see Section 5.3.2). This section describes more thoroughly the files accessed and written by VADER in terms of format and contents. Table 5.11 provides an overview of all files used by VADER.

Table 5.11 Overview of Files Used by VADER

File	I/O	Contents	Notes
vader.key	I	Realization, site, and analyte specifics, release model coefficients, remediation information, recharge rates, non-default file names, site release simulation period, and more.	Provides command input to VADER. Organizes and conducts the run. See Section 5.4 for further information.
inventory	I	Inventory added per year over all years from beginning of site operation to site closure.	Supplies the inventory of material to released.
input-esd Copied to input	I/O	STOMP input keyword file processed by subroutine VZRSTP. To contain time series of releases to the vadose zone for aggregate solids, liquids, and also liquids volumes.	Provides STOMP the source term or boundary conditions in quantity per second.
vader.river	O	Time series of direct releases to the river in quantities per year.	To be processed by GWDROP.

File	I/O	Contents	Notes
remediate	I	Incoming remedial actions from previous STOMP runs in quantities per year.	Import: Transfer material to working site inventory.
vader.rem	I	Incoming remedial actions from previous VADER runs site in quantities per year.	Import: Transfer material to working site inventory.
vader.rem	O	Outgoing remedial actions from the working site to other sites.	Export: Transfer material from the working site to another site.
vader.table	O	Time series of quantities released, quantity remaining, decayed, annual, and cumulative.	Provide output release and mass-balance data.
vader.running	S	Scratch file written by VADER upon initiating execution. Disappears when VADER finishes.	Signals SAC ESP that VADER is running.
vader.done	O	ASCII file, existence polled by SAC ESP and STOMP. Contains special run information for SAC-ESD to direct further processing.	Signals SAC ESP when the VADER run is complete.
vader.fail	O	ASCII file, existence polled by SAC ESP. Contains special run information for SAC to direct further processing.	Signals SAC ESP that the VADER run failed.
vader.log	O	Inventory results, processing information, decisions, and errors.	To document each VADER run.
vader.debug	O	Record error messages and intermediate results in case VADER crashes.	To assist in debugging VADER and/or inputs.

5.5.1 VADER Input files

A more detailed description of the VADER input files is provided in this section. Example files are provided for most of the files.

5.5.1.1 vader.key File

Section 5.4 describes the vader.key file keywords. Table 5.12 provides an example file. This file is immediately opened and parsed for command information when VADER is invoked.

Table 5.12 Example vader.key File

```
! Comment lines are indicate by the exclamation point

TITLE "SAC GW/VZ Alternate Assessment Run 1 Pyramids  VADER.key 12/21/2002”
AGGREGATE ID="Memfis"  TITLE="Pharaoh’s Winter Home"
ANALYTE ID="Sr90" HALFLIFE=28.72 TYPE="NR"
CUTOFF  Analyte=1.e-12 Volume=1.e-9
DELTAT 1.0 !Default iteration step size is 1 to represent one year.
PERIOD START=1944  CLOSURE=2050 STOP=3050
REALIZATION  01  25

! VADER processes each model in order presented
! All possible models are shown here
MODELS Cement  STARTRel=1959 STOPREL=2050 Ac=1000 Vc=10000 Dc=2.42
MODELS Soil       A=1500 h=50 B=2.12 Kd=0.023 Cs=1.13e-04 Tw=0.55
MODELS Cake     StartRel=1953 A=200. Cs=0.022 COMMENT “Use Default Density”
MODELS Liquid   F=1.00 COMMENT "Tank Liquors”
MODELS Glass    StartRel=1956 Fg=0.01
MODELS Reactor StartRel=1977 Fr=0.02
MODELS River    StartRel=1966 F=1.0

! Default recharge units are cm/yr.
RECHARGE START=1944 END=1963 12.34
RECHARGE START=1964 END=1987 32.21
RECHARGE START=1988 END=3050 34.56

! Incoming remediations
REMEDIATION FROM="SXTANK” TO="MEMFIS"

! Outgoing remediations
REMEDIATION YEAR=1960 FROM="MEMFIS" TO="Kairo" SOIL=.11 CAKE=.28 CMNT=.15
  REACTOR=0.13 GLASS=.3 LIQUID=0
REMEDIATION YEAR=1961 FROM="Memfis" TO="KAIRO" SOIL=.27

END "End of the example file"
```

No FILES lines will be needed for normal runs in the SAC GW/VZ context because VADER is programmed to look for files having default names in the directory indicated by the command line argument.

5.5.1.2 Inventory File for Use in VADER

The inventory file is generated by the INVENTORY Module. It contains quantities of analyte in each waste form added to the site inventory each year (see Section 4.5.2.4). Inventories are assumed to be already decayed to the year they become available in the inventory file. This implies that the total amount available subject to release mechanisms is the net accumulation of all the annual input amounts each year, decayed to that year.

Table 5.13 provides an example inventory file. This file is generated by Inventory (see Section 4.5.2.4 for a detailed description of the file structure). One file with all sites and analytes is written for each realization. Fields within each record are separated by commas and character items are defined explicitly by quotes. VADER uses keyword-parsing routines to extract annual quantities as source terms for the release models.

Table 5.13 Example Inventory File

```
1, 3,"Go For Broke" (Realization 1, three sites, the Go For Broke Analysis)
"Site 1",7               (aggregation site 1 has data for seven years)
1944,1,2,"m^3","liquid",1.E3   (in 1944, 1 waste type, 2 contaminants, 1E3 m^3
of liquid waste)
"Sr90", "Ci",2.                         (2 Ci of Sr-90 in the liquid release)
"U234","Ci",500.                     (500 Ci of U-234 in the liquid)
1945,2,2,"m^3","liquid",2.E3,"glass",100., (2000 m^3 liquid, 100 m^3 of glass)
"Cs137","Ci",10.,500             (10 Ci in liquid and 500 Ci in glass)
"Sr90", "Ci",2.,300
1946,1,2,"m^3","liquid",200.        (in 1946, 200 E3 m^3 of liquid waste)
"Sr90", "Ci",2.                        (2 Ci in liquid)
```

```
"U234","Ci",500.
1947,1,2,"m^3","liquid",10.,
"Cs137","Ci",40.
"Sr90", "Ci",2.
"Site 2",3                         (aggregation site 2 has data for 3 years)
1948,1,3.","m^3",liquid,1050.,
"Cs137","Ci",40.
"Sr90", "Ci",2.
"H3","CI",2000.
1949,3,2,"m^3","liquid"2000.,"cmntf",200.,"cmntg",100,
"Sr90", "Ci",2.,30.,87
"H3","Ci",2000.,150.,78
1950,3,2,"m^3","liquid",2000.,"cmntf",100.,"cmntg",205,
"Cs137","Ci",40.,10.,555
"Sr90", "Ci",2.,30.,444
"Site 3",2                             (aggregation site 3 has data for 2
years)
1951,2,3,"m^3","liquid",13000.,"cmntg",666
"Cs137","Ci",40.,1.1
"Sr90", "Ci",2.,2.2
"H3","Ci",2000.,0.0
1952,2,2,"m^3","liquid",18000.,"cmntf",78,
"Cs137","Ci",40.,55.5,
"CCl4","kg",23.45,43.21
```

It is expected that the waste form names in the Inventory file can be associated with the release model names in the MODELS keyword. To this end, VADER makes waste form and release model associations as per Table 5.14.

Table 5.14 WASTE Form to Release Model Mapping in VADER

Waste Form (INVENTORY)	**Canonical Waste Form (VADER)**	**Release Model Name (vader.key)**
SOILxx[(a)]	SOIL	SOIL
GLASS	GLASS	GLASS
CEMENT	CMNT	CMNT
CMNTxx[(a)]	CMNT	CMNT
REACTOR	REACTOR	REACTOR
CORE	REACTOR	REACTOR
CAKE	CAKE	CAKE
SLUDGE	CAKE	CAKE
RIVER	RIVER	RIVER
LIQUID	LIQUID	LIQUID

(a) The “xx” in a waste form name indicates VADER can accept variants of that waste form name.

Table 5.14 shows that there are two mapping operations. Variant release model names (such as CEMENT, CMNT, and CMNTxx) are mapped to the canonical release model name. Variant waste form names are mapped to a canonical waste form name, which has embedded in it the exact canonical release model name to apply.

VADER makes release model name comparisons in a case-insensitive manner. VADER expects exact (case-insensitive) matches when the COLLAPSE option is false and can perform root waste form name matches for waste variants when the COLLAPSE option is true (the default). The COLLAPSE option is set using the MODELS keyword (see Section 5.4.9). For example, when the COLLAPSE option is true, the cement waste form variants *cmntf* and *cmntg* are pooled into root waste form *cmnt* because the generic release model *cmnt* is embedded in each waste name.

5.5.1.3 Remediation Files for Use in VADER

STOMP and VADER build remediation files in the same format. VADER reads both kinds of remediation files and performs date conversion to years based on the stated time units for each record.

Table 5.15 shows an example STOMP-built remediation file written at site *116-B-1* to specify import of 0.02902 Ci of strontium-90 into the VADER working site (*600-148*) SOIL inventory in the year 1999. Because a waste form name is not stated, the waste form is assumed to be SOIL, following the STOMP assumptions. The year 1999 is expressed in elapsed seconds from simulation start date (00:00:00 January 1, 1944) to 00:00:00 January 1, 1999, to conform with STOMP practice. Waste units are assumed to be either Curies for radioisotopes or kg for nonradioactive chemicals. The following is the vader.key keyword that informs VADER to read this file (see Section 5.4.14):

```
REMEDIATE YEAR=1999 FROM="116-B-1" TO="600-148"
```

Table 5.15 Example STOMP-built Remediation File

```
~STOMP Remediation Transfers Report
#SAC STOMP input created by ESP   05/18/2001 - 16:58:35
#SAC Case ID          : SAC Rev. 0 Initial Assessment
#SAC Template         : stomp\116-B-1\template.key
#SAC Site ID          : 116-B-1
#SAC Site NW Easting  :   565517.8
#SAC Site NW Northing :   145314.3
#SAC Site SE Easting  :   565559.3
#SAC Site SE Northing :   145272.8
#SAC Analyte          : Sr90
#SAC Realization      :     1
1,
 1.735689600000E+09,s,600-148,sr90, 2.902E-02,
```

Table 5.16 shows an example VADER-built remediation file at working site 116-B-1 to export strontium-90 to site 600-148. The effective years are expressed in years AD.

Table 5.16 Example VADER-Built Remediation File

```
~VADER Remediation Transfers Report
 #SAC VADER template keyword file for the 116-B-1 site
 #SAC Run Date         : 05.21.2001 - 16:08:35
```

```
#SAC Case_ID          :
#SAC Keyword File in  : Vader\116-B-1\
#SAC Site ID          : 116-B-1
#SAC Site NW Easting  : 000.000
#SAC Site NW Northing : 000.000
#SAC Site SE Easting  : 000.000
#SAC Site SE Northing : 000.000
#SAC Analyte          : Sr90
#SAC Realization      :  1
2,
1999,yr,600-148,sr90, 23.45, SOIL
2000,yr,600-148,sr90, 54.32
2001,yr,CoreDump,sr90,555,CORE
```

The REMEDIAT keyword (see Section 5.4.14) allows exports from all waste forms in the inventory by specifying the fraction of the current cumulative inventory. The file in Table 5.16 is built by remediation commands expressed as

```
REMEDIATE YEAR=1999 FROM="116-B-1" TO="600-148"     SOIL=0.45
REMEDIATE YEAR=2000 FROM="116-B-1" TO="600-148"     SOIL=0.80
REMEDIATE YEAR=2001 FROM="116-B-1" TO="CoreDump" CORE=1.00
```

The data in Table 5.16 are explained as follows. The 0.45 and 0.80 are arbitrary examples. At 1999, let the Sr-90 net cumulative inventory in soil at the working site 116-B-1 be 52.11 Curies. In 1999, 45% (23.45 Ci) of this is sent to site 600-148, leaving the inventory at 28.66 Curies. By 2000, this has decayed to 27.97 Ci. In 2000, 80% (22.38 Ci) of this quantity is transferred to site 600-148, leaving 5.59 Ci in the working site soil inventory. Other waste forms can be transferred this way. In 2001, a reactor core at 116-B-1, with 555 Ci of Strontium-90, is transferred to a special CoreDump site. The transfer accounts for 100% of the core inventory.

In the first line of remediation transfers in Table 5.15, SOIL is stated explicitly as the waste form. VADER would then recognize the 23.45 Ci of waste to be passed to the SOIL inventory. In the second line, *SOIL* has been left off. In this case VADER will default to the SOIL waste. In the third line, VADER has been told the transfer material is a CORE waste form.

5.5.1.4 STOMP Input Template as Modified by VADER

The STOMP input file is named *input-esp* by default. This file supplies VADER a template into which the release source term is written for subsequent use by STOMP. VADER copies the file *input-esp* to the file *input* and then adds lines that express the source term information. Currently, VADER adds three types of information to the end of *input* file: the source term related to solids released to the vadose zone, the source term related to liquids released to the vadose zone, and the volume of liquids released. All of these releases are provides as rates (amount per second). Table 5.17 provides an example for excerpted records for this file after modification by VADER.

Table 5.17 Excerpted Records from an input-esp File Modified by VADER

```
#SAC STOMP input created by ESP   03/11/2002 - 15:14:28
#SAC Case ID          : SAC Rev. 0 Shakedown2 Assessment
#SAC Template         : vadose/216-A-1/template_stomp.key
```

```
#SAC Site ID          : 216-A-1
#SAC Site NW Easting  :   575517.1
#SAC Site NW Northing :   136086.4
#SAC Site SE Easting  :   575526.3
#SAC Site SE Northing :   136077.2
#SAC Analyte          : Tc99
#SAC Realization      :     1

~Simulation Title Card
1,
216A-6,
W E Nichols,
Pacific Northwest National Laboratory,
January 9 2002,
8 AM PST,
2,
Template 216A - For shallow disposal sites (e.g., Cribs, Burial Grounds)
South 200E (Purex/BC Cribs), Cribs/Burial Grounds, Waste Chemistry Desig 6

                           … (records deleted) …

~Source Card
# New Source card added by VADER: 03/13/2002 - 17:02:08
2,
Aqueous Volumetric,1,1,1,1,580,580,27,
0.00000E+00,s,0.00000E+00,m^3/s,
0.00000E+00,s,0.00000E+00,m^3/s,
3.16224E+07,s,0.00000E+00,m^3/s,
3.16224E+07,s,0.00000E+00,m^3/s,
6.31584E+07,s,0.00000E+00,m^3/s,
6.31584E+07,s,0.00000E+00,m^3/s,
9.46944E+07,s,0.00000E+00,m^3/s,
9.46944E+07,s,0.00000E+00,m^3/s,
1.26230E+08,s,0.00000E+00,m^3/s,
1.26230E+08,s,0.00000E+00,m^3/s,
1.57853E+08,s,0.00000E+00,m^3/s,
1.57853E+08,s,0.00000E+00,m^3/s,
1.89389E+08,s,0.00000E+00,m^3/s,
1.89389E+08,s,0.00000E+00,m^3/s,
2.20925E+08,s,0.00000E+00,m^3/s,
2.20925E+08,s,0.00000E+00,m^3/s,
2.52461E+08,s,0.00000E+00,m^3/s,
2.52461E+08,s,0.00000E+00,m^3/s,
2.84083E+08,s,0.00000E+00,m^3/s,
2.84083E+08,s,0.00000E+00,m^3/s,
3.15619E+08,s,0.00000E+00,m^3/s,
3.15619E+08,s,0.00000E+00,m^3/s,
3.47155E+08,s,0.00000E+00,m^3/s,
3.47155E+08,s,2.76509E-06,m^3/s,
3.78691E+08,s,2.76509E-06,m^3/s,
3.78691E+08,s,0.00000E+00,m^3/s,
3.78691E+08,s,0.00000E+00,m^3/s,
Solute,Tc99,1,1,1,1,580,580,27,
0.00000E+00,s,0.00000E+00,1/s,
0.00000E+00,s,0.00000E+00,1/s,
3.16224E+07,s,0.00000E+00,1/s,
3.16224E+07,s,0.00000E+00,1/s,
6.31584E+07,s,0.00000E+00,1/s,
6.31584E+07,s,0.00000E+00,1/s,
9.46944E+07,s,0.00000E+00,1/s,
9.46944E+07,s,0.00000E+00,1/s,
1.26230E+08,s,0.00000E+00,1/s,
1.26230E+08,s,0.00000E+00,1/s,
1.57853E+08,s,0.00000E+00,1/s,
1.57853E+08,s,0.00000E+00,1/s,
1.89389E+08,s,0.00000E+00,1/s,
1.89389E+08,s,0.00000E+00,1/s,
2.20925E+08,s,0.00000E+00,1/s,
2.20925E+08,s,0.00000E+00,1/s,
2.52461E+08,s,0.00000E+00,1/s,
2.52461E+08,s,0.00000E+00,1/s,
2.84083E+08,s,0.00000E+00,1/s,
2.84083E+08,s,0.00000E+00,1/s,
3.15619E+08,s,0.00000E+00,1/s,
3.15619E+08,s,0.00000E+00,1/s,
3.47155E+08,s,0.00000E+00,1/s,
3.47155E+08,s,2.66996E-12,1/s,
3.78691E+08,s,2.66996E-12,1/s,
3.78691E+08,s,0.00000E+00,1/s,
3.78691E+08,s,0.00000E+00,1/s,
```

The STOMP template is read by default in VADER, but it can be turned off by the following keyword entry: “FILES STOMP_TEMPLATE NOCOMPUTE” (see Section 5.4.7). This option is useful for VADER development efforts in which it is not necessary to process a STOMP template.

5.5.2 VADER Output Files

VADER calculates analyte mass/activity flux from a given release site with respect to time (influx from upper boundary). This calculation is expressed as a time series of releases and remaining inventory for each realization of model coefficients at each site for each analyte. Overall results are stored in a general VADER output table file (vader.table) in a spreadsheet-compatible format. Optionally, VADER will also generate a STOMP keyword input file and/or a vader.river file for input to GWDROP. In addition, vader.done or vader.fail files, a vader.log file, and an optional vader.debug file will be generated.

5.5.2.1 Releases and Remaining Inventory to vader.table File

VADER records analyte releases and mass balance in an output text file in a form shown in Table 5.18. This example file contains excerpted records from a file showing Sr-90 inventory, remediation, release, and quantity remaining for several waste forms at a hypothetical site, Memfis. This file starts with four header records and is followed by waste form information. Volumes are in m^3 and quantities are in Curies.

Table 5.18 Excerpted Records from a vader.table File

```
Title: "Remediating Pyramids for the Environmental Dynasty"
Site: "Memfis"
Analyte: "Sr90"
Realization: "01"
```

```
Waste: "CMNT"
Coeffs: " Ac=1000 Vc=10000 Dc=2.42 HL=28.78 "
 107
 Year  Qw(cm) QTY_IN  RMD_IN  RMD_OUT Release    RelDec     NetRel     CumRel   TotDecay    CumQTY
 1944  1.234   0.0000   0.0000   0.0000      0.0000      0.0000      0.0000    0.0000    0.0000        0.0000
 1945  1.234   0.0000   0.0000   0.0000      0.0000      0.0000      0.0000    0.0000    0.0000        0.0000
. . . .
 1954  1.234   123.00   0.0000   0.0000      12.530      0.29818     12.232    12.232    2.6288       107.84
 1955  1.234   45.000   0.0000   0.0000      13.865      0.32995     13.535    25.476    3.3071       135.67
. . . .
 2050  3.702   0.0000   0.0000   0.0000      0.0000      0.00000     0.0000    16.351    0.0000       0.0000
Wastes: "SOIL"
Coeffs: " A=1500 h=50 B=2.12 Kd=0.023 Cs=1.13e-04 Tw=0.55 HL=28.78 "
107
Year  Qw(cm) QTY_IN   RMD_IN  RMD_OUT Release    RelDec      NetRel      CumRel  TotDecay  CumQTY
 1944  1.234   0.0000   0.0000   0.0000      0.0000      0.0000      0.0000      0.0000   0.0000     0.0000
 1945  1.234   0.0000   0.0000   0.0000      0.0000      0.0000      0.0000      0.0000   0.0000     0.0000
. . . .
 1954  1.234   0.0000   0.0000   0.0000      0.0000      0.0000      0.0000      0.0000   0.0000     0.0000
 1955  1.234   10.000   0.0000   0.0000      0.27434   0.6528E-02  0.26782     0.2678   0.23144     9.4942
. . . .
 2050  3.702   0.0000   0.0000   0.0000  0.3489E-02   0.8303E-04  0.3406E-02 8.8086  0.92577E-03 0.3798E-01
Wastes: "LIQUID"
Coeffs: " HL=28.78 F=1.00 "
107
Year  Qw(cm)  QTY_IN  RMD_IN  RMD_OUT Release  RelDec     NetRel    CumRel     TotDecay    CumQTY
 1944  1.234   0.0000   0.0000   0.0000      0.0000   0.0000     0.0000   0.0000       0.0000       0.0000
 1945  1.234   0.0000   0.0000   0.0000      0.0000   0.0000     0.0000   0.0000       0.0000       0.0000
. . . .
 1954  1.234   0.0000   0.0000   0.0000      0.0000   0.0000     0.0000   0.0000       0.0000       0.0000
 1955  1.234   31.600   0.0000   0.0000      31.600   0.0000     31.600   30.848       0.0000       0.0000
. . . .
 2050  3.702   0.0000  0.0000   0.0000       0.0000   0.0000    0.0000   0.1103E+06  0.0000       0.0000
Wastes: "Aggregate Releases from Solid Waste Forms"
Coeffs: "All solids models pooled"
 107
 Year  Qw(cm)  QTY_IN RMD_IN  RMD_OUT Release    DecRel    NetRel    CumRel   TotDecay   CumQTY
 1944  1.234    0.0000  0.0000   0.0000      0.0000      0.0000     0.0000    0.0000     0.0000       0.0000
 1945  1.234    0.0000  0.0000   0.0000      0.0000      0.0000     0.0000    0.0000     0.0000       0.0000
. . . .
1954  1.234     123.00  0.0000   0.0000      12.530      0.29818   12.232    12.232     2.6288       107.84
 1955  1.234    60.567  0.0000   0.0000      14.140      0.33648   13.803    25.744     3.6711        150.60
. . . .
2050  3.702     0.0000  0.0000   0.0000   0.416E-02   0.990E-04  0.406E-02 25.302    0.5836E-02   0.2394
```

5.5.2.2 VADER River File

This optional file represents direct outflow of liquid contaminants into the river. It is generated when the VADER keyword file contains a RIVER release model specification (see Section 5.4.9). If there is no RIVER waste form in the Inventory file, the file is still written showing zero releases. The default name for this file is vader.river.

Releases to the river are modeled in VADER as an instantaneous pass-through effective on the date specified in the inventory file. Therefore, river releases for radioactive analytes are not decayed. Because the volumes and quantities are taken directly from the input INVENTORY file and passed to GWDROP, no error-checking is performed. Volume units are m^3, and quantity units are Ci for radioactive analytes and kg for nonradioactive analytes. Table 5.19 provides excerpted records from a vader.river file.

Table 5.19 Excerpted Records from a vader.river File

```
FILE_NAME:   vader.river
Run_Date:    03.13.2002    16:45:02
RUN_TITLE:   VADER template keyword file for the 1908-NE site
SITE:        1908-NE
ANALYTE:     Sr90    m^3   Ci
Realization: 1
WasteForm:   RIVER
NREC:         57   Volume   Release
   1944  0.000000E+00  0.000000E+00
. . .
   1971  0.000000E+00  0.000000E+00
   1972  0.499000E+09  0.106000E+01
   1973  0.000000E+00  0.000000E+00
   1974  0.536000E+09  0.326000E+00
   1975  0.507000E+09  0.337000E+00
   1976  0.511000E+09  0.414000E+00
   1977  0.475000E+09  0.198000E+01
   1978  0.485000E+09  0.131000E+01
   1979  0.487000E+09  0.171000E+01
   1980  0.483000E+09  0.218000E+01
   1981  0.530000E+09  0.175000E+01
   1982  0.505000E+09  0.344000E+01
   1983  0.510000E+09  0.548000E+01
   1984  0.466000E+09  0.613000E+01
   1985  0.525000E+09  0.836000E+01
   1986  0.516000E+09  0.947000E+01
   1987  0.481000E+09  0.231000E+01
   1988  0.106000E+09  0.289000E+01
   1989  0.244000E+07  0.260000E+01
   1990  0.220000E+07  0.637000E+00
   1991  0.222000E+07  0.106000E+00
   1992  0.233000E+07  0.959000E-01
   1993  0.261000E+07  0.110000E+00
   1994  0.199000E+07  0.764000E-01
   1995  0.222000E+07  0.248000E+00
   1996  0.225000E+07  0.142000E+00
   1997  0.242000E+07  0.162000E+00
   1998  0.246000E+07  0.221000E+00
   1999  0.248000E+07  0.122000E+00
   2000  0.000000E+00  0.000000E+00
```

5.5.2.3 Releases to the Vadose Zone (STOMP)

VADER writes aggregate solid and liquid flux quantities and liquid volumes of releases for a given site, analyte, and realization into a STOMP template file, a file formatted for input as boundary conditions to STOMP. Times are written in number of elapsed seconds referenced to 00:00:00 January 1, 1944 (unless changed by project management). Fluxes are written in mass (kg) or activity (Ci) units per second, and volumes are written in m^3/sec, to accommodate STOMP requirements. VADER overwrites the aggregate release from all release model sources in correct position in the template without disturbing the remaining parts of the template file, merely adding quantities to flux values already existing in the STOMP input file. VADER processes only one STOMP input file template per run. Initially the default STOMP file name is *input-esp*. VADER first copies this file to a new file called *input*. After *input* is overwritten with the source term records, VADER closes it. Table 5.17 provides an example for this file.

5.5.2.4 vader.log File

VADER echoes principal control data such as run date, analysis title, realization, site, analyte, release models, and coefficient values to a log file. VADER will record unusual events and error messages in this file. Table 5.20 provides an example VADER.log file.

Table 5.20 Excerpted Records from a **vader**.log File

```
         V     V     A     DDDOD    EEEEEEE  RRRRRR
         V     V   A A    D    DD  E        R     R
         V     V   A   A   D     D  E        R     R
         V     V  A     A  D     D  EEEEE    RRRRRR
          V   V   AAAAAAA  D     D  E        R  R
           V V    A     A  D    DD  E        R   R
            V     A     A  DDDDD    EEEEEEE  R    R

                             VADER        Version 0.21
                          Last Modified on 15 Feb 2002
                 ----------------------------------------------
                    Vadose Zone Environmental Release Module
                 ----------------------------------------------
                    Developed By Battelle Memorial Institute
                     Pacific Northwest National Laboratories
                              Richland, Washington

         Current Run ID = 20020313164501   User Name = SAC GW/VZ

            System Date = 03-13-2002   System Time = 16:45:01.954

           The software used to generate this output is experimental
              and has not been formally tested or peer reviewed.

                             Review Signatures

Input Prepared By: ______________________________       Date: _______________

Input Reviewed By: ______________________________       Date: _______________
1MAIN-VADER VADER Keyword File is ./vader.key
```

```
1ECHO_CHECK OF VADER INPUT KEYWORD FILE SPECIFICATIONS
 -----------------------------------------------------

 TITLE   VADER template keyword file for the 1908-NE site

 AGGREGAT  SITE=1908-NE          T  Outfall

 ANALYTE  T ANALYTE=Sr90    Units=Ci  Type=NR  HalfLife=0.287800E+02  Yr

 BALANCE  1995 2000 2050 2100 2200 2300 2400 2500
          2600 2800 3000 3050

 CUTOFF  CutOff Analyte below   0.100000E-12
         CutOff Volume  below   0.100000E-12

 DELTAT     1.000

 DEBUG

  Unit Type Open File-name
    1  A  R  ./vader.key
    2  A  R  ./inv.dat
    3  A  R  ./remediate
    7  A  R  ./sac_esd
    8  A  C  ./input-esp
   10  A  C  ./vader.table
    9  A  C  ./vader.log
   11  A  C  ./vader.done

 MODELS   Number Models =  1
 WasteName Model Compute Collapse
 RIVER      T T  1944  3050
                           F = 0.10000E+01 unitless

 PATHNAME  PATHNAME=./

 PERIOD    Start Year=    1944
           Closure Year=  2050
           Stop Year=     3050

 REALIZAT  Realization Number is 1    Max Realization Number is   2

 RECHARGE Number RECHARGE lines is   1.  Units are  cm/yr
 Start   End   Qw(cm)
  1944  3050  0.1000

 REMEDIAT  Number Remediation Records is    0

 EXIT VADER_KEY_SHOW

 ELAPSE Finishing preliminaries

    Date: 03-13-2002
    Time: 16:45:02.024
 ENTERING Read_Inv
 --------------------------------------------------------
 DISPLAY_INV
 INVENTORY REPORT
 (w/o reference to Remediations)
1XRealization No., Number of release sites
 Analyte is Sr90
 Site is    1908-NE
 Realization ID            1
 Number Waste Forms Present            1 (Non-zero Quantities)
 Waste Form   RIVER
 Rel. Model   RIVER
  1972 m^3    0.49900E+09
  1972 Ci      1.0600

  1974 m^3    0.53600E+09
  1974 Ci     0.32600

 . . .

  1999 m^3    0.24800E+07
  1999 Ci     0.12200

 DISPLAY_INV  End Inventory Listing
 --------------------------------------------------------
 All Analyte Data is in Keyword File
 Do not access SAC-ESD file

 ELAPSE Begin the Calculations Loop

    Date: 03-13-2002
    Time: 16:45:02.238
 ---------------------------------------------------------
 Entering RIVER RIVER
 Processing Release Model RIVER   at index i= 1  Comment:

  Coeff   Value       Units
 Start Release during  1944
 Stop  Release during  3050
  F     0.10000E+01    unitless
 HL     0.28780E+02    yr
 EXIT SUBROUTINE RIVER Ierr=           0
 No Outgoing Remediations Specified in Keyword File
 No REMEDIATIONS file written to working directory
 STOMP Template File INPUT CREATED
 STOMP input template pathnames:./input-esp
 STOMP final template pathnames:./input
 Write Vader.done file to ./vader.done
                            11           0
 NOTICE: All Waste Forms in INVENTORY are associated with a Release Model
 TOTAL CPU TIME:    2416500.

 ELAPSE  Finish VADER Run

    Date: 03-13-2002
    Time: 16:45:23.480
  Successful VADER run
```

5.5.2.5 vader.debug File

Intermediate results and error messages associated with errors encountered during VADER execution are written to the vader.debug file. This includes messages associated with the DEBUG keyword commands. Information in this file is intended to be self-documenting. Table 5.21 provides an example vader.debug file.

Table 5.21 Example vader.debug File

```
  PROGRAM VADER 0.21     version date:15 Feb 2002

 ELAPSE Initiate a VADER Run

    Date: 03-13-2002
    Time: 16:45:01.956
 PROCESS_KEYWORDS
 Attempt to open default keyword file ./vader.key
  Opened VADER template file ./vader.key
 ENTER Fetch_Analyte_Units with analyte Sr90   in    units
  Fetch_Analyte_Units Default Units for Sr90   are Ci
 CONVERSION Qw mm/yr cm/yr  0.1000E+01  0.1000E+00
1ECHO_CHECK OF VADER INPUT KEYWORD FILE SPECIFICATIONS
 -----------------------------------------------------

 TITLE   VADER template keyword file for the 1908-NE site

 AGGREGAT  SITE=1908-NE          T  Outfall

 ANALYTE  T ANALYTE=Sr90    Units=Ci  Type=NR  HalfLife=0.287800E+02  Yr

 BALANCE  1995 2000 2050 2100 2200 2300 2400 2500
          2600 2800 3000 3050

 CUTOFF  CutOff Analyte below   0.100000E-12
         CutOff Volume  below   0.100000E-12

 DELTAT     1.000

 DEBUG



  Unit Type Open File-name
    1  A  R  ./vader.key
    2  A  R  inv.dat
    3  A  R  remediate
    7  A  R  sac_esd
    8  A  C  input-esp
   10  A  C  vader.table
    9  A  C  vader.log
   11  A  C  vader.done

 MODELS   Number Models =  1
 WasteName Model Compute Collapse
 RIVER      T T  1944  3050
```

```
                           F = 0.10000E+01 unitless


 PATHNAME  PATHNAME=./

 PERIOD    Start Year=    1944
           Closure Year=  2050
           Stop Year=     3050

 REALIZAT  Realization Number is 1    Max Realization Number is   2

 RECHARGE Number RECHARGE lines is   1.  Units are  cm/yr
 Start   End  Qw(cm)
  1944  3050  0.1000

 REMEDIAT  Number Remediation Records is    0

 EXIT VADER_KEY_SHOW
 Enter Check_For_Overlap
 Exit Check_For_Overlap
 BUILD_PATH_NAME ./

 BUILD_PATH_NAME: Building Pathnames to Vader Input and Output Files
 PATHNAME         ./
 SAC_ESD FILE     ./sac_esd
 INVENTORY FILE   ./inv.dat
 STOMP TEMPLATE   ./input-esp
 TABLE FILE       ./vader.table
 RIVER FILE       ./vader.river
 REMEDIATIONS-OUT ./vader.rem
 LOG FILE         ./vader.log
 RUNNING FILE     ./vader.running
 DONE FILE        ./vader.done
 Remediations-in files are declared in another section
 OPENER: File ./inv.dat Opened OK
 OPENER: File ./vader.table Opened OK
 OPENER: File ./vader.log Opened OK
 ENTERING Read_Inv
 READ_INV: read inventory file from unit            2

Error number    1 encountered in routine DISPLAY_INV
Message: WARNING: Inventory Volume Units Not Consistent!!

Error number    3 encountered in routine DISPLAY_INV
Message: WARNING: Inventory Volume Units Not Consistent!!

Error number    3 encountered in routine MAIN-VADER
Message: Inventory Display Error

 ENTERING INITIALIZER
 m =            1    Collapse= T
 Input argument     Model_Name = RIVER
 Release Model      Model=       RIVER
 Waste Form      WasteForm=       RIVER
 ENTER STEPPER  with RIVER     m=           1
 ####            1 RIVER               1           1
```

```
OPENER: File ./vader.river Opened OK
OPENER: File ./input-esp Opened OK
OPENER: File ./input Opened OK
Ierr, InvEnd, IremErr, ModIerr,iVZRSTPerr    0    0   0
          0           0
OPENER: File ./vader.done Opened OK
TOTAL CPU TIME:    2416500.

ELAPSE  Finish VADER Run

   Date: 03-13-2002
   Time: 16:45:23.480
 Successful VADER run
```

5.5.2.6 vader.done file

The vader.done file is built when VADER runs to a successful completion, meaning that all input files were read and all calculations were completed. The primary purpose for this file is to signal the SAC ESP that VADER has finished and the STOMP source term is ready for processing. The vader.done file also signals SAC ESP whether there is a vader.river file with non-zero releases to send to GWDROP. The file supplies summary information about the materials released; lists the waste forms, including variants, found in the inventory; and lists which release models were associated with each waste form variant. If some inventory items were not mapped to a release mode, this is indicated as well, because it indicates that the complete inventory was not addressed by this VADER run. This information is more comprehensive in the vader.table file. Clock time finished and CPU time to execute are also provided. Table 5.22 provides an example vader.done file.

Table 5.22 Example vader.done File

```
VADER.done c:\sac\code\mess\vadose\memfis\Sr90\03\vader.done
 Memfis
 Sr90
 3
 Number Models= 6        CumRel(StopYr) MaxAnnualRel
CMNT           1     1     2  34.079      34.807
SOIL           1     1    36  176.37      62.546
CAKE           1     1     3  0.12286     0.31409E-01
GLASS          1     1     6  0.0000      9.0458
REACTOR        1     1     4  0.0000      27.226
RIVER          1     1     2  0.0000      3821.7

Waste Name     Release Model
CMNTXXX        CMNT
LIQUID         LIQUID
GLASS          GLASS
CEMENT         CMNT
CAKE           CAKE
SOIL           SOIL
SOILX          SOIL
SOILY          SOIL
RIVER          RIVER
```

```
Notice:  All Waste Forms Successfully mapped to a release model

 ELAPSE  Finish VADER Run

 Date: 01-04-2002
 Time: 13:11:25.170
 TOTAL CPU TIME:  0.437776357
 MAIN-VADER Successful VADER run
 Error Code Ierr =  0
```

5.6 Consistency and Traceability Checks

The release models, combined with radioactive decay, express releases as uncoupled first-order differential equations describing exponential growth and especially decay (the soil-barrier model having two coupled coefficients, is an exception; see Appendix D of Kincaid et al. (1998). Calculations are performed using numerical methods and are by definition approximate – solution accuracy depends on the numerical method and the time step. Calculated release time series are expected to be smooth curves, with occasional jumps when inventory pulses kick in and when quantities are removed for remedial actions. This provides a basis for inspection of the output to verify correct calculations and data processing. Some possible verifications are the following:

- For a given set of VADER runs, the model coefficients Q_w, K_d, C_s, C_o, L_e, etc. can be extracted from either the specific keyword files provided by SAC ESP, the echo-check data written into the vader.log file, or the vader.table file for each VADER run. The values actually used in each computation are written to the log file.
- The time series (quantity released, cumulative quantity released, remaining cumulative inventory, etc.) for each waste form is written to the vader.table file. Plots of the quantities over time should reveal relatively stable patterns of growth and decay consistent with inventory, half-life, and release model coefficients with a time-stepping solution of one year.
- In VADER, mass-balance (or activity-balance) calculations mean that original inventories can be obtained by back-calculating from releases, accounting for radioactive decay and remediation actions. The vader.table file provides complete information for performing mass-balance calculations.

6.0 VZDROP – Vadose Zone to Groundwater Mass Transfer Module

6.1 Code Purpose

VZDROP (Vadose Zone Data Restructure for Other Programs) is a data transfer utility code that aggregates analyte mass or activity releases produced by the SAC vadose zone flow and transport module (STOMP code) and translates these into analyte mass inputs in the SAC groundwater transport module (CFEST code).

VZDROP may be run either within the SAC framework under the control of the SAC environmental stochastic processor (ESP), or independently to translate STOMP releases into CFEST inputs provided the 1) the SAC modification set has been used in the STOMP code, and 2) the ~SAC Release card has been properly used in the STOMP input files for all STOMP simulations. Independent use of VZDROP has been valuable for groundwater model sensitivity studies.

In addition, a FLUID mode exists that, when invoked, permits VZDROP to serve as a data transfer utility that aggregates liquid discharges to ground (as expressed in STOMP input file ~source cards) and translates these into liquid sources in the groundwater flow module (CFEST code). This mode is not used within the SAC framework but was developed to meet a specialized need to use SAC inventory data on artificial liquid discharges to automatically build the nodal sources and sinks for a CFEST L3I file. Currently, this mode is restricted to representing vadose zone liquid influxes as groundwater influxes.

6.2 Algorithms and Assumptions

This section provides a brief description of algorithms and major assumptions for VZDROP.

6.2.1 Algorithms

6.2.1.1 Nearest CFEST Node Determination for Vadose Zone Sites

Each vadose zone site is associated with two pairs of coordinates that define the northwest and southeast corners of a rectangular area, aligned to cardinal directions that constitutes the vadose zone site areal extent. These are defined in the SAC header information written at the beginning of STOMP input and output files when STOMP is used within the SAC framework. Table 6.1 provides an example header.

Table 6.1 Example SAC Header for STOMP Input/Output Files

```
#SAC STOMP input created by ESP   05/29/2001 - 12:09:56
#SAC Case ID          : SAC Rev. 0 Initial Assessment
#SAC Template         : stomp\116-B-3\template.key
#SAC Site ID          : 116-B-3
#SAC Site NW Easting  :   565353.6
#SAC Site NW Northing :   144529.9
```

```
#SAC Site SE Easting  :   565358.4
#SAC Site SE Northing :   144525.1
#SAC Analyte          : I129
#SAC Realization      :     3
```

VZDROP extracts the location information from these file headers and computes a center location for each vadose zone site:

$$x_C = MIN(x_{NW}, x_{SE}) + \frac{1}{2}\left|x_{NW} - x_{SE}\right|$$

$$\mathrm{y_C = MIN(y_{NW}, y_{SE}) + \frac{1}{2}\left|y_{NW} - y_{SE}\right|}$$

where (x_c,y_c) are the coordinates at the center of the vadose zone region and the coordinates of the northwest and southeast corners of the vadose zone site are given by (x_{NW},y_{NW}) and (x_{SE},y_{SE}), respectively.

VZDROP then computes the distance from these center coordinates, (x_c,y_c), to every CFEST grid node defined in the CFEST LP1 file:

$$d_n = \sqrt{(x_n - x_c)^2 + (y_n - y_c)^2}$$

where d_n is the distance from the center of the vadose zone location to CFEST grid node *n* with coordinates (x_n,y_n). The node with the minimum distance d_n is selected as the nearest CFEST node; this becomes the node to which all releases for this vadose zone site will be assigned by VZDROP.

To prevent VZDROP from assigning releases to certain nodes that would cause severe computational problems, special treatment is used in VZDROP. Boundary nodes assigned constant head (Dirichlet) conditions should not be specified as nodal sources in CFEST. Also, sink nodes (pumping wells) are problematic, as the mass that would be specified as a source by VZDROP would appear to be the extraction of that quantity of analyte mass in CFEST. To prevent assigning releases to Dirichlet condition nodes (e.g., on the Columbia River boundary) or sink nodes, all such nodes are reassigned to coordinates (0,0) before the nearest node calculations discussed above. This step ensures that these nodes are well off the Hanford Site and can never be chosen as the nearest node to any vadose zone release site. The assignment to (0,0) for these nodes is only for VZDROP purposes; the grid node coordinates recorded in CFEST input files are not changed by VZDROP.

In future development, it is expected that VZDROP will be enhanced to allow for geographic distribution of vadose zone releases to multiple CFEST nodes based on areal intersections of the vadose zone release area and CFEST grid finite elements. However, such enhancement has not been completed in SAC Rev. 0.

6.2.1.2 Distribution of Analyte Releases from Vadose Zone Sites to Groundwater

The chief purpose of VZDROP is to aggregate vadose zone releases over both space and time into nodal sources for the CFEST code. The algorithm for doing this works under the convention that CFEST time steps for nodal sources are fixed, and STOMP releases are divided in time to match the times steps for which CFEST nodal sources apply.

VZDROP begins this task by comparing the list of nearest CFEST nodes identified earlier as corresponding to vadose zone site locations to the list of nodes already declared in the unmodified CFEST L3I file. A revised list is prepared that includes all nodes already declared plus any additional nodes necessary to accommodate the vadose zone release sites. As a precaution, VZDROP will ensure that any node declared as a source in any CFEST time step is declared in every CFEST time step in the VZDROP-modified CFEST L3I file.

To aggregate in time, each CFEST time step for which a nodal source endures has starting time t_{C0} and ending time t_{C1} (note that for the subsequent time step, t_{CF0} will equal t_{C1} of the previous time step). In CFEST, these times are expressed in days, but for VZDROP calculations, they are converted to seconds elapsed since the baseline for all SAC codes, midnight preceding January 1, 1944. Similarly, we consider every analyte release computed by the STOMP code to occur over time steps, each of which we can denote as beginning at t_{S0} and ending at t_{S1}. VZDROP considers each CFEST nodal source time step, assigning times t_{C0} and t_{C1} for the step, and then evaluating every CFEST node to determine if it is on the list of nodes that will receive STOMP-predicted analyte releases. For those nodes that will, VZDROP examines each STOMP release site to determine if it releases to the CFEST node being evaluated. For each STOMP release site that does, VZDROP then evaluates every time step of the analyte release predicted by STOMP, assigning times t_{S0} and t_{S1}, to determine if there is any overlap with the time range specified by t_{C0} and t_{C1}. When overlap is found, the length of time of the overlap is determined:

$$\Delta \boldsymbol{t} = \mathrm{MIN}(t_{C1}, t_{S1}) - \mathrm{MAX}(t_{C0}, t_{S0})$$

The duration is used to express the fraction of the STOMP time step *n* that is within the current CFEST time step:

$$f_{duration,n} = \frac{\Delta t}{t_{S1} - t_{S0}}$$

This fraction, in turn, is used to assign the portion of the STOMP release for the current time step to the current CFEST time step at the current CFEST node:

$$Q_C = \sum_{n=1,m} f_{\mathrm{duration,n}} q_{S,n}$$

where quantity is an accumulation of all STOMP releases to the current CFEST node for the current CFEST nodal source time range.

The accumulation is over the product of the fraction of each STOMP release time range (from 1 to *n* release time steps) within the current CFEST time step ($f_{duration,n}$) and the quantity released within each STOMP release time range ($q_{S,n}$) for each STOMP vadose zone site for which the current CFEST node is the nearest node to the release site. When this accumulation is completed over all STOMP release sites and release time ranges, the total mass source for each CFEST node Q_c is computed. If the analyte is radioactive (as indicated by the user input in the VZDROP keyword control file), the array of sources is then converted from activity (curies) to mass (kilograms [kg]) using the specific activity provided in the VZDROP keyword control file.

Before storing the mass source, it is converted to a concentration source if the liquid source already present in the CFEST L3I file for the current time step and node is nonzero. To do this, the liquid source rate already specified in the CFEST L3I file is multiplied by the CFEST time step duration to determine the volumetric liquid quantity of the source for the current time step. The analyte mass (kg) is divided by the volumetric liquid quantity (m^3) to obtain a concentration (in kg/m^3). This value is further multiplied by the inverse of the CCCMAS variable given in the CFEST LP1 file to convert it to the concentration units CFEST will expect. For nodes with no liquid sources, VZDROP applies the mass as a dry mass injection, expressed in kilograms per day with the appropriate CFEST indicator value in the fluid source quantity (-1.0E-30) to mark the source as a dry mass injection node. All values for the nodal source terms are computed and stored in an array to be used in overwriting the CFEST L3I file with updated nodal source values at the conclusion of VZDROP execution.

6.2.1.3 A Note About Time Drift Effect

The algorithm used to distribute STOMP releases in time to CFEST time steps can have a time drift effect that may result in backing up releases in time. To illustrate this effect, consider a CFEST problem with six-month time steps, beginning January 1, and a STOMP problem with six-month time steps that begin April 1 (i.e., the time steps are the same size but out of phase). We depict the time steps in Figure 6-1 to help to visualize the backing up effect. For this case, the algorithm would assign all of the release during STOMP time step 1 to CFEST time step 1. For STOMP time step 2, however, half of the time step (3 months) occurs before CFEST time step 1 ends, and the remainder (3 months) occurs in the CFEST time step 2. So VZDROP would assign 50 percent of the release from STOMP time step 2 to CFEST time step 1 and the other 50 percent to CFEST time step 2. Now assume that the releases predicted by STOMP are as follows:

- STOMP time step 1, no release
- STOMP time step 2, 2.0 kilograms
- All subsequent STOMP time steps, no release.

VZDROP would assign 1.0 kilogram to CFEST time step 1 and 1.0 kilogram to CFEST time step 2. Here is where the SAC framework can appear to back up a release in time; because the source is uniform within a single time step, CFEST would appear to be receiving analyte mass at the rate of 1.0 kilogram per six months beginning on January 1, while STOMP reported the mass did not release from the vadose zone until April 1 of the same year. Hence, on the face of the matter, it would appear as if the groundwater system received the vadose zone release three months before the vadose zone system released it.

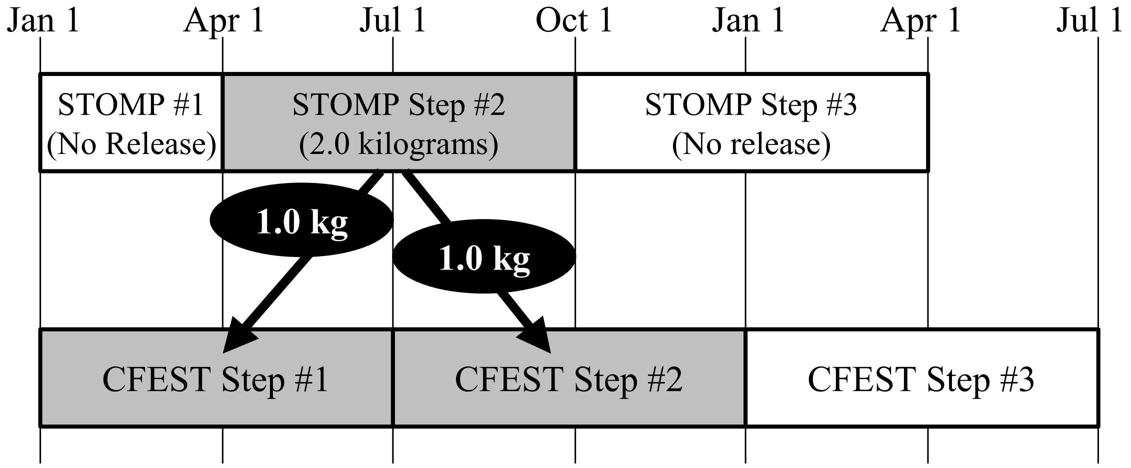


Figure 6-1 Depiction of Hypothetical CFEST and STOMP to Describe the Backing Up Effect

The backing up effect is not an error but rather reflects the temporal resolution of the SAC framework that is limited to the magnitude of the larger time steps of the two codes involved in this translation. Other design choices could have forced the releases forward in time to complete CFEST time steps subsequent to STOMP release time steps, but it was not desirable to artificially delay releases in this application. Less noticeable, the same effect can also result in nonzero source terms in CFEST later than the last nonzero STOMP release time step. Care should be taken not to over-interpret apparent discrepancies in analyte arrival time that are less than the magnitude of the largest time step in use between the STOMP and CFEST simulations.

6.2.2 Major Assumptions

The following major assumptions are made:

- Both CFEST and STOMP simulations begin at midnight preceding January 1, 1944
- CFEST time steps are expressed in days in the CFEST input files
- All of the release from a given vadose zone release site is specified as an analyte influx to the CFEST grid node nearest the center of the vadose zone site release area, without regard to the number of CFEST nodes that may coincide with the vadose zone site release area.

6.3 Code Environment

VZDROP was primarily developed as an internal program to be invoked by the master SAC control software, SAC ESP. However, VZDROP has proven to be a versatile tool that can be invoked independent of SAC ESP for use in case studies and other purposes.

6.3.1 Location in Processing Sequence

In the normal order of processing, VZDROP execution takes place between STOMP and CFEST as shown in Figure 1-2. Within the SAC framework, VZDROP can only be invoked for a given analyte and a given realization after successful simulation of all vadose zone sites by STOMP for that analyte and realization. This is necessary because the SAC groundwater transport module aggregates releases from all vadose zone sites; any uncompleted vadose zone site simulations would represent missing analyte mass in the groundwater transport module simulations.

When executed independently, VZDROP can be used to translate vadose zone releases from any number of STOMP release files specified by the user in the VZDROP keyword control file into analyte mass nodal sources for a CFEST L3I file.

6.3.2 How Code Is Invoked

VZDROP may be used in either Microsoft Windows or Linux environments, provided the code has been compiled by an appropriate Fortran 95 compiler for the environment. In the Microsoft Windows environment, VZDROP is executed by either of the following DOS commands:

```
VZDROP
VZDROP vzdrop_case1.key
```

In the first example, VZDROP is invoked and expected to use the default VZDROP keyword control file (vzdrop.key), which is located in the current directory. In the second example, VZDROP is invoked, and the VZDROP keyword control file is explicitly named vzdrop_case1.key.

In the Linux environment, VZDROP is executed through any of the following Bourne Shell or C Shell commands:

```
vzdrop.exe
vzdrop.exe vzdrop_case1.key
```

In the first example, VZDROP is invoked and expected to use the default VZDROP keyword control file (vzdrop.key), which is located in the current directory. In the second example, VZDROP is invoked, and the VZDROP keyword control file is explicitly named vzdrop_case1.key.

6.3.3 Code Control and Keyword Descriptions

A single keyword control file controls the VZDROP code. This ASCII text file identifies the names and locations of STOMP release files containing the record of analyte activity or mass release from vadose zone to groundwater, the CFEST files needed for reference and to be modified so they include analyte source information, the specific activity of the analyte, and some consistency check variables (SAC analyte, realization).

6.3.3.1 Keyword Control File Format

The keyword control file for VZDROP is prepared by the SAC ESP when VZDROP is used within the SAC framework. However, a user may also prepare the keyword control file directly. In either case, the input file format is the same. The format of the file is as follows:

- **First Line:** The literal string “~VZDROP Key Input” or “~VZDROP Key Input FLUID”
- **Second Line:** Name of the CFEST .LP1 input file
- **Third line:** Name of the CFEST .L3I input file
- **Fourth line:** Analyte Type (the string “Radioactive” or “Chemical”), Specific Activity (real number), Specific Activity Units (string such as “Ci/g”), SAC Analyte ID (6 character), SAC Realization number (integer)
- **Fifth line:** Number of STOMP release (or input) files (integer)
- **Succeeding lines:** Each line contains the name of a STOMP ‘release’ (or ‘input’) name.

The modifier FLUID added to the first line of the keyword control file directs VZDROP to impose the aqueous volumetric sources in specified STOMP input files as the liquid sources in the CFEST L3I file, instead of processing analyte mass. If the FLUID modifier is present, then instead of a list of STOMP release files, a list of STOMP input files must be provided. No other changes are necessary to invoke this option.

The SAC analyte ID and SAC realization number are specified for consistency checking purposes: VZDROP will examine the SAC header of each specified STOMP release file and ensure that the SAC analyte ID and SAC realization number there match the values specified in this keyword control file.

6.3.3.2 Example Keyword Control Files

The first example of a VZDROP keyword file provided in Table 6.2 illustrates how to direct VZDROP to translate a single STOMP release file into an analyte source for a CFEST input file. Full paths are specified for a Microsoft Windows environment; hence, this control file would be suitable for execution from any location, provided the local drive mappings are correct.

Table 6.2 VZDROP Keyword Control File Example 1

```
~VZDROP Key Input
e:\integ_01\cfest\u238\han-input.lp1
e:\ integ_01\cfest\u238\han-input.l3i
Radioactive,9.7245E+3,Ci/g,H3,10,
1
e:\integ_01\stomp\SXTank\U238\1\release
```

When executed using this control file, VZDROP would read analyte release information found in the file e:\integ_01\stomp\SXTank\U238\1\release, convert the release from activity expressed in curies (Ci) to mass expressed in kilograms using the specific activity of 9,724.5 curies per gram (the value for tritium in this case), and revise the CFEST file e:\ integ_01\cfest\u238\han-input.l3i to incorporate vadose zone releases as an analyte mass nodal source term at the CFEST node nearest the location of the STOMP release (as given by header information in the STOMP release file).

Processing multiple vadose zone releases is illustrated in the second example of a VZDROP keyword file provided in Table 6.3. Ten vadose zone sites have been identified as principal tritium sources on the Hanford Central Plateau, and it is desired to translate the vadose zone releases from only these 10 sites into analyte sources in a CFEST input file for a case study. In this example, the CFEST files are in the local directory (so no path is specified), but a full path specification is given for each STOMP release file as the desired files are located in a separate directory structure.

Table 6.3 VZDROP Keyword Control File Example 2

```
~VZDROP Key Input
han_input_CA.lp1
han_input_CA.l3i
Radioactive,9.7245E+3,Ci/g,H3,10,
10
L:\Initial\Stomp\216-A-10\H3\10\release
L:\Initial\Stomp\216-A-5\H3\10\release
L:\Initial\Stomp\216-A-21\H3\10\release
L:\Initial\Stomp\216-U-8\H3\10\release
L:\Initial\Stomp\216-S-7\H3\10\release
L:\Initial\Stomp\216-S-21\H3\10\release
L:\Initial\Stomp\216-S-9\H3\10\release
L:\Initial\Stomp\216-S-1#2\H3\10\release
L:\Initial\Stomp\216-S-4\H3\10\release
L:\Initial\Stomp\216-S-3\H3\10\release
```

When the VZDROP code is used within the SAC framework, the SAC ESP code prepares the keyword control file internally. In this case, the file is always named vzdrop.key and always contains as many STOMP release files as there are vadose zone sites identified by the AGGREGATE keywords in the SAC ESD file (see Section 2.1.1). In the SAC Initial Assessment, for example, 719 vadose zone sites were simulated and so the VZDROP keyword control file created by the SAC ESP would always specify 719 STOMP release files.

6.4 Data Files

The VZDROP keyword control file, described in Section 6.3.3, identifies the specific STOMP release files that are inputs to VZDROP, the CFEST LP1 file that is an input for VZDROP, and the CFEST L3I file that is both an input and an output for VZDROP. All of these files are generated either by STOMP or primarily for CFEST, so only brief descriptions are provided here.

6.4.1 Input Files

Both CFEST input files (the LP1 and L3I) file and all STOMP release files specified in the VZDROP keyword control file constitute inputs to the VZDROP code. Note that the CFEST L3I file is both an input and the output of VZDROP. The final product of VZDROP is a modified CFEST L3I file that includes revised nodal source inputs based on STOMP release results.

6.4.1.1 STOMP Release Files

The SAC modification set to the STOMP code includes a subroutine that records releases of analyte activity or mass to groundwater with time in a special release file. This feature is invoked by use of the STOMP ~SAC Release Card in a STOMP input file. In response, STOMP generates a release file with the following format:

- **First line:** The literal string "~STOMP SAC Mass/Activity Releases to Aquifer"
- **Header information:** A variable number of header lines, treated as comments, denoted by '#' in column 1)
- **Succeeding lines:** For a variable number of lines (read until End-Of-File). Each line contains the following information: Time (real), Time Units (character), Release Elevation (real), Release Elevation Units (character), Released Mass or Activity of Analyte (in kilograms or curies) (real).

Table 6.4 provides an example release file (all of the release records except the first three and last three have been omitted for brevity in this presentation). In this example, releases of tritium (H3) are recorded by STOMP for vadose zone site 216-A-8, Realization 5, of the SAC Rev. 0 Initial Assessment.

Table 6.4 Example STOMP Release File

```
~STOMP SAC Mass/Activity Releases to Aquifer
#SAC STOMP input created by ESP   05/29/2001 - 12:09:56
#SAC Case ID          : SAC Rev. 0 Initial Assessment
#SAC Template         : stomp\216-A-8\template.key
#SAC Site ID          : 216-A-8
#SAC Site NW Easting  :   575759.8
#SAC Site NW Northing :   136213.9
#SAC Site SE Easting  :   575799.6
#SAC Site SE Northing :   136174.1
#SAC Analyte          : H3
#SAC Realization      :     5
 1.035390337094E+09,s, 7.620000000000E-02,m, 5.414612006876E-06,
 1.041470000000E+09,s, 7.620000000000E-02,m, 1.561485706124E-06,
 1.067020337094E+09,s, 7.620000000000E-02,m, 1.202660496040E-05,
 ...
1.123433280000E+10,s, 7.620000000000E-02,m, 2.830681928383E-10,
 1.321402325324E+10,s, 7.620000000000E-02,m, 0.000000000000E+00,
 3.502893600000E+10,s, 7.620000000000E-02,m, 0.000000000000E+00,
```

6.4.1.2 CFEST LP1 and L3I files

The LP1 and L3I files are the primary input files for the CFEST code and are discussed extensively in the CFEST Users Guide (CFEST 2000). Presentation of the format of these files is unnecessary for the purposes of describing VZDROP functionality; the user is referred to the CFEST documentation for a complete description of the LP1 and L3I file formats.

The CFEST LP1 file is read to provide VZDROP with grid-specific information, including node identification indices and locations and finite element arrangement. The LP1 file is not modified by VZDROP.

The CFEST L3I file is read to gather information on time steps, nodal fluid sources, and the times at which nodal source terms change. The file is also read to provide VZDROP with the template it uses to create a replacement L3I file with updated nodal mass sources.

6.4.2 Output Files

The output files from VZDROP are a modified CFEST L3I file and a log file. The following sections describe these files.

6.4.2.1 Modified CFEST L3I file

The only numerical output produced by VZDROP is a modified CFEST L3I file containing updated nodal mass sources. No other information in the L3I file is modified by VZDROP. The portion of the CFEST L3I file that is modified is that containing nodal sources. VZDROP makes three primary modifications:

- All nodal sources are rewritten such that if a node is specified as a source in any time step, it appears as a source in every time step (with zero quantity if it did not appear in a given time step in the original L3I file). This modification is necessary to ensure uniform treatment of nodal sources by the VZDROP code.
- New nodes are added to the nodal source list for CFEST nodes receiving STOMP releases that did not appear in the original L3I file nodal source list.
- All mass sources by node and time step are overwritten using the source information from the STOMP release files specified in the VZDROP keyword control file. To assist in interpreting the results, each modified or new source node is commented with a identification tag that includes the name VZDROP, a date stamp, and the SAC vadose zone site ID for each site that contributed releases to that CFEST nodal source.

VZDROP does not preserve nodal mass sources specified in the original CFEST L3I file – these are overwritten with new information based on STOMP releases. This precludes two problems. First, a CFEST template might be provided that already contains nodal mass sources – without overwriting, a VZDROP run would produce a CFEST L3I file that contained both old and new masses in the nodal sources. Second, VZDROP is often invoked repeatedly as changes are made to the vadose zone data. Without forced overwriting, each consecutive execution would "accumulate" mass, making nonsense of the inputs. Therefore, by design, VZDROP overwrites any nodal mass sources in the original CFEST L3I file.

An overly simplistic example is provided to illustrate how VZDROP changes the CFEST L3I file. For this example, the CFEST L3I file has only two time steps and three nodal sources. The relevant portion of the L3I file would appear as

```
2    1    2
  26  0.0000E+00  2.5000E+01  0.0000E+00 ,, TIMESTEP #1,,Jan44-Jun44
  83  0.0000E+00  2.5000E+01  0.0000E+00 ,, 1199-34-15A/B,
 118  0.0000E+00  2.5000E+01  0.0000E+00 ,, North Richland Well Field,
   0
  26  0.0000E+00  2.5000E+01  0.0000E+00 ,, TIMESTEP #2,,Jul44-Dec44
  83  0.0000E+00  2.5000E+01  0.0000E+00 ,, 1199-34-15A/B,
1448 -1.0000E-30  2.5000E+01  0.0000E+00 ,,
   0
```

The first row shown in the example directs CFEST that the nodal sources will change at two time steps, specifically time step numbers 1 and 2. Then nodal sources are specified for three nodes (26, 83, and 118) in the first time step and for three nodes (26, 83, and 1448) in the second time step. The second column contains the fluid source term, the third contains column the fluid temperature, and the fourth column contains the mass source term. Comments follow the fourth column. A node with ID = 0 terminates source input for the time step and begins the next time step.

VZDROP is executed for this L3I file to incorporate the vadose zone releases from a single STOMP simulation at the vadose zone release site identified as 216-U-8. As it happens, 216-U-8 is nearest to CFEST node number 1448. When completed, the same portion of the L3I file will appear as

```
    2    1    2
      26  0.0000E+00  2.5000E+01  0.0000E+00 ,, TIMESTEP #1,,Jan44-Jun44
      83  0.0000E+00  2.5000E+01  0.0000E+00 ,, 1199-34-15A/B,
     118  0.0000E+00  2.5000E+01  0.0000E+00 ,, North Richland Well Field,
    1448 -1.0000E-30  2.5000E+01  0.0000E+00 ,, Source Updated [VZDROP
11/19/2001 16:29:49]216-u-8;
       0
      26  0.0000E+00  2.5000E+01  0.0000E+00 ,, TIMESTEP #2,,Jul44-Dec44
      83  0.0000E+00  2.5000E+01  0.0000E+00 ,, 1199-34-15A/B,
     118  0.0000E+00  2.5000E+01  0.0000E+00 ,, North Richland Well Field,
    1448 -1.0000E-30  2.5000E+01  1.0000E+00 ,, Source Updated [VZDROP
11/19/2001 16:29:49]216-u-8;
       0
```

Three major changes were made. First, VZDROP ensured that all source nodes declared in any time step appeared in all time steps; thus, node 118 was included in time step 2, and node 1448 was added to time step 1. Second, VZDROP imposed a dry mass source (fluid flux = -1.0E-30 indicates dry mass injection) at a rate of 1.0 kg/day to node 1448 for the duration of time step 2. Third, VZDROP added a comment to node 1448 nodal sources in all time steps to show it had updated the source information for this node on November 19, 2001 and that the source included one SAC vadose zone site, 216-U-8.

Practical applications of VZDROP result in considerably more complex CFEST L3I files, but the general treatment is the same as for this illustrative example. A user may typically check for VZDROP modifications by opening the modified L3I file with any text editor and searching on the string VZDROP. Every source node record modified by VZDROP will contain the text string VZDROP in the comment portion of the line.

6.4.2.2 Log File

VZDROP generates a log file to provide the user with a report on any errors encountered and a summary of the analyte mass transfer from the vadose zone flow and transport module to the groundwater flow and transport module. The file is always named vzdrop.log and is written to the current directory from which VZDROP was invoked.

Table 6.5 provides an example log file for a successful run (detailed information on CFEST time steps 3 to 108 are omitted for brevity in this presentation). The log file will identify that the program VZDROP generated this file, what version was used, and when the version was created. Next, date and time stamps are provided demarking when execution began. The number of CFEST source nodes that were given in the original (unmodified) CFEST L3I input file are recorded as well as the number of revised CFEST source nodes. Next, a summary of the total vadose zone released mass arriving in each CFEST nodal source time range is given in tabular format. If the analyte is radioactive, the table will provide both activity and mass per time step, but for nonradioactive materials, only mass will be recorded. After the tabular output by time step, a summary for the entire release is given showing the total of all STOMP releases for all vadose zone sites over all time ranges in contrast with the total mass recorded to the CFEST L3I file for all nodes over all time ranges. The mass balance error is noted. A final date and time stamp will end the file, denoting when execution ended.

Table 6.5 Example VZDROP Log File

```
              V     V  ZZZZZZZ  DDDDD    RRRRRR    OOOOO   PPPPPP
              V     V       Z   G    D   R     R O      O  P     P
               V   V       Z    G     D  R     R  O     O  B     P
               V   V      Z     G     D  RRRRRR   O     O  PPPPPP
                V V      Z      G     D  R   R    O     O  P
                V V     Z       G    D   R    R   O     O  P
                 V     ZZZZZZZ  DDDDD    R     R   OOOOO   P


                         VZDROP    Version 1.7
                         Last Modified on 26 Jun 2001
          ---------------------------------------------------------
            SAC Rev. 0 Vadose Zone Data Restructure Other Programs
          ---------------------------------------------------------
                  Developed by Battelle Memorial Institute
                   Pacific Northwest National Laboratories
                            Richland, Washington

Message originating in routine VZDROP
Message: VZDROP Execution History
         Execution Start : 11/19/2001 at 16:29:49

Message originating in routine VZTOSZ
Message: Number of Original Source/Sink Nodes =  138
         Number of Revised Source/Sink Nodes  =  138


Analyte Source Total by CFEST Time Step
---------------------------------------------------------
CFEST       Start         End         Total         Total
 Time        Time        Time        Source        Source
 Step         (d)         (d)          (Ci)          (kg)
---------------------------------------------------------
 0001       0.000     182.625   0.00000E+00   0.00000E+00
 0002     182.625     365.250   0.00000E+00   0.00000E+00
 0003     365.250     547.875   0.00000E+00   0.00000E+00
.
.
.
 0109   19723.500   19906.125   1.52346E+02   1.56662E-05
 0110   19906.125   20088.750   1.52346E+02   1.56662E-05
 0111   20088.750   20088.750   0.00000E+00   0.00000E+00

Message originating in routine VZTOSZ
Message: Total Release by STOMP =  6.51072E+05 Ci, or  6.69518E-02 kg
         Total Inject to CFEST  =  6.49282E+05 Ci, or  6.67677E-02 kg
         Discrepancy = 1.79021E+03  Ci (  0.275 %)

Message originating in routine VZDROP
Message: VZDROP Execution Successful
         Execution Finish: 11/19/2001 at 16:29:52
         Execution Duration:      3.0 sec
```

All warning and error messages are also recorded to the log file. The user should make a regular practice of checking the VZDROP log files to assure that mass balance errors are negligible and that no serious errors occurred during execution. As a safeguard, VZDROP will not modify the CFEST L3I file until all calculations are completed and only then if no errors were detected during code execution. If error messages are present, they provide information on the cause of the trouble and where in the source code the error was encountered.

7.0 GWDROP – Groundwater to River Mass Transfer Module

7.1 Code Purpose

GWDROP (Ground Water Data Restructure for Other Programs) is a data transfer utility code that performs the following functions:

- Reads water and contaminant mass fluxes computed by CFEST (groundwater flow and transport module) and prepare input files for use by MASS2 (Columbia River flow and transport module),
- Reads liquid release data from VADER (vadose zone release module) that are sent directly to the river, bypassing the unsaturated and saturated zone transport modules. These releases are included in the input files for use by MASS2 (Columbia River flow and transport module).
- Reads contaminant concentrations computed by CFEST (groundwater flow and transport module) and also writes the concentration data into the ECDA files.

Thus, GWDROP reads input from CFEST and VADER, sends output to MASS2, and inserts data into the existing ECDA files. A user's guide for the CFEST96 code is published separately (CFEST 1996).

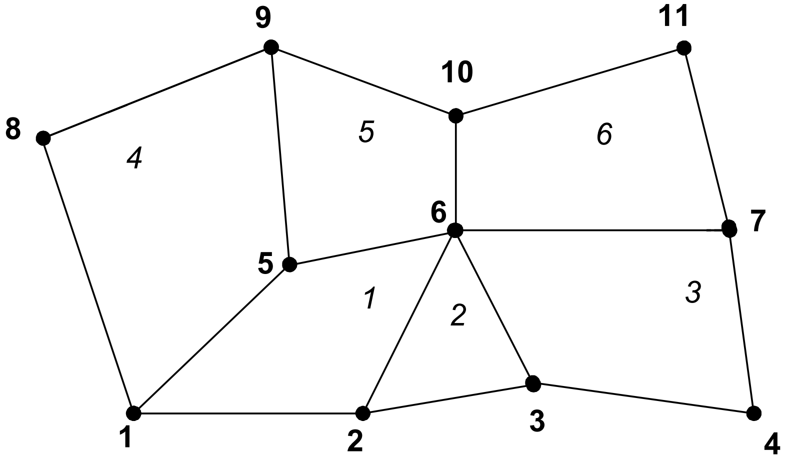
7.2 Algorithms and Assumptions

One function of GWDROP is interfacing data configured for the CFEST finite element grid system with the river cell geometry of MASS2. Though these two grid systems lay one over the other, and use the same system of X,Y coordinates, their grid layouts are in no way dependent one upon the other.

GWDROP determines how flow and mass fluxes from CFEST finite element nodes are distributed to the MASS2 river cells they overlap. For example, if a CFEST node were to exactly overlay 2 river cells without any breaks, and the river cells are of the same size, then half of the flow and mass from the CFEST node would go to one cell and the other half into the other cell. However, most CFEST nodes distribute flow and mass to many river cells. Also, many river cells overlay, and thus receive flow and mass flux from, multiple CFEST nodes. GWDROP takes all of these geometric anomalies into account through the use of a polygon intercept algorithm, which computes the joined area of intercept between 2 arbitrary polygons, in this case 2xN sided CFEST node polygons and quadrangle shaped MASS2 river cells.

7.2.1 CFEST Node Polygons Geometry

A CFEST finite element grid is defined by a set of interconnected points (nodes). The finite element grid forms a geometry of X,Y nodes on a plane. All elements of this grid are made up of either 3 or four nodes, each a near neighbor of the other. See example in Figure 7-1. Example node numbers are 1 through 11. Example element numbers are 1 through 6.



Element	Node	Node	Node	Node
1	1	2	6	5
2	2	3	6	
3	3	4	7	6
4	1	5	9	8
5	5	6	10	9
6	6	7	11	10

Figure 7-1 Example Nodes and Elements Layout

Via the LP1 file, CFEST publishes the list of node numbers, the nodal X,Y locations, and a list of element numbers. The element numbers list identifies the 3 or 4 node numbers that make up the element. In the figure, element 2, made from nodes 2, 3, and 6, is a triangle. All other elements in the figure are 4 sided. Element nodes groupings are also shown in the figure.

Node polygons are the unit of area used in GWDROP for making intercept computations with MASS2 cell geometry. Flux at a CFEST node is considered to be associated with the area of a node polygon. To visualize node polygons, consider the elements that touch the node. Then identify the center point of each element (not the center-of-mass). Also identify the mid-point on all node-connecting lines that touch the object node. For notation, point C_i is the center of element i, and $M_{i\text{-}j}$ is the mid-point of line connecting node i and node j. Now pass a broken line around the object node that passes through the element centers and node-connecting line centers. See Figure 7-2. Nodes 5, 6, and 7 have their nodal polygons identified by broken lines around them. The node 5 polygon is constructed by connecting points C_1, $M_{5\text{-}6}$, C_5, $M_{5\text{-}9}$, C_4, and $M_{1\text{-}5}$ in that order. Node 6 polygon is the broken line connecting C_1, $M_{2\text{-}6}$, C_2, $M_{3\text{-}6}$, C_3, $M_{6\text{-}7}$, C_6, $M_{6\text{-}10}$, C_5, and $M_{5\text{-}6}$. Note that Node 5 polygon and node 6 polygon do not have same number of sides. Node 5 polygon has 6 sides and node 6 polygon has 10 sides. Node 7 is assumed to be on an edge of the finite element grid and thus is constructed from points C_3, $M_{4\text{-}7}$, C_7, $M_{7\text{-}11}$, C_6, and $M_{6\text{-}7}$.

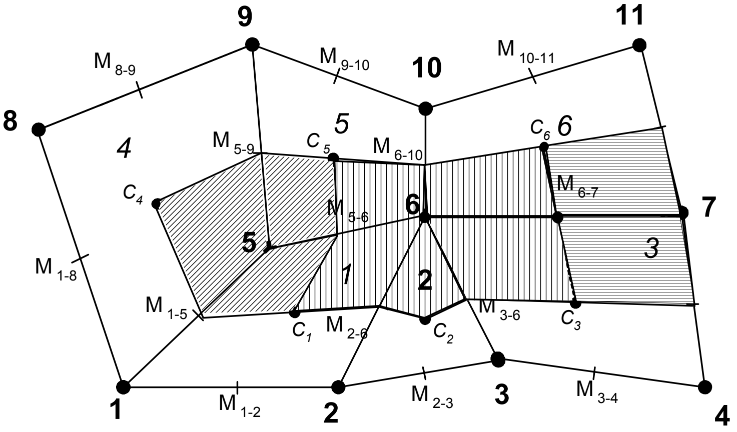


Figure 7-2 Example with Nodal Polygons Shaded

Not all CFEST nodes output to river cells. Nodes that output to the river are defined in the river interface nodes list file, a file that contains a simple list of node numbers. Nodes that are not listed in this file will not have entries in COV or TMS files.

7.2.2 MASS2 Cell Geometry

The MASS2 river cell definition files have file names ending with a numeric extension such as 023. These normally are sequential numbers starting at 000 and ending at NR-1, e.g., 000-059 when NR=60. For example the files surfwater.000, surfwater.001, etc., surfwater.059.

Format of river cell files is NI and NJ in the first record, where NI=number of I's and NJ=number of J's. All remaining lines contain I, J, easting, northing, and elevation. The I-dimension runs the length of the river. The J-dimension runs across the river. There are NI nodes along the river's run and NJ nodes across the river. There are NI*NJ nodes, and (NI-1)*(NJ-1) cells in a river cell file. Cells are quadrangles made up of 4 nodes. Cells could be rectangles, but generally the sides are not required to be parallel. See Figure 7-3 for the general layout.

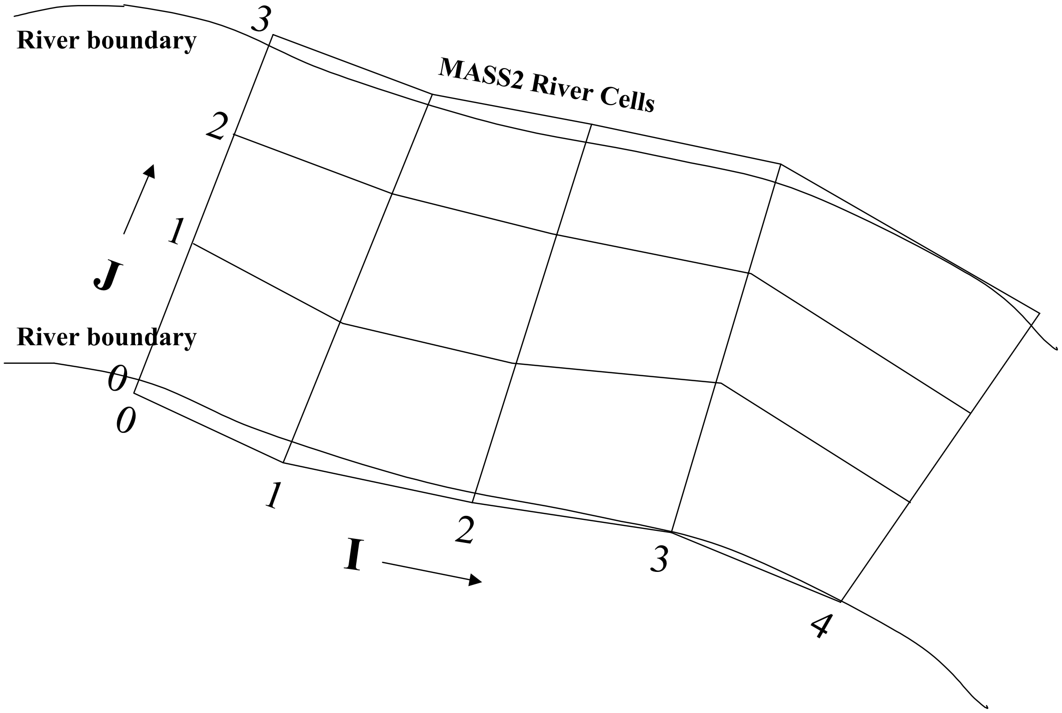


Figure 7-3 MASS2 River Cell Layout

The river cell geometry is defined by NR number of river cell files, NI(R) number of I's, NJ(R) number of J's, XR(R,I,J), YR(R,I,J). Each file contains a different NI and NJ. NI and NJ arrays are dimensioned at NR. XR and YR are dimensioned at (NR,NImax,NJmax), though many of the XR,YR entries are unused. In GWDROP, the elevation entry is not used.

Note that in a direction I or J, there is one fewer river cell than there are river nodes. The first river cell is made from nodes (0,0), (0,1), (1,1), and (1,0). The river cell at I,J is made from nodes (I,J), (I+1,J), (I+1,J+1), and (I,J+1). The last cell in river file R is (NI(R)-2,NJ(R)-2)…(NI(R)-1,NJ(R)-1).

The COV coverage files have nearly the same format as the cell definition files, except that in the COV files only the cells and their indexes are written out. The reference I,J for COV file cells is the minimum I and J of the cell. Thus, there are one fewer entries in I and one fewer in J in a COV file than in a cell file.

7.2.3 POLYCEN Mode for Flux Transfer from CFEST to MASS2

GWDROP uses two methods for determining the fluxes flowing from a CFEST finite element grid to river cells. One method is POLYINT (see Section 7.2.4) where an entire river cell is joined to zero, one, or more CFEST nodal polygons. This section describes the other method, POLYCEN.

POLYCEN mode uses only river cell center points to determine flux distributions. In this algorithm, a river cell center is compared with all CFEST node polygons until the one is found that the river cell center is determined to be inside.

A table is constructed that associates all river cell indexes (R, I, J), cell areas, and node number of the CFEST node polygon that the cell lays within. The algorithm then collects the river cells for each CFEST node number to determine the fraction of its flux going to each river cell. This association is done according to the areas of river cells. Larger cells receive more flux, smaller cells less. If a cell's center point does not lie within any CFEST node polygons, then the cell receives zero flux.

7.2.4 POLYINT Mode for Flux Transfer from CFEST to MASS2

GWDROP utilizes two methods for determining the fluxes flowing from CFEST to river cells. One method is POLYCEN (see Section 7.2.3) where river cell center points are matched up with CFEST nodal polygons. This section describes the other method, POLYINT.

POLYINT mode uses the entire river cell area to determine fluxes. The POLYINT polygon algorithm is used to join polygons, one polygon being the river cell quadrangle, the other polygon being the 2xN sided CFEST nodal polygon.

For any river cell, a search is performed to determine all of the CFEST nodal polygons that the cell intercepts with. The following are possible:

- The river cell intercepts no CFEST node polygons
- The river cell lies entirely within one and only one CFEST node polygon
- The river cell lies within two or more CFEST node polygons.

If a cell intercepts no CFEST nodes, then it receives no flux. For the second and third case, the river cell receives flux from one or more CFEST nodes. The proportions of flux sent from any CFEST node to a river cell is determined by the proportion of river cell areas associated with each CFEST node.

7.2.5 Balance Diagnostics

GWDROP computes two balances, as diagnostics, in order to check the reliability of the computations and output data. These are area balance and mass balance.

Area balance is computed as a diagnostic and written to the main log file. There are two area summations that drive the area balance computation, 1) total area of the CFEST elements, and 2) total area of all nodal polygons generated by GWDROP. CFEST element areas are calculated on an element-by-element basis. These are totaled to get the sum of CFEST element areas. Nodal polygon area is also computed on a node-by-node basis. These are totaled to get the total nodal polygon area. These two areas should be identical.

Mass balance is computed in GWDROP as a diagnostic and written to the main log file. Mass balance uses mass flux values from the CFEST M TAB file. Flow flux balance uses flow flux values from the

CFEST Q TAB file. The input numbers from CFEST for all node polygons for mass flux and flow flux are totaled and saved.

For the balance check, data are read back from the GWDROP-generated TMS files and the VADER-generated release files, and adjusted for values not originating in CFEST. Then, the flux going to each river cell is computed from the contributing CFEST node or combination of nodes. The difference between the CFEST contributions and the adjusted GWDROP values should be zero. The actual difference is reported in the GWDROP log file.

7.2.6 COV and TMS File Naming Algorithms

GWDROP reads nodal polygon flux from CFEST TAB files and distributes the output into a potentially larger number of river cells. Transfer to MASS2 is described in terms of two file types, TMS and COV. TMS files contain time series of mass or flow fluxes. A specific TMS file contains the data for one CFEST node, or for a combination of CFEST node(s) and VADER inputs. COV files contain the coverage for river cells. One entry in a COV file specifies the input for a single river cell.

An entry in a COV file tells MASS2 how to compute input for that entry's river cell. The COV file entry contains a TMS file number (ID) and a fraction between 0.0 and 1.0 (FR). ID selects a single time series TMS file, which must be multiplied by the fraction FR to get actual input. The reason for the fraction is that several river cells may derive input from the same TMS file, each one receiving a different fraction of the total distribution. File names for coverage files are COV-NNN.DAT where NNN is the same number as in the river cell description files from MASS2, (for example, rivercellfile.NNN).

There are 3 file-naming conventions for a TMS file. Each has the format TMS-NNNN-FLOW.DAT or TMS-NNNN-ANALYTE.DAT, where ANALYTE is analyte name, and NNNN is one of the following:

- NNNN = the CFEST node number (1-9999). This numbering scheme is used when a CFEST node contributes to one or more river cells, where these river cells are exclusive to the CFEST node. In this mode, multiple river cells share flux from one and only one TMS file. Note that for the POLYCEN mass flux distribution mode, all cells behave this way except for cells driven by VADER point releases.
- NNNN = a sequential serial number plus 10000. This numbering scheme is used when multiple CFEST nodes contribute to a river cell. In this case there is one and only one river cell per TMS file. The serial number is not a CFEST node number. The first line of the TMS file lists the CFEST node numbers in a comment statement. For TMS files of this type, GWDROP does the final flux computation, thus the fraction for the COV file is 1.0.
- NNNN = a sequential serial number plus 20000. This numbering scheme is used when a river cell gets its input from one or more CFEST nodes and also from one or more VADER releases. For TMS files of this type, GWDROP does the final flux computation, thus the fraction for the COV file is 1.0.

Suppose that the following data have been extracted from a larger COV file. Note a few cells get zero flux.

```
  192    4
    0    0 10001   1.000000
    0    1 10164   1.000000
    0    2 10310   1.000000
    0    3     0   0.000000
    1    0 10002   1.000000
    1    1  2846   0.366522
    1    2  2846   0.239829
. . .
    5    0 20001   1.000000
    5    1 10168   1.000000
    5    2     0   0.000000
    5    3     0   0.000000
```

Examples for TMS file name associated with this example COV file are the following:

- TMS-2846.DAT is a time series file containing the flux from a single node number, 2846. The time series data must be multiplied by the fractions in the COV file entries (river cell (R=1,I=1,J=1) yields 0.366522, and (R=1,R=1,R=2) yields 0.239829) for use in MASS2.
- TMS-10001.DAT is a time series file containing the fluxes from nodes number 2842, 2843, 2846, and 2847 in different proportions for river cell (R=1,I=0,J=0). The COV entry fraction is 1.0, thus the TMS file contains the final flux for this river cell. GWDROP did the math to combine fluxes from the 4 nodes for this cell.
- TMS-20001.DAT is a time series file containing the flux from one or more CFEST nodes, scaled to correct output, with some point release data from a VADER file added in. This TMS file provides input to a single river cell (e.g., R=1,I=5,J=0). The COV entry fraction is 1.0, thus the TMS file contains the total flux to river cell (1,5,0).

7.2.7 Time and Delta-Time Algorithms

GWDROP deals with 2 different time scales, that of CFEST and that of MASS2. In CFEST, there is no absolute time, only delta time from the start of the run, expressed in units of days. MASS2 uses absolute calendar date-and-time, 21-MAR-2002 08:46:00, for example. Delta-time is then calculated as the elapsed time between two absolute date-and-time entries.

GWDROP uses the standard Gregorian calendar to compute absolute time at advancing time steps, based on the delta-time from CFEST. For very long simulations, this approach introduces slippage between the calendars because CFEST has no algorithm for leap years. Slippage can be kept under about 1.47 days for all elapsed simulation times by assuming a calendar year is equal to 365.2425 days in CFEST. This approach takes into account the 4-year, 100-year, and 400-year leap-year algorithms of the Gregorian calendar. By this calculation, one expects 146097 days in any 400-year period regardless of when it begins and ends. Thus, 146097/400 = 365.2425 days per average year.

7.2.8 Concentration Data Extraction

GWDROP sends analyte concentrations to the ECDA files at the impact locations and years specified in the ESD keyword file (see Sections 2.1.15 and 2.1.27). Site-wide concentrations are derived from the CFEST binary RESULT file, not from TAB mass and volume files. CFEST and ECDA do not use the same system of units and therefore a units conversion is necessary. Section 2.2.1 provides the format and data units for the ECDA files.

The main purpose of algorithm LCONCSUB is to convert from the concentration data formats used by CFEST to those needed in GWDROP. The algorithm is named after a subroutine in the CFEST source code. The LCONCSUB method reads CFEST binary RESULT files, which contain concentrations at CFEST nodes and time-steps. CFEST models analytes using mass units rather than activity units. The ECDA data for radioactive analytes are stored in terms of activities rather than mass. GWDROP translates between these two sets of units.

If the data are from a 3-D CFEST model, the concentration saved for later output to the ECDA files is the maximum concentration from over all depths in the model domain. The data are adjusted for units changes, but no other changes are required.

If the data are from a 2-D CFEST model, the concentration data are taken from the surface node. The concentrations are adjusted for units changes, then multiplied by ($H_{HEAD} – Z$) and divided by T_{THICK},

where:

H_{HEAD} is hydraulic head (meters)
Z is elevation (meters)
T_{THICK} is the transport thickness parameter (meters).

This last calculation is a rough method for translating concentrations in a 2-D domain to equivalent real-world concentrations.

7.3 Code Execution Environment

GWDROP was developed as an internal utility program to be invoked by the master SAC control software, SAC ESP.

7.3.1 Location in the Processing Sequence

GWDROP execution takes place between CFEST and MASS2 in the processing sequence. VADER runs must also precede GWDROP runs. The order in which codes are executed is provided in Figure 1-2.

7.3.2 How the GWDROP Code Is Invoked

The GWDROP code can run either under the Windows operating system or the Linux operating system. Under the Windows operating system (Releases 95, 98, NT or 2000) the GWDROP code executes in a DOS box. A run of the GWDROP code is initiated by entering any of the following command lines:

```
GWDROP GWDROP.LIS
GWDROP –D GWDROP.LIS
GWDROP –D GWDROP.LIS  >>X.OUT
```

Under the Linux operating system GWDROP is executed through any of the following Bourne Shell or C Shell commands:

```
gwdrop.exe gwdrop.lis
gwdrop.exe–D gwdrop.lis
gwdrop.exe –D gwdrop.lis >x.out
```

In the above examples, GWDROP.LIS is the full or partial path and name for the GWDROP input files list file that describes the options and input files to be processed in GWDROP (see Section 7.4.1.1). The option –D indicates to activate GWDROP debug diagnostic output. This output is very voluminous and should be redirected to a file as in the third command line. Option –D should never to be used in normal processing. It should only when used when extended diagnostic output is needed for debugging, troubleshooting, and problem solving. Under Linux file names are case sensitive, while under Windows they are not.

Currently Windows and Linux are the only operating systems supported in SAC. Furthermore, the only operating system variable needed in GWDROP is the directory separation slash character (\) or (/). The backwards slash (\) is the directory separation character in Windows. In Linux, the forward slash (/) is the directory separation character in file names.

GWDROP examines all file names contained on the command line argument and in the input file list file. If any back slash (\) characters appear, then Windows is assumed to be the operating system. If only the forward slash (/) appears in file names, then Linux is assumed. This slash character is then utilized in all GWDROP path name computations, such as for appending local file names to paths.

7.3.3 Code Control and Keyword Descriptions

The GWDROP input files list file contains a list of all files needed to run GWDROP along with several option values. The exact file name of this file, nominally GWDROP.LIS, is passed to GWDROP on the command line. This section provides the format of this file. Section 7.4.1.1 provides an annotated example file.

This file contains one entry per line. The first 16 entries are required to be on a specific line. Data on lines 17 to the end of file are less restricted. Table 7.1 provides descriptions of the entries for the first 16 lines.

Table 7.1 Definition of Fixed Content Lines in the GWDROP.LIS File

Line	Description
1	Path and name of the GWDROP log file.
2	Relative path to CFEST CTL file.
3	Relative path to CFEST LP1 file (unused entry).
4	Relative path to CFEST L3I file (unused).
5	Relative path to Columbia River elements file (unused).

Line	Description
6	Relative path to Columbia River nodes file.
7	Relative path to POLYGONS.DAT file.
8	TMS files prefix, MASS2\TMS for example.
9	COV files prefix, MASS2\COV for example.
10	Simulation start date and time in the format *MM-DD-YYYY HH:MM:SS*.
11	Polygon intercept options, enter POLYINT or POLYCEN. Additional modifiers for this line are TTHICK, SKIP, RIVER, and CLIP. • TTHICK – transport layer thickness for a 2-D CFEST model run. • SKIP - Number of CFEST time planes to skip. • RIVER – Denotes the river file prefix for use in Table 7.2. By default the river files character is R. Additional river files can be specified by using the letter S. Then using "RIVER S" causes GWDROP to process the S river files rather than the R river files. • CLIP NODES – If specified, clips every node output to MASS2 in every time step to positive. • CLIP TOTAL – If specified, clips non-positive nodes to zero when total output to MASS2 is negative. • CLIP NONE – No clipping is performed. This is the default. Example: **`POLYINT : TTHICK 20.000 : SKIP 1 : RIVER R : CLIP NODES`**
12	Realization number to process. Enter an integer.
13	Q.TAB file I format for reading node number list. Example **`(A,T13,0500I15)`**.
14	Q.TAB file G format for reading flow flux values. Example **`(A,T13,0500G15.7)`**.
15	M.TAB file I format for reading node number list. Example **`(A,T13,0500I15)`**.
16	M.TAB file G format for reading flow flux values. Example **`(A,T13,0500G15.7)`**.

Variable context items are entered beginning with line 17. The context is determined by the set of one-character keys described in Table 7.2.

Table 7.2 Definition of Variable Context Lines for the GWDROP.LIS File

Key	Description
K	Relative path and name for the ESD keyword file.
Q	Relative path and name for the LLHELD TAB flow flux file.
A	Analyte parameters (analyte name, radioactive status, and specific activity) for the M: entry in the next line. Example: A:H3,RADIOACTIVE,9724.5.
M	Relative path and name of the LLHELD TAB mass flux file for the analyte for the preceding A: entry.

Key	Description
B	Relative path to the CFEST binary files directory – files required are B01, B03, B07, and ZZZ.
R	Absolute path the river cell geometry files from MASS2. More than one R: entry can be made.
S	Name of the alternate river cell geometry – typically a coarse grid.
V	Relative path and name for a VADER release file. More than one V: entry can be made. The floating-point numbers following the file names are X,Y locations for the release. Example: **`VADER\116-B-7\H3\05\Vader.river, 565257.3, 145324.5`**
-	-: Comments out a previously used entry.

Normally there will be a K entry, followed by Q, A, M, and several R entries. Note that A and then M always go together. The A entry specifies the analyte properties associated with the M item that follows. GWDROP can handle multiple A and M pairs, that is, GWDROP can process of multiple analytes in a single run. However, neither CFEST nor SAC ESP currently support this ability. Thus, there will always be Q, A, and M entries in that order without any repeats. MASS2 river cell definition file names come from R: entries. Note key-R is the default but can be changed by using the RIVER option in the 11th item Table 7.1. If there are S entries, they typically are for a coarse river cell grid.

7.4 Data Files

GWDROP reads a control file, and many different files from VADER, CFEST, and MASS2. In addition, GWDROP writes a log file and potentially hundred of files for use in MASS2 or the SAC impacts codes.

7.4.1 Input Files

GWDROP reads a control file, and many different files from VADER, CFEST, and MASS2. The following sections describe the input files.

7.4.1.1 Input List File (GWDROP.LIS)

The GWDROP input files list file contains a list of all files needed to run GWDROP along with several option values. The exact file name of this file, often GWDROP.LIS, is passed to GWDROP on the command line. Section 7.3.3 describes the format of this. Table 7.3 provides an example file for a run with realization 21 for the analyte I129. Each line has been annotated with a line number in order to enhance the descriptions.

Table 7.3 Example GWDROP.LIS File for I129, Realization 21

```
 1. GWDROP\I129\21\gwdrop.log
 2. CFEST\I129\21\cfest.ctl
 3. CFEST\I129\21\input.lp1
 4. CFEST\I129\21\input.l3i
 5. GWDROP\Col_River_Elem.dat
 6. GWDROP\Col_River_Node.dat
 7. GWDROP\I129\21\POLYGONS.DAT
 8. GWDROP\I129\21\TMS
 9. GWDROP\I129\21\COV
10. 01-01-1944 00:00:00
11. POLYINT : TTHICK 20.000 : SKIP 1 : RIVER R
12.      21
13. (A,T13,0500I15)
14. (A,T13,0500G15.7)
15. (A,T13,0500I15)
16. (A,T13,0500G15.7)
17. K:ESD_Initial.key
18. Q:CFEST\I129\21\hheldq001.tab
19. A:I129,RADIOACTIVE, 1.7869000E-04
20. M:CFEST\I129\21\hheldm001.tab
21. B:CFEST\I129\21
22. R:G:\Initial\GWDrop\hanfnad83m-pt.000
23. R:G:\Initial\GWDrop\hanfnad83m-pt.001
24. R:G:\Initial\GWDrop\hanfnad83m-pt.002
25. V:VADER\116-B-7\I129\21\vader.river,  565257.4, 145324.6
26. V:VADER\116-D-5\I129\21\vader.river,  573502.2, 152319.1
27. V:VADER\116-DR-5\I129\21\vader.river, 573625.6, 152441.5
28. V:VADER\116-F-8\I129\21\vader.river,  580963.8, 148107.7
29. V:VADER\116-H-5\I129\21\vader.river,  578113.4, 152821.3
30. V:VADER\116-K-3\I129\21\vader.river,  568911.6, 146964.9
31. V:VADER\132-C-2\I129\21\vader.river,  565719.8, 145481.0
32. V:VADER\1908-NE\I129\21\vader.river,  570870.8, 149317.8
```

River element and river node files are specified on lines 5 and 6 of GWDROP.INP. The river elements file is not presently used. The river nodes file provides GWDROP with a list of CFEST nodes that interface with the river model. This is a simple sequential list of node numbers.

Lines 8 and 9 of GWDROP.LIS specify the prefix to TMS time series and COV coverage files. The time series file prefix is GWDROP\I129\21\TMS and the coverage file prefix is GWDROP/I129\21\COV. The name and path for the ESD keyword file is provided on line 17 of GWDROP.LIS. The ESD keyword file specification is identified with a K: prefix.

Line 11 of GWDROP.LIS provides a list of options for GWDROP processing as follows:

- POLYINT – Specifies that POLYINT (polygon intercept mode) rather then POLYCEN (polygon vs. cell centers mode) be utilized for computing distribution of fluxes from CFEST nodal polygons to MASS2 river cells.
- TTHICK 20.000 – Specifies the transport layer thickness. Units are standard CFEST simulation thickness units.
- SKIP 1 – Specifies that first time step from CFEST is to be skipped in the TMS files time series output.
- RIVER R – Specifies to utilize the R: entries in the GWDROP.LIS as the river configuration.
- CLIP NODES – Specifies to clip all outputs to MASS2. Entering this option tells GWDROP to clip any nodes with negative output found in any time step. Negative values from CFEST are possible due to numerical processing noise. Entering this option affects mass balance.

- CLIP TOTAL – Specifies to clip all output to MASS2 only if the total output to MASS2 in any time step is found to be negative. This option also affects mass balance, though not as much as CLIP NODES.
- CLIP NONE – Specifies to not clip any outputs to MASS2. This is the default, i.e., when no CLIP options specified.

TAB file formats are specified on lines 13 through 16 of GWDROP.LIS. These formats provide the FORTRAN formats of records that CFEST or HHELD writes to TAB files. The first two formats are for Q flow files. The second two formats are for M mass flux files. The *I* version is for reading the list of CFEST node numbers. The *G* version is for reading flow or mass data. The *A* is basically ignored. The data in the files begins in column 13. In all cases field widths are 15 characters.

Starting on line 18 of GWDROP is a list of Q TAB files, M TAB files, and Analyte IDs. The prefixes Q:, M:, and A: are used to identify these specifications. Generally there is one and only one Q.TAB and M.TAB file, as shown in this example. However, GWDROP can also support a singe Q file and multiple M files. M file specifications are always preceded by an A: specification that which provides analyte information. The flow TAB file is read from CFEST\I129\21\hheldq001.tab. Analyte properties are name=I120, status=RADIOACTIVE, and specific activity=0.00017869 curies per gram. The mass TAB file for iodine-129 is read from CFEST\I129\21\hheldm001.tab.

The path to binary files output by the corresponding CFEST run is specified on line 21 of GWDROP.LIS. The B: prefix keys this entry. In this example, the path is CFEST\I129\21. GWDROP reads several binary files from this directory in order to compute concentrations for ECDA. The input files and input paths from CFEST are specified on lines 2, 3, 4, 20, and 21 of GWDROP.INP.

Lines 22 through 24 of GWDROP.LIS specify the path and names of MASS2 river cell geometry files. Prefix R: is present on these lines to identify river files. The path name is G:\Initial\GWDrop\hanfnad83m-pt.###, where ### is 000, 001, or 002. Each of these files can portray a different geometry, and generally are situated end-to-end along the river flow direction.

Lines 25 through 32 of GWDROP.LIS provide information about the VADER files. The prefix V: identifies VADER files. The generalized path is VADER\SITEID\ANALYTE\REAL. All the files have the name vader.river. The floating-point numbers following the file names are X,Y locations of the release site.

7.4.1.2 CFEST ASCII Files

GWDROP reads the CTL, LP1, L3I, Q.TAB, HHELDQ.TAB, HHELDM.TAB, and M.TAB ASCII files used by CFEST. The name of the CFEST control file, nominally CFEST.CTL, is obtained from the GWDROP.LIS file in the second entry (see Table 7.1). Items used from CFEST.CTL include path to binary files, path to LP1 file, path to L3I, and some units. The path to LP1 and L3I files are obtained from the CFEST.CTL file. That is why the corresponding entries in Table 7.1 have been labeled as *unused*.

The Q.TAB file contains flow flux for a list of CFEST nodes. The M.TAB file contains the mass flux for the same list of nodes. This is not the same node list as contained in the sixth item of Table 7.1. The HHELDM.TAB file contains mass flux for the analyte being processed. Analyte properties come from the A: entry of the GWDROP.LIS file. There are no analyte properties contained in any of the CFEST files. To get curies, GWDROP multiplies mass from the TAB file by specific activity obtained from the A: entry of the GWDROP.LIS file.

7.4.1.3 CFEST Binary Files

The CFEST binary files used by GWDROP are B01, B07, B08, and ZZZ. The path to these files is provided in the GWDROP.LIS file at the B: item. These files are output from a CFEST model run.

The B01 file contains the same data as the LP1 file but in binary format. The B07 file contains the simulation results. The B08 file contains the same data as in the L3I file but in binary format. File ZZZ contains initial structure in a legacy format.

7.4.1.4 CFEST TAB Files

GWDROP reads two TAB files, one the Q file containing flow flux, and the other the M file containing mass flux. These TAB files are generated by LGETHHELD, the CFEST get held node utility.

Reading of Q and M TAB file data is accommodated via the format strings contained within the GWDROP.LIS file. These formats are included in the thirteenth to sixteenth lines of that file (see Table 7.1). The format strings used in LGETHHELD to write the TAB files should be input to GWDROP. Generally, there is one record in a TAB file for each simulation time step, beginning at the starting date/time as contained in line 10 of the GWDROP.LIS file.

7.4.1.5 River Interface Nodes List File

GWDROP requires a river interface nodes list file. This file tells GWDROP which nodes constitute the mass balance interface between CFEST and MASS2. Format of the river interface nodes list file is one node number per line. This file must be is prepared by the GWDROP user.

There are actually three node lists. The first is a list of CFEST nodes that comes from the LP1 (or B01) file. This list defines all of the nodes of the finite element grid. The second is a list of CFEST held nodes. This list is contained in the LP1 file and defines the list of held nodes; nodes for which flux is to be computed and output. Finally there is the river interface nodes list special to GWDROP that defines the node numbers that interface to the river. The nodes in the interface list must be a subset of the held node list. Nodes in the held nodes list, but not in the river interface list, are not used as output to the river model. Interface nodes that are not held nodes, if any, receive zero flux.

7.4.1.6 SAC ESD Keyword File

GWDROP reads the ESD Keyword file see in order to access the following keywords:

- REALIZATION (Section 2.1.20)
- LOCATION (Section 2.1.15)
- TIMES (Section 2.1.27).

These keywords provide data needed by GWDROP for indexing into the ECDA files. The ESD keyword file is named at the K: entry of the GWDROP.LIS file (see Table 7.2).

7.4.1.7 Environmental Concentration and Map Files

Concentration data for every combination of analyte and impact type (human, cultural, economic, or ecological) are contained in a separate file. GWDROP reads from, and writes to, a concentration data file for every combination of impacts type and analyte being simulated. If there are 4 impact types and 10 analytes, then 40 concentration data files will be required, along with 4 (one for each impacts type) record map files. Sections 2.2.1 and 2.2.2 describe the format for each of these files. The concentration data and map file names are supplied in the ESD keyword file (see Section 2.1.10).

7.4.1.8 MASS2 River Cell Files

The MASS2 river cell definition files have file names ending with a three-digit numerical extension. These extensions normally are sequential numbers starting at 000 and ending at NR-1, for example, the extensions are in the range 000-009 when NR=10. NR denotes the number of blocks that subdivide the river grid system. For example, the file names for SAC, Rev. 0, runs are hanfnad83m-pt.000, hanfnad83m-pt.001, and hanfnad83m-pt.002.

The format of the first record of a river cell file is two integers, NI and NJ. Where NI=number of I's (number of points in the grid along the river) and NJ=number of J's (number of points in the grid across the river). All remaining lines contain I, J, easting, and northing as well as elevation. See Section 7.2.2 for a definition of the river cell geometry. Table 7.4 provides the first 21 lines from an example river cell file. The file actually contained 966 lines of data.

Table 7.4 First 21 Lines from a MASS2 River Cell File

```
  193     5
   0    0     557375.625     144890.703     339.498
   0    1     557359.062     144961.297     339.508
   0    2     557342.625     145031.906     339.526
   0    3     557326.062     145102.500     339.559
   0    4     557309.625     145173.109     339.607
   1    0     557865.438     144995.781     355.607
   1    1     557843.938     145085.000     356.028
   1    2     557822.438     145174.172     356.447
   1    3     557801.062     145263.359     356.853
   1    4     557779.562     145352.531     357.204
   2    0     558299.250     145241.766     370.010
   2    1     558277.188     145332.609     369.935
   2    2     558255.125     145423.406     369.864
   2    3     558233.062     145514.203     369.794
   2    4     558211.125     145604.969     369.734
   3    0     558730.250     145490.250     368.528
```

```
    3    1     558703.500     145587.000     368.455
    3    2     558676.812     145683.688     368.378
    3    3     558650.250     145780.359     368.297
    3    4     558623.625     145877.000     368.240
```

7.4.1.9 VADER River Release Files

GWDROP reads the VADER river release files in order to supplement CFEST fluxes passed to MASS2. VADER flux is passed to the river cell where the VADER X,Y is located. This X,Y location is passed to GWDROP in the GWDROP.LIS file as qualifiers to V: records (see Table 7.2).

Section 5.5.2.2 describes the contents of a VADER river release file. Table 5.19 provides an example file. GWDROP ignores the data in the first seven lines of the file. The value for NREC is read from the eighth line (number of records of data), and then the records of following data are read. Each following record contains values for the variables YEAR, VOL, and QUA; where YEAR is an integer year, VOL is the volume of discharge in m^3 and QUA is the quantity of an analyte discharged in Curies or kilograms. VOL and QUA are floating point numbers.

7.4.2 Output Files

GWDROP writes a log file and potentially hundred of files for use in MASS2 or the SAC impacts codes. The following sections describe these output files.

7.4.2.1 GWDROP Log file

Generally, the log file is created at GWDROP\GWDROP.LOG, as established in the Environmental Stochastic Processor (ESP) module prior to execution of GWDROP. Code that writes to the log file is scattered throughout GWDROP and provides a rough summary of code execution. Valuable information includes area and mass balances and names of files opened for reading. Any file not found is reported in an error message to the log file. The approximate formats of items included in this log file are:

- Every log file record begins with the current date and time to one thousandth of a second followed by two colons, e.g., `11-SEP-2001 08:46:00.000 ::`
- First line is the GWDROP version ID, e.g., `GWDROP version 1.13 build 25 Jul 2001 started...`
- Files opened, e.g., : `GWDROP opening/reading file FILE_NAME_OF_FILE ...`
- General status e.g., : `GWDROP reading MASS2 river cell files…`
- List of non-river nodes mapped to river nodes. If a nodal polygon with flux output does not map onto a node from the river nodes list, then it is mapped to the closest node included in the river nodes list. The list of such nodes is reported in the log file, e.g., `Non river node 321 mapped to river node 2310.`
- Area balance from CFEST, e.g., `Total of all CFEST element area = 806762012.0000000`.
- Area balance to MASS2, e.g., `Total of all NODE POLYGON areas = 806762012.0000001`.
- Flow (volume) balance ratio, e.g., `Maximum Q balance ratio = 4.23E-15`.

- Mass (kg) balance radio, e.g., `Maximum M balance ratio = 2.10E-15`.
- Number of negative concentrations, e.g., `Number of negative CONC from CFEST = 43`.
- Done message, e.g., `GWDROP finished/successful`.

The relative path to the GWDROP log file is specified on line 1 of the GWDROP.INP (see Table 7.1). File names can also be specified in absolute full path names. Table 7.5 provides an example log file for a run for realization 21 of I129.

Table 7.5 Example GRDROP Log File

```
11-SEP-2001 08:46:34.828 :: GWDROP version 1.13 build 25 Jul 2001 started...
11-SEP-2001 08:46:34.843 :: GWDROP reading CFEST LP1 input file...
11-SEP-2001 08:46:34.843 :: GWDROP opening/reading file CFEST\CCl4\21\input.lp1
11-SEP-2001 08:46:35.062 :: GWDROP reading CFEST border nodes file...
11-SEP-2001 08:46:35.062 :: GWDROP allocating dynamic memory...
11-SEP-2001 08:46:35.062 :: GWDROP computing CFEST node/element areas...
11-SEP-2001 08:46:35.109 :: Total of all CFEST element area = 806762012.0000000
11-SEP-2001 08:46:35.109 :: Total of all NODE POLYGON areas = 806762012.0000001
11-SEP-2001 08:46:35.156 :: GWDROP reading CFEST flow & analyte files...
11-SEP-2001 08:46:35.406 :: GWDROP opening/reading file CFEST\CCl4\21\hheldq001.tab
11-SEP-2001 08:46:37.953 :: GWDROP opening/reading file CFEST\CCl4\21\hheldm001.tab
11-SEP-2001 08:46:40.187 :: GWDROP reading MASS2 river cell files...
11-SEP-2001 08:46:40.187 :: GWDROP opening/reading file G:\Initial\GWDrop\hanfnad83m-pt.000
11-SEP-2001 08:46:40.234 :: GWDROP opening/reading file G:\Initial\GWDrop\hanfnad83m-pt.001
11-SEP-2001 08:46:40.234 :: GWDROP opening/reading file G:\Initial\GWDrop\hanfnad83m-pt.002
11-SEP-2001 08:46:40.281 :: GWDROP opening/reading file G:\Initial\GWDrop\hanfnad83m-pt.000
11-SEP-2001 08:46:40.296 :: GWDROP opening/reading file G:\Initial\GWDrop\hanfnad83m-pt.001
11-SEP-2001 08:46:40.296 :: GWDROP opening/reading file G:\Initial\GWDrop\hanfnad83m-pt.002
11-SEP-2001 08:46:42.359 :: GWDROP reading CFEST river nodes file...
11-SEP-2001 08:46:42.359 :: GWDROP opening/reading file GWDROP\Col_River_Node.dat
11-SEP-2001 08:46:42.406 :: Non river node 11 mapped to river node 14
11-SEP-2001 08:46:42.406 :: Non river node 12 mapped to river node 20
11-SEP-2001 08:46:42.406 :: Non river node 1985 mapped to river node 1988
11-SEP-2001 08:46:42.406 :: Non river node 2591 mapped to river node 2595
11-SEP-2001 08:46:42.406 :: Non river node 2593 mapped to river node 2594
11-SEP-2001 08:46:42.406 :: Non river node 2651 mapped to river node 2654
11-SEP-2001 08:46:42.406 :: Non river node 2652 mapped to river node 2653
11-SEP-2001 08:46:42.406 :: Non river node 3041 mapped to river node 3025
11-SEP-2001 08:46:42.406 :: Non river node 3042 mapped to river node 3058
11-SEP-2001 08:46:42.406 :: Non river node 3056 mapped to river node 3059
11-SEP-2001 08:46:42.406 :: Non river node 3057 mapped to river node 3058
11-SEP-2001 08:46:42.406 :: Non river node 3100 mapped to river node 3106
11-SEP-2001 08:46:42.406 :: Non river node 3101 mapped to river node 3102
11-SEP-2001 08:46:42.406 :: Open/read VADER\116-B-7\I129\21\Vader.river
11-SEP-2001 08:46:42.437 :: Open/read VADER\116-D-5\I129\21\Vader.river
11-SEP-2001 08:46:42.468 :: Open/read VADER\116-DR-5\I129\21\Vader.river
11-SEP-2001 08:46:42.484 :: Open/read VADER\116-F-8\I129\21\Vader.river
11-SEP-2001 08:46:42.500 :: Open/read VADER\116-H-5\I129\21\Vader.river
11-SEP-2001 08:46:42.531 :: Open/read VADER\116-K-3\I129\21\Vader.river
11-SEP-2001 08:46:42.546 :: Open/read VADER\132-C-2\I129\21\Vader.river
11-SEP-2001 08:46:42.578 :: Open/read VADER\1908-NE\I129\21\Vader.river
11-SEP-2001 08:46:42.593 :: Open/read VADER\116-B-7\I129\21\Vader.river
11-SEP-2001 08:46:42.593 :: Open/read VADER\116-D-5\I129\21\Vader.river
11-SEP-2001 08:46:42.593 :: Open/read VADER\116-DR-5\I129\21\Vader.river
11-SEP-2001 08:46:42.593 :: Open/read VADER\116-F-8\I129\21\Vader.river
11-SEP-2001 08:46:42.609 :: Open/read VADER\116-H-5\I129\21\Vader.river
11-SEP-2001 08:46:42.609 :: Open/read VADER\116-K-3\I129\21\Vader.river
11-SEP-2001 08:46:42.609 :: Open/read VADER\132-C-2\I129\21\Vader.river
11-SEP-2001 08:46:42.609 :: Open/read VADER\1908-NE\I129\21\Vader.river
11-SEP-2001 08:49:36.859 :: GWDROP writing CFEST output files for MASS2...
```

```
11-SEP-2001 08:52:34.296 :: Maximum Q balance ratio = 3.68E-14
11-SEP-2001 08:52:34.296 :: Maximum M balance ratio = 3.27E-15
11-SEP-2001 08:52:34.296 :: GWDROP writing ECDA output files...
11-SEP-2001 08:54:10.250 :: Number of negative CONC from CFEST = 143
11-SEP-2001 08:54:10.250 :: GWDROP finished/successful.
```

7.4.2.2 Time Series Files (TMS)

TMS files contain the time series for flow and mass fluxes being sent to MASS2. See Section 7.2.6 for the file naming conventions used for TMS files and COV files. The file name contains either FLOW or the analyte name, e.g., H3, U238, etc. As an example, Table 7.6 provides the first five years of a flow TMS file for serial number 10001. The numbers on the comment line are the CFEST node numbers contributing flux to the river cell mapped to serial 10001.

Table 7.6 Excerpted Records from a Flow TMS file for Serial Number 10001

```
# GWDROP\CCl4\08\TMS-10001-FLOW.DAT     2842     2843     2846     2847
07-01-1944 14:23:59     0.000000000000000E+00
12-31-1944 04:47:59    -0.109513882379411E+07
07-01-1945 21:35:59    -0.218858688180587E+07
12-31-1945 12:00:00    -0.328178503691099E+07
07-02-1946 02:24:00    -0.437512642446478E+07
12-31-1946 19:11:59    -0.546877324842484E+07
07-02-1947 09:36:00    -0.656279277083112E+07
01-01-1948 00:00:00    -0.765718727997929E+07
07-01-1948 14:23:59    -0.875193253110469E+07
12-31-1948 04:48:00    -0.984699777995582E+07
```

As a second example, Table 7.7 provides the first five years of data for a tritium TMS flux file for serial number 10007.

Table 7.7 Excerpted Records from a Flux TMS file for Serial Number 10007

```
# GWDROP\H3\08\TMS-10007-H3.DAT     2916     2917
07-01-1944 14:23:59     0.000000000000000E+00
12-31-1944 04:47:59     0.000000000000000E+00
07-01-1945 21:35:59     0.125004162948587E-32
12-31-1945 12:00:00     0.894362208875261E-30
07-02-1946 02:24:00     0.449334424148022E-20
12-31-1946 19:11:59     0.417628826019358E-19
07-02-1947 09:36:00     0.210719851798628E-18
01-01-1948 00:00:00     0.765512494485426E-18
07-01-1948 14:23:59     0.224367904579722E-17
12-31-1948 04:48:00     0.563907797302216E-17
```

7.4.2.3 Coverage Files (COV)

COV and TMS files provide the input to MASS2 river model. The COV files characterize TMS file distributions for all cells in a river model system. See Section 7.2.6 for the file naming conventions used in COV files. The first line contains the I,J dimensions, same as river cell descriptions (see Section 7.4.1.8). Remaining lines contain I, J, TMS file key, and a fraction between 0.0 and 1.0. The fraction gets multiplied by flow or mass fluxes in MASS2 in order to compute inputs. Table 7.8 provides a three-river cells column example.

Table 7.8 Example COV File Written by GWDROP

```
 192    4
   0    0 10001   1.000000
   0    1 10164   1.000000
   0    2 10310   1.000000
   0    3     0   0.000000
   1    0 10002   1.000000
   1    1  2846   0.366522
   1    2  2846   0.239829
   1    3     0   0.000000
   2    0 10003   1.000000
   2    1 10165   1.000000
   2    2 10311   1.000000
   2    3     0   0.000000
```

7.4.2.4 Environmental Concentration Files

Concentration data for every combination of analyte and impact type (human, cultural, economic or ecological) are contained in separate files. GWDROP reads from, and writes to, a concentration data file for every combination of impact type and analyte being simulated. If there are 4 impact types and 9 analytes, then 36 concentration data files will be required, along with 4 (one for each impacts type) record map files. GWDROP does not modify the map files. The format for each of these files is described in Sections 2.2.1 and 2.2.2, respectively. The concentration data and map file names are supplied using the FILE keyword (see Section 2.1.10) in the ESD keyword file.

An ECDA concentration file for a specific analyte contains data for all environmental media and all realizations for the overall problem being simulated. GWDROP processes groundwater data for only one analyte and one location during any one run. Thus, GWDROP reads groundwater records from the ECDA that contain data for all realizations, modifies the data for a single realization, and writes the entire record back to the file. Section 7.2.8 describes the translation between concentrations computed by CFEST and the concentrations in the ECDA files.

7.4.3 Example POLYGONS.DAT File

A nodal polygons output file path name POLYGONS.DAT can be specified on line 7 of the GWDROP.LIS file (see Table 7.1). This file receives a listing of the CFEST nodal polygons in Arc/Info format. The format is node number followed by a list of X,Y pairs, followed by END. Table 7.9 provides an example file for containing two nodal polygons.

Table 7.9 Example Nodal Polygons Output File from GWDROP

```
   14      597266.250      103603.000
           597268.500      103356.000
           597444.500      103011.250
           598058.500      103030.500
           597793.000      103352.000
           597582.000      103587.500
   END
    19     595930.250      103814.500
           596389.000      103947.000
           596669.750      103782.750
           596824.000      104038.000
           596888.250      104067.500
           596512.500      104285.000
           596226.750      104419.000
           596190.500      104345.000
           595793.750      104360.500
           595956.000      104098.000
   END
```

8.0 CRDROP – Columbia River Concentration Extraction Module

8.1 Code Purpose

CRDROP (Columbia River Data Restructure for Other Programs) is a data transfer utility code that reads output files generated by the river model (MASS2), extracts concentration data, and places the concentration data into the ECDA files for human, ecological, economic, and cultural impact locations.

8.2 Algorithms and Assumptions

No release or transport calculations are performed by the CRDROP code. The processes involve reading data from river model result files and placing selected data into the ECDA concentration data files (see Section 2.2.1). The algorithm chooses the river solution at the closest location to the desired location for impacts calculations. Section 8.5.2 describes the selection algorithm.

8.3 Code Environment

The CRDROP code is run under the control of the ESP. The CRDROP code is run for one realization and one analyte at a time.

8.3.1 Location in Processing Sequence

The CRDROP code runs after the MASS2 (river) model has completed execution for an analyte for a given realization. Figure 1-2 shows the location of CRDROP in the processing sequence. Prior to running CRDROP for the first time, the utility code CRDROP_INDEX must be run to establish a cross-index file giving the river location that corresponds to each impact location (for human, ecological, economic, and cultural impact locations). Section 8.5 describes the CRDROP_INDEX code.

8.3.2 How the CRDROP Code Is Invoked

CRDROP may be used in either Microsoft Windows or Linux environments, provided the code has been compiled by an appropriate Fortran 95 compiler for the environment. In the Microsoft Windows environment CRDROP is executed by the following DOS command:

```
   CRDROP  ‘ESDfilename’  runid  N1 Num analyte(1) … analyte(num)
```

In the Linux environment CRDROP is executed through the following Bourne Shell or C Shell commands:

```
   crdrop.exe  ‘ESDfilename’  runid  N1 Num analyte(1) … analyte(num)
```

where:
   CRDROP or crdrop.exe is the name of the executable program

‘ESDfilename’ is the name of the Environmental Settings Data file
runid is the date and time stamp for the run
N1 is the number of the current realization being processed
Num is the number of analyte names following
Analyte(1) is the name of the first analyte (see Section 2.1.2)
Analyte(num) is the name of the last analyte.

The ESD file name may contain path information. The runid parameter is composed of year, month, day, hour (using a 24 hour clock), minute, and second for the run. For example, a run on May 21, 2001 at 20 seconds after 1:05 PM would have a runid entry of 20010521130520.

If no value is given for Num, then all analytes in the ESD keyword file with modifier COMPUTE will be included in the analysis. However, the typical SAC directory structure prohibits analysis of more than one analyte at a time.

An example invocation of CRDROP for realization 12 of an I129 analysis in a Windows environment would be the following:

```
CRDROP  “D:\Test2\ESD_Test2.key”  “20020315131005”  12  1  “I129”
```

8.3.3 Code Control and Keyword Descriptions

Parameters controlling the operation of the code are provided in two keyword files: crdrop.key and the ESD keyword file (the ESD keyword file name is provided on the CRDROP command line). Section 2.1 describes the keywords provided in the ESD keyword file. The following sections describe the keywords used in the crdrop.key file. Table 8.1 lists the keywords used by the CRDROP code. The TITLE and USER keywords are required and the run will error terminate if non-blank quote strings are not provided for either keyword.

Table 8.1 Summary of Keywords Used by the CRDROP Code

Keyword	**Source**	**Information Used by CRDROP**
ANALYTE	ESD file, or	Analyte selection using modifiers ID, TYPE, and COMPUTE.
	Command line	Analyte selection using name(s) provided on the command line.
END	ESD file and CRDROP.KEY	Signifies end of a keyword file.
EXECUTE	CRDROP.KEY	Cause complete execution of CRDROP.
FILE	ESD file	Necessary path and file names using modifiers I_HUMAN, I_ECOLOG, I_ECONOM, I_CULTUR, and ANALYTE.
FILE	CRDROP.KEY	Summary report file name and pathway to the river model output files.
REALIZATION	ESD file	Number of realizations in analysis.

Keyword	Source	Information Used by CRDROP
REPORT	CRDROP.KEY	Name of output report for CRDROP run summary and error messages.
TITLE	CRDROP.KEY	Title of crdrop.key keyword input file.
USER	CRDROP.KEY	User responsible for developing the crdrop.key keyword input file.

8.3.3.1 END Keyword for CRDROP

The END keyword signifies the end of all keyword data. It should be the last keyword in the keyword file. Any data in the keyword file after the END keyword will be ignored. The syntax for this keyword record is the following:

```
END
```

There are no modifiers, numerical values, or quote strings associated with the END keyword.

8.3.3.2 EXECUTE Keyword for CRDROP

The EXECUTE keyword (if present) causes a full analysis to be attempted. If the keyword is not present, then the analysis will terminate after reading the input data and performing checks on the data. Running the code without the EXECUTE record is normally performed only to test the input data set. Once the data has been checked, the EXECUTE record should be present. When the CRDROP code is exercised under the control of the ESP, the EXECUTE record must be present or the analysis will not proceed correctly. An example EXECUTE keyword record is as follows:

```
EXECUTE
```

There are no modifiers or quote strings associated with the END keyword.

8.3.3.3 FILE Keyword for CRDROP

This keyword is used to specify the file name of the location cross-index file and the path to the output files for the river model. Modifiers are used to specify which path/file is being defined and quote strings (up to 72 characters) are used to specify the path/file name. The syntax for the FILE keyword record is the following:

```
FILE [MASS2=“quote 1” | INDEX_CR=“quote 1”]
```

The modifier MASS2 indicates the directory path (no file name) to the river model output files. The actual river model output file names are generated during execution of CRDROP (from the analyte and realization being analyzed) and are not included in the quote string. The modifier INDEX_CR indicates the path/file name for the location cross-index file. Both modifiers may be included on the same FILE keyword record. An example keyword record is as follows:

```
FILE MASS2=”F:\test\river\” INDEX_CR “CrossIndex.grd”
```

There are no numerical values associated with the FILE keyword.

8.3.3.4 REPORT Keyword for CRDROP

The REPORT keyword provides the file name for writing summary information and error messages. The REPORT keyword must be the first record of the crdrop.key file, or the run will error terminate. The syntax for this record is as follows:

```
REPORT [“quote1”]
```

where quote1 is a string (up to 72 characters) containing the path/file name for the summary output report.

There are no modifiers or numerical values associated with the REPORT keyword record.

8.3.3.5 TITLE Keyword for CRDROP

The TITLE keyword is used to provide a descriptive title that will be written to the output file specified for summary information in the REPORT keyword record. If the title is not supplied, the program will error terminate. The syntax for this keyword record is the following:

```
TITLE ["quote"]
```

The title is entered in a quote string, which must be enclosed in double quotes. Titles up to 72 characters long are supported. The following example defines a title for a run of the code.

```
TITLE "Example title line for the CRDROP.KEY keyword control file."
```

There are no modifiers or numerical values associated with the TITLE keyword.

8.3.3.6 USER Keyword for CRDROP

This keyword is used to identify the individual responsible for preparing input in the keyword file. If the user name is not supplied the program will error terminate. The user name is written to the output file specified for summary information in the REPORT keyword record. Syntax of the USER keyword record is the following:

```
USER [“user name”]
```

Where the “user name” is a string of length up to 20 characters. An example USER keyword is the following:

```
USER “John Doe”
```

There are no modifiers or numerical values associated with the USER keyword.

8.4 Data Files

The CRDROP code reads from two keyword files, a location cross-index file, river model output files, and the ECDA files. Summary information is written to an output file and river related concentration data is written to the ECDA files. Table 8.2 provides a summary of the files accessed by the CRDROP code.

Table 8.2 Files Accessed by the CRDROP Code

File	Use	Description
CRDROP keyword file	Input	Primary source of run control information, file names.
ESD keyword file	Input	Source of general information for the overall analysis.
Location cross-index file	Input	Provides location cross-index values between the river model and the impacts models.
River model result files	Input	Provides concentration values generated by the river model.
CRDROP summary file	Output	Summary output information for the CRDROP analysis, including error messages, if any.
ECDA files	Input/output	Environmental Concentration Data Accumulator files, opened and modified by the CRDROP code.

8.4.1 Input Files

The following sections describe details of files providing input to the CRDROP code.

8.4.1.1 Data Provided in the CRDROP Keyword File

The CRDROP keyword file provides file names for selected input and output files. Section 8.3.3 defines the keywords for this file. The general syntax of keywords is described in Section 10.0.

The first keyword record in the file must be the REPORT keyword, which provides the name of the file to be used for saving information on the analysis scope and any error messages generated by CRDROP. If the first record is not the REPORT keyword, then the analysis will be terminated. The TITLE and USER keyword records must also be present (anywhere after the REPORT keyword) or the code will error terminate.

8.4.1.2 Data Read from the ESD file by the CRDROP Code

The CRDROP code reads control information from the ESD keyword file to determine the analytes to include, the names of the ECDA files to access, and the number of realizations allowed for the analysis. The input is read using the keyword format syntax as described in Section 10.0. The ESD keywords used by CRDROP are ANALYTE, FILE, OS, REALIZAT, and END. Section 2.1 describes these keywords.

8.4.1.3 Description of the Location Cross-Index File

The location cross-index file provides predetermined index values that correlate the river model position index values with the impacts code locations. The file may be generated by use of the pre-processor CRDROP_INDEX described in Section 8.5. The file has the information and format indicated in Table 8.3.

Table 8.3 Content and Format of the Location Cross-Index File

Record	**Information**	**Description**
First record in file	Descriptive title for file	Character string up to 72 characters.
	Number of impact groups included in the file	Integer number, maximum of four (human, ecological, economic, and cultural.
For each impact type:		
First record of data set for impact	Name of impact	String (10 characters maximum) giving the name of the impact type to consider. Allowed names are “HUMAN”, “ECOLOGICAL”, ”ECONOMIC”, and “CULTURAL.”
	Number of locations	Integer, number of location to be evaluated for the current impact type.
Remaining records for data set for impact	Name of impact location	String (six characters) giving the name of the location for the current impact type.
	River location index	Integer, index of river location that corresponds to the current impact location.

Table 8.4 shows an example cross-index file.

Table 8.4 Example Cross-Index Output File

```
"Test keyword file for INDEX_CRDROP",  4," Impact types"
"HUMAN",   6," locations for this impact type"
"H001  ",      0,       0.0,       0.0,
"H002  ",      7,  654818.9,  145789.7,
"H003  ",      0,       0.0,       0.0,
"H004  ",     26,  655018.9,  145789.7,
"H005  ",      0,       0.0,       0.0,
"H006  ",     48,  655218.9,  145789.7,
"ECOLOGICAL",   8," locations for this impact type"
"EL01  ",    128,  655517.9,  145185.2,
"EL02  ",      0,       0.0,       0.0,
"EL03  ",      0,       0.0,       0.0,
"EL04  ",      0,       0.0,       0.0,
"EL05  ",     81,  655117.9,  145185.2,
"EL06  ",     97,  655012.3,  144831.1,
"EL07  ",     65,  654917.9,  145185.2,
"EL08  ",      0,       0.0,       0.0,
"ECONOMIC",  10," locations for this impact type"
"EC01  ",      0,       0.0,       0.0,
"EC02  ",      0,       0.0,       0.0,
"EC03  ",      0,       0.0,       0.0,
"EC04  ",   1045,  662718.6,  144230.1,
"EC05  ",      0,       0.0,       0.0,
"EC06  ",   1047,  662618.6,  144230.1,
"EC07  ",   1046,  662605.4,  144210.3,
"EC08  ",      0,       0.0,       0.0,
"EC09  ",   1030,  662585.4,  144210.3,
"EC10  ",      0,       0.0,       0.0,
"CULTURAL",   7," locations for this impact type"
"C001  ",      0,       0.0,       0.0,
"C002  ",      0,       0.0,       0.0,
"C003  ",   1558,  666647.7,  145006.1,
"C004  ",      0,       0.0,       0.0,
"C005  ",   1559,  666627.7,  145006.1,
"C006  ",   1561,  666619.5,  145040.6,
"C007  ",   1561,  666609.5,  145040.6,
```

The cross-index file prepared by the CRDROP_INDEX code also contains the x and y coordinate value for the river location in addition to the river location index values (following the river location index on the last line described in Table 8.3). However, this information is not read or used by the CRDROP code.

8.4.1.4 Description of MASS2 Output Files Accessed by the CRDROP Code

The river model generates files of concentration parameters at each location defined in the river model grid system. One file is generated for each combination of analyte and realization. When CRDROP is run, all analyte files for one realization are accessed.

Each river model output file contains data for one combination of analyte and realization. These files have names coded to the analyte and realization associated with the data contained in the file. The file names are constructed as follows.

```
“analyte”_”realization”.riv
```

where “analyte” is the ID of the analyte, “realization” is a 4-character string giving the realization number with zero fill in front of the number, and riv is the file extension. The following are example file names:

```
   H3_0001.riv
   U238_0020.riv
   CRVI_0002.riv
```

The content of the river model output file is organized in groups by time periods (years). Within each time data set, concentration data is provided for three media for all MASS2 river locations. The table below describes the file format and content of MASS2 river model output files.

Table 8.5 Content of MASS2 River Model Output Files

Record	Parameter	Type	Description
1	Title	String	Descriptive title for the file.
	Analyte name	String	Name of analyte for which data is provided in this file.
	Number of time periods	Integer	Number of time period for which data is provided in this file.
	Number of time locations	Integer	Number of location points to be supplied for each time period in this file.
2	Title for current time	String	Descriptive title for the current time period.
	Time	Integer	Year of the current time data set.
3	Concentration 1	Real number	Average annual surface water analyte concentration for current year and location.
	Concentration 2	Real number	Average annual pore water analyte concentration for current year and location.
	Concentration 3	Real number	Average annual bottom sediment analyte concentration for current year and location.

Table 8.6 provides an example partial river output file for the analyte Tc99. The units of the concentration parameters are as evaluated by the MASS2 river model. CRDROP makes no modifications to the numerical values in transferring to the ECDA files.

Table 8.6 Excerpted Records from a MASS2 River File

```
"MASS2 Output", "Tc99", 1107, 2368
"Time 1", 1945
0.0000e+00 0.0000e+00 0.0000e+00
1.0000e-03 1.0000e-03 1.0000e-03
2.0000e-03 2.0000e-03 2.0000e-03
3.0000e-03 3.0000e-03 3.0000e-03
4.0000e-03 4.0000e-03 4.0000e-03
5.0000e-03 5.0000e-03 5.0000e-03
6.0000e-03 6.0000e-03 6.0000e-03
7.0000e-03 7.0000e-03 7.0000e-03
8.0000e-03 8.0000e-03 8.0000e-03
9.0000e-03 9.0000e-03 9.0000e-03
1.0000e-02 1.0000e-02 1.0000e-02
1.1000e-02 1.1000e-02 1.1000e-02
1.2000e-02 1.2000e-02 1.2000e-02
… (lines deleted)
"Time 2", 1946
2.8568e+01 2.8568e+01 2.8568e+01
2.8569e+01 2.8569e+01 2.8569e+01
2.8570e+01 2.8570e+01 2.8570e+01
2.8571e+01 2.8571e+01 2.8571e+01
2.8572e+01 2.8572e+01 2.8572e+01
2.8573e+01 2.8573e+01 2.8573e+01
2.8574e+01 2.8574e+01 2.8574e+01
```

```
2.8575e+01 2.8575e+01 2.8575e+01
2.8576e+01 2.8576e+01 2.8576e+01
2.8577e+01 2.8577e+01 2.8577e+01
2.8578e+01 2.8578e+01 2.8578e+01
2.8579e+01 2.8579e+01 2.8579e+01
2.8580e+01 2.8580e+01 2.8580e+01
2.8581e+01 2.8581e+01 2.8581e+01
2.8582e+01 2.8582e+01 2.8582e+01
2.8583e+01 2.8583e+01 2.8583e+01
2.8584e+01 2.8584e+01 2.8584e+01
… (lines deleted)
```

8.4.2 Output Files

Details of files modified or generated by the CRDROP code are described in the following sections.

8.4.2.1 CRDROP Report File

The summary output report contains a heading with the run time, title, and user name from the crdrop.key and ESD files, and a summary of the processing that was performed. Any error messages generated during the run are also written to this file. The file is not used by other SAC codes and is generated only as a record of the analysis and for review of error messages to recover from abnormal situations. Table 8.7 provides an example summary output file for CRDROP.

Table 8.7 Example Summary Output File for CRDROP

```
               CCCCC  RRRRR   DDDDDD  RRRRR     OOO   PPPPP
              C     C R    R  D     D R    R   O   O  P    P
              C       R    R  D     D R    R  O     O P    P
              C       RRRRR   D     D RRRRR   O     O PPPPP
              C       R  R    D     D R  R    O     O P
              C     C R   R   D     D R   R    O   O  P
               CCCCC  R    R  DDDDDD  R    R    OOO   P


               CRDROP       Version 1.7
               Last Modified on 12 Dec 2001
       -----------------------------------------------
        Columbia River Data Restructure Other Programs
                  SAC Rev. 0 Data Processing
       ----------------------------------------------
           Developed By Battelle Memorial Institute
            Pacific Northwest National Laboratory
                     Richland, Washington

Current Run ID = 200203271614   User Name = Engel, Eslinger,
     System Date = 03-28-20   System Time = 00:50:00


 The software used to generate this output is experimental
    and has not been formally tested or peer reviewed.
```

```
                             Review Signatures

Input Prepared By: ______________________________       Date: _______________

Input Reviewed By: ______________________________       Date: _______________


======================== Echo of the Problem Definition ========================

Title: SAC Rev. 0 Median2 Assessment
User:  Engel, Eslinger,
Current realization is    1 of    1 total realizations
Operating System is: UNIX,  Delimiter is: /
Number of analytes specified on the command line:   1
 Analyte names: Tc99

Processing based on keyword data
  File = crdrop.key
 Allocate space using FIRST_ALLLOCATE
 Read crdrop.key file to get analyte names to be included, NANALYTES = 10
 After Read crdrop.key file to get analyte names to be included, NANALYTES = 1
 Read and process map information for Human Impact type
 Read and process map information for Ecological Impact type
 Read and process map information for Economic Impact type
 Read and process map information for Cultural Impact type
 Read cross-index values for MASS2 grid locations
 Start analyte loop for NANALYTES:  1
  Opening MASS2 output file: Tc99_0001.RIV
  Starting to read MASS2 output file
 Finished current analyte number:  1
 CRDROP Normal Termination
```

8.4.2.2 Modifications Made to the ECDA Files

The ECDA files contain concentrations of analytes at specific locations in the environment. The CRDROP program updates these concentrations for the analytes and realization (input from the command line) at the selected river locations defined for each impact type. The concentrations are extracted from the river model output files and written to the ECDA files.

The order of the modifications is controlled by the information read from the MASS2 river model output files. Data is read for one analyte (per file) and for one year at a time, with media concentrations read for all river locations. If the time is associated with an impact type (i.e., human, ecological, economic, or cultural), then the river model concentration data is read and saved by CRDROP. The location cross-index information (from the location cross-index file) is used to extract the concentration data for the locations needed by the ECDA impact type. For each location and media, an ECDA file is read and media data for the analyte, time, and all realizations are extracted in one array. The concentration data for the associated time, location, and media are written into the extracted array, and the array is re-written to the ECDA file. This process is repeated for all locations and media for the current analyte and time.

8.5 CRDROP_INDEX Utility Program

8.5.1 Code Purpose

The purpose of the CRDROP_INDEX code is to generate the location cross-index file used by the CRDROP code. This file provides information that correlates the river model position index values with the impact locations. By pre-calculating these position index values, considerable time is saved during the execution of the CRDROP program.

8.5.2 Algorithms and Assumptions

The CRDROP_INDEX code reads the MASS2 river grid locations and the location coordinate data for each impact type and then processes each impact location. The processing involves finding the river grid location that is the nearest distance to the given impact location. When all impact location values have been processed, the array of MASS2 grid index values is written to the location cross-index file for use by the CRDROP program.

The distance between a river location (point 1) and an impact location (point 2) is evaluated using the following algorithm, based on the Pythagorean theorem.

$$z = \sqrt{(X_2 - X_1)(X_2 - X_1) + (Y_2 - Y_1)(Y_2 - Y_1)}$$

where:

z = distance between river location (X_1,Y_1) and impact location (X_2,Y_2) (meters)
X_1 = x-coordinate of the river location (meters)
Y_1 = y-coordinate of the river location (meters)
X_2 = x-coordinate of the impact location (meters)
Y_2 = y-coordinate of the impact location (meters).

The index of the river location point corresponding to the minimum value of z is saved to the cross-index grid output file. The process is performed for all impact types defined in the ESD file (see Section 2.1) for the analysis.

If the value of z is greater than the distance limit criterion value (DIST_LIMIT), then an error message is written for the impact location coordinate. All impact location coordinates are tested before an error termination occurs so multiple errors can be detected and reported in one analysis.

8.5.3 Code Environment

The CRDROP code is intended to be run once for each SAC analysis. It can be run in a standalone mode or under the control of the SAC ESP.

8.5.3.1 Location in Processing Sequence

The CRDROP_INDEX code can be run prior to initiation of the overall analysis. However, it must have the MASS2 (river model) grid file available. CRDROP_INDEX must be run to establish a cross-index file giving the river location that corresponds to each impact location (for human, ecological, economic, and cultural impact locations) prior to running the CRDROP code. Staff members responsible for implementation of the river model also prepare the river model grid file.

8.5.3.2 How the CRDROP_INDEX Code Is Invoked

CRDROP_INDEX may be used in either Microsoft Windows or Linux environments, provided the code has been compiled by an appropriate Fortran 95 compiler for the environment. In the Microsoft Windows environment, CRDROP_INDEX is executed by the following DOS command:

```
CRDROP_INDEX fntmp runid
```

In the Linux environment, CRDROP_INDEX is executed through the following Bourne Shell or C Shell commands:

```
crdrop_index.exe fntmp runid
```

where fntpm is the path and file name for the ESD file, and runid is the date/time stamp for the current analysis. The fntmp path is used as the base directory for the analysis. The path is extracted from this character string and used to write the temporary run file and the files crdrop_index.done or crdrop_index.fail files. The runid parameter is composed of year, month, day, hour (using a 24 hour clock), minute, and second for the run. For example, a run on May 21, 2001 at 20 seconds after 1:05 PM would have a runid entry of 20010521130520.

8.5.4 Keyword Descriptions for CRDROP_INDEX

The keywords used to control the CRDROP_INDEX code are always read from the file named index.key. The following sections describe these keywords.

8.5.4.1 END Keyword for CRDROP_INDEX

The END keyword signifies the end of all keyword data. It should be the last keyword in the keyword file. Any data in the keyword file after the END keyword will be ignored. The syntax for this keyword record is the following:

```
END
```

There are no modifiers, numerical values, or quote strings associated with the END keyword.

8.5.4.2 EXECUTE Keyword for CRDROP_INDEX

The EXECUTE keyword (if present) causes a full analysis to be attempted. If the keyword is not present, then the analysis will terminate after reading the input data and performing checks on the data. Running the code without the EXECUTE record is normally performed only to test the input data set. Once the data has been checked, the EXECUTE record should be present. When the CRDROP_INDEX code is exercised under the control of the ESP, the EXECUTE record must be present or the analysis will not proceed correctly. An example EXECUTE keyword record is as follows:

```
EXECUTE
```

There are no modifiers, quote strings, or numerical values associated with the REPORT keyword record.

8.5.4.3 DISTANCE Keyword for CRDROP_INDEX

The DISTANCE keyword defines the maximum distance allowed between the impact location and the nearest river grid point. After the nearest location is identified, the distance between the two points is evaluated and compared to the DISTANCE criteria value. If the distance is greater than the criteria an error message is written and abnormal termination of the analysis is indicated in the output file. An example DISTANCE keyword record is as follows:

```
DISTANCE 1000
```

This keyword indicates a distance criterion of 1000 meters is to be used in the analysis. There are no modifiers or quote strings associated with the DISTANCE keyword.

8.5.4.4 FILE Keyword for CRDROP_INDEX

This keyword is used to specify the file name of the river model location index file and the location cross-index output file. Modifiers are used to specify which path/file is being defined and quote strings (up to 72 characters) are used to specify the path/file name. The syntax for the FILE keyword record is the following:

FILE {MASS2GRI=“quote$_1$”} {INDEX_CR=“quote$_2$”}

The modifier MASS2GRI indicates that the associated quote string is the directory path (no file name) to the river model grid file. The actual river model output file names are generated during execution of CRDROP (from the analyte and realization being analyzed) and are not included in the quote string. The modifier INDEX_CR indicates that the associated quote string is the path (including the file name) for the location cross-index file. Both modifiers may be included on the same FILE keyword or may be entered on separate keywords. An example keyword record is as follows:

```
FILE  MASS2GRI=”F:\test\river\”  INDEX_CR=“C:\Junk\CrossIndex.grd”
```

There are no numerical values associated with the FILE keyword.

8.5.4.5 REPORT Keyword for CRDROP_INDEX

The REPORT keyword provides the file name for writing summary information and error messages. The REPORT keyword must be the first record of the index.key file, or the run will error terminate. The syntax for this record is as follows:

```
REPORT “quote1”
```

where quote$_1$ is a string (up to 72 characters) containing the path and file name for the summary output report. There are no modifiers or numerical values associated with the REPORT keyword.

8.5.4.6 TITLE Keyword for CRDROP_INDEX

The TITLE keyword is used to provide a descriptive file title for the INDEX.KEY file. The problem title will be written to the output file specified for summary information in the REPORT keyword record. If the title is not supplied, the program will error terminate. The syntax for this keyword record is the following:

```
TITLE ["quote"]
```

The title is entered in a quote string, which must be enclosed in double quotes. Titles up to 72 characters long are supported. The following example defines a title for a run of the code.

```
TITLE "Example title line for the CRDROP.KEY keyword control file."
```

There are no modifiers or numerical values associated with the TITLE keyword.

8.5.4.7 USER Keyword for CRDROP_INDEX

This keyword is used to identify the individual responsible for preparing input in the keyword file. If the user name is not supplied the program will error terminate. The user name is written to the output file specified for summary information in the REPORT keyword record. Syntax of the USER keyword record is the following:

```
USER [“user name”]
```

where the “user name” is a string of length up to 20 characters. An example USER keyword is the following:

```
USER “John Doe”
```

There are no modifiers or numerical values associated with the USER keyword.

8.5.5 Data Files

The CRDROP_INDEX code reads two keyword files, a river model grid file, and the ECDA map files. Summary information is written to an output file and an impact location cross-index file. Table 8.8 summarizes the files accessed by the CRDROP code.

Table 8.8 Files Accessed by the CRDROP_INDEX Code

File	Use	Description
CRDROP_INDEX keyword file	Input	Primary source of run control information, file names.
ESD keyword file	Input	Source of general information for the overall analysis, including location coordinates.
ECDA Map Files	Input	Provides location names for each impact model to be included in the analysis.
River model grid point locations file	Input	Provides coordinates of all grid points for which concentration values are generated by the river model.
CRDROP_INDEX run summary file	Output	Summary output information for the CRDROP_INDEX analysis, including error messages, if any (name crdrop_index.run).
CRDROP_INDEX.DONE	Output	File is written to the base directory if the program completed successfully.
CRDROP_INDEX.FAIL	Output	File is written to the base directory if the program did not complete successfully.

8.5.5.1 Input Files

The purpose of files providing input to the CRDROP_INDEX code are described as follows:

- **ECDA Map Files:** ECDA map files are read to provide the location names for each of the impact endpoints (human, ecological, economic, and cultural). The location names are used to extract the location coordinates from the ESD file. Section 2.2.2 describes the format of this file.
- **Environmental Setting Data File:** The ESD keyword file is read to obtain the location coordinates for all impact locations to be included in the analysis. All locations provided in the ESD keyword file will be included, and cross-index values will be determined between these locations and the river grid points. Keywords from the ESD keyword file read by CRDROP_INDEX are the FILE, LOCATION, and END keywords. Section 2.1 defines the syntax for these keywords.
- **Keyword Input File**: The run control information is read from a file named INDEX.KEY, which is a keyword-input file specific to the CRDROP_INDEX program. Section 8.5.4 describes the keyword records.

- **River Model Grid Location Coordinate File:** The river model grid location coordinate file provides coordinate values for each grid location used by the river model. The coordinates are read by the CRDROP_INDEX code and used to generate cross-index values for the impact locations. The first record in a river model grid file contains a title, the number of grid locations in the file, and the units of the x,y coordinate values. The title is not used. The number of grid locations is used to read and save the x,y coordinate values. The units string must be METERS or the analysis will not continue. The remaining records all have the same format: one pair of x,y coordinates. The units for the coordinates are meters. Table 8.9 provides the first few lines of an example of a river model grid location file. The files typically contain several thousand records.

Table 8.9 Excerpted Records from a River Model Grid Location File

```
"MASS2 Grid Points", 28568, METERS
   2147805.1     477754.6
   2147856.7     477793.4
   2147968.1     477875.8
   2148092.2     477965.8
   2148222.8     478059.7
   2148355.9     478155.7
```

8.5.5.2 Output Files

Report File: The report file contains summary information on the analysis performed by CRDROP_INDEX. If an abnormal termination has occurred, this file should be reviewed for error messages to determine the cause of the abnormal termination.

Table 8.10 Example Report file from CRDROP_INDEX

```
 CCCC   RRRR   DDDDD   RRRR     OOO    PPPP      III  N     N  DDDDD  EEEEE X    X
C    C  R   R  D    D  R   R   O   O   P   P      I   NN    N  D    D E      X  X
C       R   R  D    D  R   R  O     O  P   P      I   N N   N  D    D E       XX
C       RRRR   D    D  RRRR   O     O  PPPP       I   N  N  N  D    D EEEE    XX
C       R R    D    D  R R    O     O  P          I   N   N N  D    D E       XX
C    C  R  R   D    D  R  R    O   O   P          I   N    NN  D    D E      X  X
 CCCC   R   R  DDDDD   R   R    OOO    P   ===== III  N     N  DDDDD  EEEEE X    X


                       CRDROP_INDEX   Version 1.00
                       Last Modified on 10 December
               -----------------------------------------------
                Columbia River Data Restructure Other Programs
                        Location Cross-Index Generation
                          SAC Rev. 0 Data Processing
               ----------------------------------------------
                   Developed By Battelle Memorial Institute
                    Pacific Northwest National Laboratory
                             Richland, Washington

        Current Run ID = 200203271525   User Name = Dennis Strenge
             System Date = 03-27-20   System Time = 15:25:23


         The software used to generate this output is experimental
            and has not been formally tested or peer reviewed.
```

```

                              Review Signatures

Input Prepared By: ______________________________       Date: _______________

Input Reviewed By: ______________________________       Date: _______________


======================== Echo of the Problem Definition ========================

Title: Test keyword file for INDEX_CRDROP
User:  Dennis Strenge
Operating System: UNIX,  Delimiter is: /

 Impact parameters are set as follows:
      Impact Type  Flag Number of Locations File Name
      -----------  ---- ------------------- -------------
      Human         T       1930            /home/ANALYSIS/Median2/Bg/ecda/HUMA_Median2_Bg.Map
      Ecological    T        674            /home/ANALYSIS/Median2/Bg/ecda/ECOL_Median2_Bg.Map
      Economic      T         13            /home/ANALYSIS/Median2/Bg/ecda/ECON_Median2_Bg.Map
      Cultural      T       3981            /home/ANALYSIS/Median2/Bg/ecda/CULT_Median2_Bg.Map

 MASS2 grid file name:         CRDROP_grid.dat
 Cross-index output file name: CrossIndex.grd

Processing based on keyword data File = index.key
  crdrop_index Normal Termination
```

Location Cross-Index File: The location cross-index file provides predetermined index values that correlate the river model position index values with the impacts code locations. The file has the information and format indicated in Table 8.3. Table 8.4 provides an example cross-index file. The name of the output file is specified on the index.key keyword record FILE GRID.

Temporary Run File: The file CRDROP_INDEX.RUN is opened in the base directory to show the ESP program that the CRDROP_INDEX program is running. When the run is complete, this file is deleted to show that the run is finished. This feature is useful only when the CRDROP_INDEX code is run under the control of the SAC ESP program. The normal operating mode is to run the CRDROP_INDEX code prior to initializing the analysis under control of the ESP program.

Run Completion Files: At the completion of the run, a file is opened and saved in the base directory indicating the status of the run. If the run completed without errors, the file crdrop_index.done is written; otherwise the file crdorp_index.fail is written. The file crdrop_index.done has two lines: the first is a zero (indicating no errors), and the second is the run id parameter, which contains the computer generated date and time stamp for the run. If the run failed, the crdrop_index.fail file is generated with a value of 101 on the first line and the same run id on the second line.

9.0 RIPSAC – Riparian Zone Module

9.1 Purpose

The riparian zone module (RIPSAC) calculates the concentrations of analytes in seep water and soil. The riparian zone model algorithms apply only in the region along the edge of the river where there is significant interaction between groundwater and river water. RIPSAC reads analyte concentrations in groundwater and surface water from binary concentration data files (see Section 2.2.1), calculates the concentrations for seep water and soil, and writes the calculated concentrations back into the same files.

9.2 Mathematical Models for RIPSAC

Concentrations of analytes in seep water (groundwater seepage face boundaries flowing to the river) are calculated from the concentration in groundwater using the following equation:

$$C_{seep} = (Df_{seep} C_{gw}) + (1 - Df_{seep}) C_{river}$$

where:

C_{seep} = Concentration of an analyte in seep water (Ci/m^3 or Kg/m^3)
C_{gw} = Concentration of an analyte in ground water (Ci/m^3 or Kg/m^3)
C_{river} = Concentration of an analyte in river (surface) water (Ci/m^3 or Kg/m^3)
Df_{seep} = Dilution factor controlling the mixing of groundwater and river water (unitless).

Concentrations of analytes in the upper layer of riparian zone soil are calculated from the concentration in seep water using the following equation:

$$C_{soil} = C_{seep} Kd_{soil}$$

where:

C_{soil} = Concentration of an analyte in soil (Ci/kg soil or kg/kg soil)
C_{seep} = Concentration of an analyte in seep water (Ci/m^3 or kg/m^3)
Kd_{soil} = Partition coefficient for the analyte between water and soil (m^3/kg).

9.3 Code Execution Environment

RIPSAC was developed as a program to be invoked by the master SAC control software, SAC ESP. However, in practice, it is generally invoked in a standalone mode.

9.3.1 Location in the Processing Sequence

The general processing sequence for SAC environmental runs is outlined in Figure 1-2. As noted in the figure, RIPSAC is the last step in the environmental transport sequence and operates using internal realization looping. Thus, only one run of RIPSAC is required to process all realizations for analytes and all impacts. RIPSAC also has the option to run on a subset of data for any combination of realizations, analytes, and impact types.

9.3.2 How the Code Is Invoked

RIPSAC may be used in either Microsoft Windows or Linux environments, provided that the code has been compiled by a Fortran 95 compiler appropriate for the environment. In the Microsoft Windows environment RIPSAC is executed either of the following DOS commands:

```
RIPSAC “Keyword File”
RIPSAC
```

In the Linux environment RIPSAC is executed through either of the following Bourne Shell or C Shell commands:

```
ripsac.exe “keyword file”
ripsac.exe
```

In the first example for both operating systems, RIPSAC is invoked and assumes the file identified by “keyword file” (file names are case sensitive under Linux) contains the controlling keywords for RIPSAC. In the second example, the keyword file name is missing from the command line, thus RIPSAC queries the user for the name of the file. In either example, if the keyword file is not found, RIPSAC writes an error message and terminates execution.

9.4 Keyword Descriptions for RIPSAC

In general, the keywords for RIPSAC can be entered in any order. The only restrictions on keyword order are the following:

- The REPORT keyword must be the first keyword in the file.
- The END keyword must be the last keyword in the file.

In the following keyword descriptions, some data are optional and some are required. Data that are required are enclosed in square brackets. For AB to be required, it would be denoted by [AB]. If only one of the three items AB, BC, CD were required, it would be written as [AB|BC|CD]. The vertical bars indicate that the user must select one of the items in the list. Optional items are enclosed in normal brackets. For DE to be an optional entry, it would be denoted by {DE}. The { } or [ ] symbols are indicators of whether the data are required, the symbols do not need to be entered when the keyword is constructed. The keyword name can contain any number of characters, however, only the first 8 characters are utilized (for example, REALIZAT = REALIZATION = REALIZATxxx). Section 10.0 further describes the general syntax for keywords. Table 9.1 summarizes the keywords used by RIPSAC.

Table 9.1 Summary of Keywords Used by RIPSAC

Keyword	Description
ANALYTE	The ANALYTE keyword defines the analytes to be used in a run of RIPSC.
DEBUG	The DEBUG keyword is used to activate dumping of intermediate calculations to the report file.

END	The END keyword signifies the end of all keyword data.
FILE	The FILE keyword is used to enter the name of the ESD keyword file.
IMPACTS	The IMPACTS keyword is used to activate calculations for the near-river-shore environment for a given impact type.
KDSOIL	The KDSOIL keyword is used to assign soil-water partition coefficients to locations for use in the soil concentration calculations.
LOCATION	The LOCATION keyword is used to associate groundwater and surface water concentrations with dilution factors in order to complete soil and seep water concentration calculations.
REALIZATION	The REALIZATION keyword is used to define the realizations to be processed in the RIPSAC code.
REPORT	The REPORT keyword is used to define the name of the output report (log) file.
TITLE	The TITLE keyword is used to define a single-line problem title.
USER	The USER keyword is used to identify the user of the program.

9.4.1 ANALYTE Keyword for RIPSAC

The ANALYTE keyword is used to define the analytes to be used in the simulation. The syntax for this keyword is

```
ANALYTE ID=“quote1”
```

The analytes requested must be a subset of the analytes for which environmental data were computed and stored by the inventory, release, and transport modules. A separate ANALYTE keyword must be entered for every analyte. An example entry for a radioactive analyte is the following:

```
ANALYTE ID=“Np237"
```

9.4.2 DEBUG Keyword for RIPSAC

The DEBUG keyword is used to activate dumping of intermediate calculations to the report file. It should be used sparingly and with only one or two realizations, otherwise the volume of output could fill the user’s hard drive. The syntax for this keyword record is the following:

```
DEBUG
```

There are no modifiers or quote strings associated with the DEBUG keyword.

9.4.3 END Keyword for RIPSAC

The END keyword signifies the end of all keyword data. It should be the last keyword in the keyword file. All data in the keyword file after the END keyword will be ignored. The syntax for this keyword record is the following:

```
END
```

There are no modifiers or quote strings associated with the END keyword.

9.4.4 FILE Keyword for RIPSAC

The FILE keyword is used to enter the name of the ESD keyword file that defines the environmental release and transport scenario for which impacts are to be calculated. The names of the files providing concentration data are contained in the ESD keyword file rather than in the RIPSAC keyword file. The syntax for the FILE keyword is the following:

```
FILE [ESD="quote1"]
```

The file name must be entered in a quote string, which must be enclosed in double quotation marks (" "). Path names up to 72 characters long are supported. An example entry is the following:

```
FILE ESD "G:\RIPSAC\Tests\Test_Case_12.Key"
```

9.4.5 IMPACTS Keyword for RIPSAC

The IMPACTS keyword is used to activate calculations for the near-river-shore environment for a given impact type. The syntax for this keyword record is the following:

```
IMPACTS {HUMAN} {ECONOMIC} {ECOLOGIC} {CULTURAL}
```

Multiple IMPACTS cards can be entered with combinations of modifiers, or a single card can be entered containing all of the modifiers. The effect of entering multiple IMPACTS keywords is cumulative. The modifiers can be entered in any order. Table 9.2 describes the modifiers associated with the IMPACTS keyword. Even though all four modifiers are identified as being optional, no calculations will be performed unless at least one modifier is entered.

Table 9.2 Modifiers Associated with the IMPACTS Keyword in RIPSAC

Modifier	Description
HUMAN	Compute concentrations for human impact locations in the riparian zone.
ECONOMIC	Compute concentrations for economic impact locations in the riparian zone.
ECOLOGIC	Compute concentrations for ecological impact locations in the riparian zone.
CULTURAL	Compute concentrations for cultural impact locations in the riparian zone.

The following example keyword would invoke calculations for three impacts types.

```
IMPACTS HUMAN CULTURAL ECOLOGIC
```

There are no quote strings associated with the IMPACTS keyword.

9.4.6 KDSOIL Keyword for RIPSAC

The KDSOIL keyword is used to assign soil-water partition coefficients to locations for use in the soil concentration calculations. The coefficients are extracted from a library of values that are defined in the

ESD keyword file (see Section 2.3.2). The KDSOIL keywords in RIPSAC are assigned as a function of impact type, location, and analyte. The syntax for this keyword is the following:

```
   KDSOIL [HUMAN | ECOLOGIC | ECONOMIC | CULTURAL] [ANALYTE=“quote1”]
      [LOCATION=“quote2”] [KDSOIL=“quote3”]
```

Multiple KDSOIL keyword entries are required if there is more than one analyte and more than one riparian zone location. There must be a KDSOIL keyword entry for every combination of analyte and riparian zone location for an impact type where soil concentrations are to be computed. If there were 4 impact types, 10 analytes, and each impact type had 100 locations where soil concentration were desired, then 4,000 KDSOIL keyword entries would be needed.

The modifiers for the KDSOIL keyword can be entered in any order. The quote string associated with the ANALYTE modifier must contain the ID string of an analyte identified in the environmental settings definition keyword file (see Section 2.1.2). The quote string associated with the LOCATION modifier must contain the ID string of a location where a soil concentration solution has been identified in the ESD keyword file (see Section 2.1.15). The quote string associated with the KDSOIL modifier must contain the ID string of KDSOIL variable identified in the ESD keyword file (see Section 2.1.14). The impact type must be one of the four types HUMAN, ECOLOGIC, ECONOMIC, or CULTURAL.

Example keyword entries include the following:

```
   KDSOIL HUMAN ANALYTE=“Cr” LOCATION=“HL#012” KDSOIL= “KD#023”
   KDSOIL ECOLOGIC ANALYTE=“Cr” LOCATION=“HL#012” KDSOIL= “KD#003”
   KDSOIL ECONOMIC ANALYTE=“Cr” LOCATION=“HL#012” KDSOIL= “KD#033”
   KDSOIL CULTURAL ANALYTE=“Cr” LOCATION=“HL#012” KDSOIL= “KD#024”
   KDSOIL HUMAN ANALYTE=“Zn” LOCATION=“HL#016” KDSOIL= “KD#013”
   KDSOIL HUMAN ANALYTE=“U234” LOCATION=“HL#012” KDSOIL= “KD#042”
   KDSOIL HUMAN ANALYTE=“H3” LOCATION=“HL#012” KDSOIL= “KD#001”
```

9.4.7 LOCATION Keyword for RIPSAC

The LOCATION keyword is used to associate groundwater and surface water concentrations with dilution factors in order to complete soil and seep water concentration calculations. The dilution factor coefficients are extracted from a library of values that are defined in the environmental settings definition keyword file. The syntax for this keyword record is the following:

```
   LOCATION [HUMAN | ECOLOGIC | ECONOMIC | CULTURAL]
      [PRIMARY=“quote1”] [SECOND=“quote2”] [DF=“quote3”]
```

Multiple LOCATION keyword entries are required. There must be a LOCATION keyword entry for every combination of location and impact type where seep water or soil concentrations are to be computed. If there were 4 impact types and each impact type has 100 locations where seep water or soil concentrations were desired, then 400 LOCATION keyword entries would be needed.

The modifiers for this keyword can be entered in any order. The impact type must be one of the four types HUMAN, ECOLOGIC, ECONOMIC, or CULTURAL. The quote string associated with the PRIMARY modifier must contain the ID string of a location identified in the ESD keyword file for the specified impact type (see Section 2.1.15). This location must have a groundwater concentration solution associated with it. The quote string associated with the SECOND modifier must contain the ID string of a location where a surface water concentration solution for the specified impact type has been identified in the ESD keyword file. The quote string associated with the DF modifier must contain the ID string of dilution variable (see Section 2.1.7) identified in the ESD keyword file.

Example keyword entries include the following:

```
   LOCATION ECOLOGIC PRIMARY =“EL#012” SECOND =“EL#092” DF = “DF#003”
   LOCATION ECONOMIC PRIMARY =“EC#012” SECOND =“EE#015” DF = “DF#033”
   LOCATION CULTURAL PRIMARY =“CL#012” SECOND =“CL#001” DF = “DF#024”
   LOCATION HUMAN PRIMARY=“HL#012” SECOND=“HL#011” DF= “DF#023”
   LOCATION HUMAN PRIMARY =“HL#016” SECOND=“HL#011” DF = “DF#013”
   LOCATION HUMAN PRIMARY =“HL#012” SECOND=“HL#013” DF = “DF#042”
   LOCATION HUMAN PRIMARY =“HL#012” SECOND=“HL#018” DF = “DF#001”
```

9.4.8 REALIZATION Keyword for RIPSAC

The REALIZATION keyword is used to define the realizations to be processed in the RIPSAC code. The syntax for this keyword record is the following:

```
   REALIZATION [ALL | LIST N1 {N2 ... Nn} | RANGE N1 N2 ]
```

The ESD keyword file defines the total number of realizations that will be processed in the suite of environmental and impacts codes. The RIPSAC code can process all realizations at one time or it can process a subset of the realizations. Table 9.3 describes the modifiers used on the REALIZATION keyword. Only one of the modifiers is allowed on a single entry of the REALIZATION keyword. Upon entry to the code, none of the realizations are selected for processing. Multiple REALIZATION keyword entries may be used with the definition of active realizations building with each additional keyword entry.

Table 9.3 Modifiers Used With the REALIZATION Keyword in RIPSAC

Modifier	Description
ALL	The presence of this modifier will activate processing of all realizations defined in the ESD keyword file.
LIST	The LIST modifier must be accompanied by one or more realization indices. All realization indices in the list will be processed. The realization indices do not have to be entered in any particular order.
RANGE	The RANGE modifier must be accompanied by two realization indices. All realizations from the first to the second indices, inclusive of the end points, will be processed. The second index must be equal to, or greater than, the first index.

The following REALIZATION keyword in the RIPSAC keyword file would invoke processing for all realizations defined in the ESD keyword file.

```
   REALIZATION ALL
```

The following set of REALIZATION keywords assume that 100 realizations are defined in the ESD keyword file. These keywords define processing in RIPSAC for realization numbers 2, 4, 6, 12, 13, 14, 15, 99, and 100.

```
REALIZATION LIST 2 4 6 100 99
REALIZATION RANGE 12 15
```

9.4.9 REPORT Keyword for RIPSAC

The REPORT keyword is used to define the name of the output report (log) file. It must be the first keyword entered in the keyword file. The syntax for this keyword is the following:

```
REPORT ["quote"]
```

The name of the report file is entered in a quote string. File names up to 72 characters long are supported, and path names can be included. An example REPORT keyword record is the following:

```
REPORT "/SAC/SystemCodes/Cultural/Test1.rpt"
```

9.4.10 TITLE Keyword for RIPSAC

The TITLE keyword is used to define a single-line problem title. The problem title will be written to output files. If the title is not supplied the program will error terminate. The syntax for this keyword is the following:

```
TITLE ["quote"]
```

The title is entered in a quote string, which must be enclosed in double quotes. Titles up to 72 characters long are supported. The following example defines a title for a run of the code.

```
TITLE "Example title line for the cultural impacts code."
```

There are no modifiers associated with the TITLE keyword.

9.4.11 USER Keyword for RIPSAC

The USER keyword is used to identify the user of the program. The user name will be written to output files. If the user name is not supplied, the program will error terminate. The syntax for this keyword is the following:

```
USER ["quote"]
```

The user name is entered in a quote string, which must be enclosed in double quotes. User names up to 16 characters long are supported. The following example defines John Q. Public as the user running the code.

```
USER "John Q. Public”
```

There are no modifiers associated with the USER keyword.

9.5 Data Files

The RIPSAC code reads four or more input files and writes a number of output files. The number of input and output files depends on the number of contaminants being analyzed. The following sections describe these files.

9.5.1 Input Files

The input files for RIPSAC are two keyword files, a KDSOIL data file, a DILUTE data file, and a suite of concentration data files. The RIPSAC keyword file controls the case RIPSAC will execute, and it points to the ESD keyword file. The suite of concentration data files are all identified in the ESD keyword file – the user of the RIPSAC only needs to identify the ESD keyword file. The input files are the following:

- **ESD Keyword File:** The ESD keyword file contains the control information the inventory and environmental transport codes use to generate concentration data files. Section 2.1 provides definitions for these keywords. Typically, the user of the RIPSAC code will not modify the ESD keyword file.
- **ESD DILUTE Fil**e: If any seep water or soil concentration solutions are requested, then the file containing dilution factors for groundwater and river water interaction coefficients must be supplied. This file is generated by the ECDA program from DILUTE keywords contained in the ESD keyword file. Section 2.3.1 defines the file format and Table 2.12 provides an example file.
- **ESD KDSOIL Fil**e: If any soil concentration solutions are requested, then the file containing soil-water partition coefficients must be supplied. This file is generated by the ECDA program from KDSOIL keywords contained in the ESD keyword file. Section 2.3.2 defines the file format and Table 2.12 provides an example file.
- **RIPSAC Keyword File:** The RIPSAC Keyword file controls the scenario to be computed by the RIPSAC code. Section 9.4 describes the format of individual keywords. Table 9.4 provides an example keyword file.
- **ECDA Concentration Files:** The ECDA concentration data files contain all the concentration data available for the impacts codes. The concentrations for all analytes for a given impact type are based on the same time and location data. There is a map file for each impacts code containing indexing information for each of the concentration data files. In addition, there is a separate concentration data file for every analyte used in the impacts code. For example, when running a case with 7 analytes for four impacts types, there will be 4 map files and 28 concentration data files. The file formats for the concentration data and map files are defined in Sections 2.2.1 and 2.2.2, respectively.

9.5.1.1 ESD Keyword File

RIPSAC reads keywords from the ESD keyword file. These keywords are read from a different file and can have a different definition than for keywords defined for internal use in RIPSAC. Section 2.1 defines the ESD keywords. The following keywords required:

- ANALYTE – definition of analytes in the environmental simulations
- END – end of the environmental settings keywords
- FILE – file names for the concentration file for each analyte
- LOCATION – locations at which concentrations were generated for the cultural impacts
- REALIZAT – number of realizations that were simulated
- TIMES – times at which concentrations were generated for the cultural impacts
- TITLE – environmental simulation title.

9.5.1.2 RIPSAC Keyword File

Section 9.4 describes the individual keywords for RIPSAC. Table 9.4 provides an example RIPSAC keyword file for nine analytes. This example file uses four locations for human impacts and four locations for ecological impacts. A RIPSAC keyword file for a production run incorporating hundreds of locations for each impact type would be several thousand lines long.

Table 9.4 Example RIPSAC Keyword File

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! Example RIPSAC Keyword File
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Report File (first keyword)
REPORT "F:\Initial_Assessment\Ripsac\RipSac.Rpt"
FILE ESD "F:\Initial_Assessment\ESD_Initial_Assessment.key"
TITLE "Riparian Zone Model (RIPSAC) Example Case"
USER "Paul W. Eslinger"
!
IMPACTS HUMAN ECOLOGICAL
!
ANALYTE ID="H3"
ANALYTE ID="Sr90"
ANALYTE ID="Tc99"
ANALYTE ID="I129"
ANALYTE ID="Cs137"
ANALYTE ID="U238"
ANALYTE ID="Pu239"
ANALYTE ID="U"
ANALYTE ID="Cr(VI)"
ANALYTE ID="CCl4"
!
EXECUTE
!
! Human Impacts: Riparian zone location matching
LOCATION HUMAN PRIMARY="HL0417" SECOND="HL0418" DF="DF5m  "
LOCATION HUMAN PRIMARY="HL0419" SECOND="HL0420" DF="DF5m  "
!
! Human Impacts: Soil-Water partition coefficients
KDSOIL LOCATION="HL0417" ANALYTE="H3    " KDSOIL="KDH   " HUMAN
KDSOIL LOCATION="HL0417" ANALYTE="Tc99  " KDSOIL="KDTc  " HUMAN
KDSOIL LOCATION="HL0417" ANALYTE="I129  " KDSOIL="KDI   " HUMAN
KDSOIL LOCATION="HL0417" ANALYTE="U238  " KDSOIL="KDU   " HUMAN
KDSOIL LOCATION="HL0417" ANALYTE="U     " KDSOIL="KDU   " HUMAN
KDSOIL LOCATION="HL0417" ANALYTE="Cs137 " KDSOIL="KDCs  " HUMAN
KDSOIL LOCATION="HL0417" ANALYTE="Sr90  " KDSOIL="KDSr  " HUMAN
KDSOIL LOCATION="HL0417" ANALYTE="Pu239 " KDSOIL="KDPu  " HUMAN
KDSOIL LOCATION="HL0417" ANALYTE="Cr(VI)" KDSOIL="KDCr  " HUMAN
KDSOIL LOCATION="HL0417" ANALYTE="CCl4  " KDSOIL="KDCCl4" HUMAN
!
KDSOIL LOCATION="HL0419" ANALYTE="H3    " KDSOIL="KDH   " HUMAN
KDSOIL LOCATION="HL0419" ANALYTE="Tc99  " KDSOIL="KDTc  " HUMAN
KDSOIL LOCATION="HL0419" ANALYTE="I129  " KDSOIL="KDI   " HUMAN
KDSOIL LOCATION="HL0419" ANALYTE="U238  " KDSOIL="KDU   " HUMAN
KDSOIL LOCATION="HL0419" ANALYTE="U     " KDSOIL="KDU   " HUMAN
KDSOIL LOCATION="HL0419" ANALYTE="Cs137 " KDSOIL="KDCs  " HUMAN
KDSOIL LOCATION="HL0419" ANALYTE="Sr90  " KDSOIL="KDSr  " HUMAN
KDSOIL LOCATION="HL0419" ANALYTE="Pu239 " KDSOIL="KDPu  " HUMAN
KDSOIL LOCATION="HL0419" ANALYTE="Cr(VI)" KDSOIL="KDCr  " HUMAN
KDSOIL LOCATION="HL0419" ANALYTE="CCl4  " KDSOIL="KDCCl4" HUMAN
!
! Ecological Impacts: Riparian zone location matching
LOCATION ECOLOGIC PRIMARY="EL0001" SECOND="EL0002" DF="DF5m  "
LOCATION ECOLOGIC PRIMARY="EL0003" SECOND="EL0004" DF="DF5m  "
!
! Ecological Impacts: Soil-Water partition coefficients
KDSOIL LOCATION="EL0001" ANALYTE="H3    " KDSOIL="KDH   " ECOLOGICAL
KDSOIL LOCATION="EL0001" ANALYTE="Tc99  " KDSOIL="KDTc  " ECOLOGICAL
KDSOIL LOCATION="EL0001" ANALYTE="I129  " KDSOIL="KDI   " ECOLOGICAL
KDSOIL LOCATION="EL0001" ANALYTE="U238  " KDSOIL="KDU   " ECOLOGICAL
KDSOIL LOCATION="EL0001" ANALYTE="U     " KDSOIL="KDU   " ECOLOGICAL
KDSOIL LOCATION="EL0001" ANALYTE="Cs137 " KDSOIL="KDCs  " ECOLOGICAL
KDSOIL LOCATION="EL0001" ANALYTE="Sr90  " KDSOIL="KDSr  " ECOLOGICAL
KDSOIL LOCATION="EL0001" ANALYTE="Pu239 " KDSOIL="KDPu  " ECOLOGICAL
KDSOIL LOCATION="EL0001" ANALYTE="Cr(VI)" KDSOIL="KDCr  " ECOLOGICAL
KDSOIL LOCATION="EL0001" ANALYTE="CCl4  " KDSOIL="KDCCl4" ECOLOGICAL
!
KDSOIL LOCATION="EL0003" ANALYTE="H3    " KDSOIL="KDH   " ECOLOGICAL
KDSOIL LOCATION="EL0003" ANALYTE="Tc99  " KDSOIL="KDTc  " ECOLOGICAL
KDSOIL LOCATION="EL0003" ANALYTE="I129  " KDSOIL="KDI   " ECOLOGICAL
KDSOIL LOCATION="EL0003" ANALYTE="U238  " KDSOIL="KDU   " ECOLOGICAL
KDSOIL LOCATION="EL0003" ANALYTE="U     " KDSOIL="KDU   " ECOLOGICAL
KDSOIL LOCATION="EL0003" ANALYTE="Cs137 " KDSOIL="KDCs  " ECOLOGICAL
KDSOIL LOCATION="EL0003" ANALYTE="Sr90  " KDSOIL="KDSr  " ECOLOGICAL
KDSOIL LOCATION="EL0003" ANALYTE="Pu239 " KDSOIL="KDPu  " ECOLOGICAL
KDSOIL LOCATION="EL0003" ANALYTE="Cr(VI)" KDSOIL="KDCr  " ECOLOGICAL
KDSOIL LOCATION="EL0003" ANALYTE="CCl4  " KDSOIL="KDCCl4" ECOLOGICAL
!
END
```

9.5.1.3 DILUTE Stochastic Data

If any seep water or soil concentration solutions are requested, then the file containing dilution factors for groundwater and river water interaction coefficients must be supplied. This file is generated by the ECDA program from DILUTE keywords contained in the ESD keyword file. Section 2.3.1 defines the file format, and Table 2.11 provides an example file. The file name is supplied in the ESD keyword file by using the FILE keyword and the DILUTE modifier (see Section 2.1.10).

9.5.1.4 KDSOIL Stochastic Data

If any soil concentration solutions are requested, then the file containing soil-water partition coefficients must be supplied. This file is generated by the ECDA program from KDSOIL keywords contained in the ESD keyword file. Section 2.3.2 defines the file format and Table 2.12 provides an example file. The file name is supplied in the ESD keyword file by using the FILE keyword and the KDSOIL modifier (see Section 2.1.10).

9.5.1.5 Concentration Data Files

Concentration data for every combination of analyte and impact type (human, cultural, economic, or ecological) are contained in a separate file. RIPSAC reads from, and writes to, a concentration data file for every combination of impacts type and analyte being simulated. If there are 4 impact types and 10 analytes, then 40 concentration data files will be required, along with 4 (one for each impacts type) record map files. Sections 2.2.1 and 2.2.2 describe the format for each of these files. The ESD keyword file supplies the concentration data and map file names (see Section 2.1.10).

9.5.2 Output Files

RIPSAC writes a report file, a run-time signal file named RIPSAC.Run, and a run completion signal file named RIPSAC.Done as well as modifies ECDA concentration data files. The number of modified concentration data files depends on the options selected in the scenario being analyzed. The following sections describe the output files.

9.5.2.1 RIPSAC Report File

The RIPSAC report file contains run information and error messages, if any. The file contains program identification and an echo of the basic options selected for the code run. Table 9.5 provides excerpted records from an example RIPSAC report file. Locations where records were deleted are identified by lines containing three dots. This example run is for processing two realizations for the single analyte CrVI (hexavalent chromium). This particular run failed because invalid concentration data were located in the groundwater medium.

Table 9.5 Excerpted Records from a RIPSAC Report File

```
RRRRR    IIIIIII  PPPPP      SSS     AAAAA     CCC
R    RR     I     P    PP  S    SS  A     A   C   CC
R    RR     I     P    PP   S    S  A     A  C
RRRRR       I     PPPPP      SS     AAAAAAA  C
R    R      I     P        S   S    A     A  C
R     R     I     P        SS   S   A     A   C   CC
R     R  IIIIIII  P          SSS    A     A    CCC


                     RipSac   1.00.B.1
             Last Modified on 14 Aug 2001


           Stochastic River-Shore Analysis
  Systems Assessment Capability (SAC), Revision 0
  ------------------------------------------------
               Developed By Battelle
       Pacific Northwest National Laboratory
               Richland, Washington
               ------------------------------------------------

          Current Run ID = 200203011343   User Name = Paul W. Eslinger
               System Date = 03-01-2002   System Time = 13:43:32


           The software used to generate this output is experimental
              and has not been formally tested or peer reviewed.


                             Review Signatures

Input Prepared By: ______________________________       Date: _______________

Input Reviewed By: ______________________________       Date: _______________


======================== Echo of the Problem Definition ========================

Title: Riparian Zone Model for the SAC Rev. 0 Shakedown 2 Assessment
User:  Paul W. Eslinger
   2 Realizations identified in the ESD keyword file
   2 Realizations utilized in this run

A map of realization activations (true/false) is:
   1-  2  T  T

File Name for RIPSAC Input Keyword Data
File: RipSac\RipSac.key

File Name for ESD Input Keyword Data
File: G:\Shakedown2\ESD_Shakedown2_RipSac.key

File Name for Dilution Data
File: G:\Shakedown2\ecda\DILUTE_Shakedown2.Dat

File Name for KDSOIL Data
File: G:\Shakedown2\ecda\KDSOIL_Shakedown2.Dat

File Name for Human      Concentration Map File
File: G:\Shakedown2\ecda\HUMA_Shakedown2.Map

File Name for Ecological Concentration Map File
File: G:\Shakedown2\ecda\ECOL_Shakedown2.Map

File Name for Economic   Concentration Map File
File: G:\Shakedown2\ecda\ECON_Shakedown2.Map

File Name for Cultural   Concentration Map File
File: G:\Shakedown2\ecda\CULT_Shakedown2.Map

File Name for Human      Media Concentrations: Analyte ID="CrVI"
File: G:\Shakedown2\ecda\HUMA_CrVI_Shakedown2.Dat

File Name for Ecological Media Concentrations: Analyte ID="CrVI"
File: G:\Shakedown2\ecda\ECOL_CrVI_Shakedown2.Dat
```

```
File Name for Economic   Media Concentrations: Analyte ID="CrVI"
File: G:\Shakedown2\ecda\ECON_CrVI_Shakedown2.Dat

File Name for Cultural   Media Concentrations: Analyte ID="CrVI"
File: G:\Shakedown2\ecda\CULT_CrVI_Shakedown2.Dat

Analye Information for 10 analytes.
    1 : H3     : ===> Not used
    2 : Sr90   : ===> Not used
    3 : Tc99   : ===> Not used
    4 : I129   : ===> Not used
    5 : Cs137  : ===> Not used
    6 : U238   : ===> Not used
    7 : Pu239  : ===> Not used
    8 : U      : ===> Not used
    9 : CCl4   : ===> Not used
   10 : CrVI   : Hexavalent Chromium
A total of  1 analytes have been requested.

Number of Human      times is 231
Number of Ecological times is 221
Number of Economic   times is 211
Number of Cultural   times is 70
Index    Human Ecologic Economic Cultural
    1     1980     1990     2000   1945
    2     1981     1991     2005   1950
    3     1982     1992     2010   1955
    4     1983     1993     2015   1960
...

Number of Human      locations is 1930
Number of Ecological locations is 674
Number of Economic   locations is 13
Number of Cultural   locations is 3981
Index    Human Ecologic Economic Cultural
    1   HL0151   EL0001   EN0011   CL0001
    2   HL0152   EL0002   EN0021   CL0002
    3   HL0153   EL0003   EN0031   CL0003
    4   HL0154   EL0004   EN0041   CL0004
...


Soil-Water KD and Dilution Factor Mapping for Human Impacts
  Location      Analyte      KDSOIL ID String      DILUTE ID String
  ------------  -----------  --------------------  --------------------
   267: HL0417   10: CrVI    KDCr                  DF5m
   269: HL0419   10: CrVI    KDCr                  DF5m
   271: HL0421   10: CrVI    KDCr                  DF5m
   273: HL0423   10: CrVI    KDCr                  DF5m
...


Soil-Water KD and Dilution Factor Mapping for Ecological Impacts
  Location      Analyte      KDSOIL ID String      DILUTE ID String
  ------------  -----------  --------------------  --------------------
```

```
     1: EL0001   10: CrVI    KDCr                 DF5m
     3: EL0003   10: CrVI    KDCr                 DF5m
     5: EL0005   10: CrVI    KDCr                 DF5m
     7: EL0007   10: CrVI    KDCr                 DF5m
...

======================== End of the Problem Definition =========================

Message originating in routine PROCESS_HUMA
Message: Entering subroutine for analyte "CrVI"
         On 03/01/2002 at 13:43:39.781

Error number   10 encountered in routine PROCESS_HUMA
Message: Negative ground water concentration encountered
         Location = HL0433  Analyte =  CrVI
         Year        = 1980
         Realization = 2
         Value       = -2.30123E-11

Error number   10 encountered in routine RipSac
Message: Abnormal Run Termination Due to Errors
         Run Completed on 03/01/2002 at 13:43:39.937
```

9.5.2.2 RIPSAC.Run File

The RIPSAC.Run file is written just as RIPSAC starts execution. Presence of the file signals the SAC ESP that a RIPSAC run is in progress. This file does not contain any data. The file is closed and deleted when the RIPSAC run terminates.

9.5.2.3 RIPSAC.Done File

The file RIPSAC.Done is always written just as RIPSAC finishes a run. This file contains a single line of text data. If the run was successful, the line reads as follows:

```
“0 Normal termination”
```

If the run was not successful, the line will have the following form:

```
“1 Error termination”
```

The first entry on the line for runs that terminate with errors will be a nonzero error code that can be used to help trace the source of the error. Typically, the ending lines in the report file (see Table 9.5) will contain more detailed information about the type of error encountered.

9.5.2.4 Concentration Data Files

Concentration data for every combination of analyte and impact type (human, cultural, economic or ecological) are contained in a separate file. RIPSAC reads from, and writes to, a concentration data file for every combination of impacts type and analyte being simulated. If there are 4 impact types and 10 analytes, then 40 concentration data files will be required, along with 4 (one for each impacts type) record map files. The format for each of these files is described in Sections 2.2.1 and 2.2.2, respectively. The concentration data and map file names are supplied using the FILE keyword (see Section 2.1.10) in the ESD keyword file.

10.0 Keyword Language Syntax

Each line of a keyword data file is parsed into numeric and character data. These are interpreted to set up control information and define input parameters. An input line can contain up to 180 characters of information. Lines longer than 180 characters are truncated to 180 characters.

Every line of the input data file is considered a keyword record, continuation record, or a comment record. Keyword records contain a keyword beginning in column one. The keyword is used to determine the purpose of the subsequent data. Continuation records are used when a keyword record requires too much data to be placed on one line. Comment lines are ignored by the reading software but are useful for annotating the input file.

The information from each keyword record and subsequent continuation lines is moved into storage arrays. Data that can be deciphered as numeric values are placed into a numeric array. Other data are classified either as secondary keywords (called modifiers) or quote strings. Secondary keywords are stored as character images in an array. All such keywords or modifiers read from the input file are changed to upper case before being stored. Quote strings are text strings that are enclosed in double quotes. These are stored exactly as they are read from the input file.

10.1 Keyword Records

Keyword records start in column one with any letter from A to Z, in either upper or lower case. The first eight letters of a keyword are stored in a variable and are used by the modeling software to determine the actions desired by the program user. All subsequent lines of text that do not have an alphabetic character or comment character in column 1 are treated as continuation lines. An example keyword record (where SAMPLEKEY starts in column one) is the following:

```
SAMPLEKEY 2 0 500 1 100
```

The word SAMPLEKEY is the keyword. The numbers 2, 0, 500, 1, and 100 are numeric data.

10.2 Continuation Records

Continuation records start with any valid separator character (except a double quote). These are treated as additional data to the previous keyword record. Section 10.4.2 identifies valid separator characters. The combined data on a keyword line and on the subsequent continuation line(s) are treated as a single block of information. All numeric values and character strings on those lines are used as input data relevant to the keyword of the keyword line. The two following keyword entries contain the same information:

```
SAMPLEKEY 2 0 500 1 100

SAMPLEKEY 2 0
  500 1 100
```

10.3 Comment Records

Any line with the characters $, !, /, or * in column 1 will be treated as a comment record. These lines are ignored by the input data record reader. Both the $ and the ! can also be used to signify in-line comments (not in column 1). Any information that follows a $ or a ! will be ignored. The * and / characters indicate a comment only if one is the first character on the input line. The * character is also used as a repetition factor in the input. Some examples of comment lines are the following:

```
$This is a comment line
/This is a comment line
!This is a comment line
*This is a comment line
```

Some examples of in-line comments are the following:

```
SAMPLEKEY 3 4.0 5.0 !Trailing information is ignored after the !
SAMPLEKEY 3 4.0 5.0 $Trailing information is ignored after the $
```

10.4 Input Data Handling

Each line of the input is read and parsed into numeric and character values. All numeric values are converted to real numbers (as opposed to FORTRAN integer). All data that cannot be interpreted as numeric information are stored as character values.

Numeric data can include a leading sign (+ or -), integer characters 0 through 9, a decimal point, and an exponent indication (“E” or “e”). The FORTRAN “double precision” exponent indicator “D” is not valid. A maximum of 10 digits is allowed when entering numeric data.

Secondary keywords, or modifiers, are character strings that could not be interpreted as numeric values. These are converted to upper case, where necessary, and stored in an array. The number of secondary keywords that are moved into the array is stored for internal use.

Only the first eight characters of any keyword or modifier are significant. Keywords or modifiers fewer than eight characters long are left justified and blank-filled.

10.4.1 Quote Strings

Quote strings are strings of literal text that must be used exactly as given in the input line. They are enclosed by double quote characters and are typically used for passing file names into a program. Only the first 72 characters of a quote string are saved. Each quote string must begin and end on a single line of the input file. When an unclosed quote is encountered, an implied quote is created at the end of that line. An example of quote string usage is the following:

```
FILE “c:\apps\human\test.dat”
```

10.4.2 Data Separators

Keywords, numeric data, and secondary keywords must be separated by any one of the following data separators: space character, comma, equal sign, colon, semicolon, left parenthesis, right parenthesis, single quote, double quote, or tab character.

Also, any character with a ASCII character storage code of fewer than 10 is treated as a separator character. This is used mainly to identify the ASCII tab character as a data separator. Double quote characters are used differently than the other separators. They indicate text strings that are stored without conversion by the program. Double quote characters cannot be used as the first character of a continuation line. As an illustration of the use of separator characters, the following keyword records all contain the same information:

```
   SAMPLEKEY 3 4.5 5.6 6.7
   SAMPLEKEY 3 (4.5,5.6,6.7)
   SAMPLEKEY 3 ( 4.5=5.6'6.7 )
   SAMPLEKEY 3:4.5  5.6:6.7
```

11.0 Stochastic Variable Generation

Many of the codes in SAC, Rev. 0, generate values for stochastic variables. All of the codes use the same suite of statistical routines to do this generation. The following are some major considerations for this process:

- Each distribution is generated using the Probability Integral Transformation method (Mood et al. 1974, p. 202)
- The uniform number generator uses a linear congruential method (Lewis et al. 1969)
- Stratified sampling is used when the number of values to be generated is greater than 1
- Most distributions may be truncated between two limits that are specified as limits in the uniform domain on the interval 0 to 1
- The user may specify a cumulative distribution function in the form of a table of values
- Information about a stochastic variable is linked to a unique character ID. Access to all information about the variable is available through use of the variable ID.

The following statistical distributions are available:

- Constant value
- Uniform distribution between two limits
- Discrete uniform distribution on a set of contiguous integers
- Loguniform (base 10) distribution between two limits
- Loguniform (base e) distribution between two limits
- Triangular distribution defined using a lower limit, mode, and an upper limit
- Normal distribution with a mean and standard deviation
- Lognormal (base 10) distribution specified by the mean and standard deviation of the logarithms of the data
- Lognormal (base e) distribution specified by the mean and standard deviation of the logarithms of the data
- User specified cumulative distribution function input as a table of probabilities and exceedance values
- Beta distribution that can be shifted and scaled from the standard (0,1) interval
- Log-ratio from a normal distribution
- Hyperbolic arcsine from a normal distribution.

11.1 Stochastic Keywords

In the following discussion, the description is presented such that the keyword name for entering stochastic variable information is STOCHASTIC. In reality, a variety of keyword names are used, including KDSOIL and DILUTE, for example. The keyword STOCHASTIC will be used in the following discussion in order to simplify the presentation. This keyword facilitates entering the statistical distribution for stochastic variables. The general syntax for the STOCHASTIC record is the following:

STOCHASTIC [“Quote$_1$”] [Dist_Index Parameters] {TRUNCATE U1 U2} {“Quote$_2$”}

The entry for Quote_1 must be a unique character string of up to 20 characters that will be used to identify this stochastic variable in subsequent uses. It is case sensitive and embedded spaces are significant. It is sometimes useful to make the character string some combination of a variable name and other data such that it can be recreated easily when stochastic data is needed. The entry for Quote_2 is a description for the stochastic variable that can be up to 64 characters long that is used for output labeling purposes. The entry for Quote_2 is optional.

The entry for Index must be an integer in the range 1 to 13 that identifies the index of a statistical distribution. Table 11.1 defines the statistical distributions. The word Parameters in the general syntax statement indicates the numerical values of parameters required for defining the statistical distribution. The additional modifier TRUNCATE can be used for all distribution types except 1, 3, and 10. If TRUNCATE is entered, it must be followed by two values in the interval 0 to 1, inclusive of the endpoints. The lower value must be less than the upper value. These two values specify the tail probabilities at which to impose range truncation for the distribution. Truncation data must be entered after all of the other parameters that define the distribution.

Table 11.1 Statistical Distributions Available in All Impacts Codes

Index	Distribution	Truncate	Parameters Required
1	Constant	No	Single value.
2	Uniform	Yes	Lower limit, upper limit.
3	Discrete Uniform	No	Smallest integer, Largest integer.
4	Loguniform (base 10)	Yes	Lower limit, upper limit.
5	Loguniform (base e)	Yes	Lower limit, upper limit.
6	Triangular	Yes	Lower limit, mode, upper limit.
7	Normal	Yes	Mean, standard deviation.
8	Lognormal (base 10)	Yes	Mean, standard deviation of logarithms.
9	Lognormal (base e)	Yes	Mean, standard deviation of logarithms.
10	User Defined	Yes	Number of pairs, data for pairs of values ($\text{Prob}(X_i),X_i$).
11	Beta	Yes	Alpha, beta, lower limit, upper limit. The mean of the distribution would be alpha/(alpha+beta) if the limits were 0 and 1.
12	Log ratio	Yes	Mean, Standard deviation (of normal), lower limit, upper limit.
13	Hyperbolic arcsine	Yes	Mean, Standard deviation (of normal).

The following is an example stochastic keyword for a variable assigned a constant of 234.432:

```
STOCHASTIC “Unique1” 1 234.432 “Define a constant distribution”
```

The constant can take any value.

The following is an example stochastic keyword for a variable assigned a uniform distribution on –2 to 7:

```
STOCHASTIC "Unique2" 2 -2.0 7
   "Define a uniform distribution on -2 to 7"
```

The two limits can take any values as long as the second value is strictly greater than the first value.

The following is an example stochastic keyword for a variable assigned a discrete uniform distribution on the integers 6 to 70:

```
STOCHASTIC "Unique3" 3 6 70
   "Define a discrete uniform distribution on 6 to 70"
```

The two limits must be integers where the second integer is strictly greater than the first integer.

The following is an example stochastic keyword for a variable assigned a loguniform (base 10) distribution on the interval 10^{-7} to 10^{-3}:

```
STOCHASTIC "Unique4" 4 1.0E-7 1.0E-3
   "Define a loguniform (base 10) distribution on 0.0000001 to 0.001"
```

The two limits must both be greater than zero and the second limit must be greater than the first limit.

The following is an example stochastic keyword for a variable assigned a loguniform (base e) distribution on the interval 10^3 to 10^6:

```
STOCHASTIC "Unique5" 5  1.0E3  1E+6
   "Define a loguniform (base e) distribution on 1000 to 1000000"
```

The two limits must both be greater than zero and the second limit must be greater than the first limit.

The following is an example stochastic keyword for a variable assigned a triangular distribution with a minimum of 2, a mode of 3, and a maximum of 7:

```
STOCHASTIC "Unique6" 6 2 3 7 "Define a triangular distribution on (2,3,6)"
```

The three values that define the triangular must all be different, and they must be entered in increasing order.

The following is an example stochastic keyword for a bioconcentration factor that is normally distributed with a mean of 125 and a standard deviation of 5 for a frog exposed to ^{14}C:

```
STOCHASTIC "BCFC14Frog" 7 125.0 5.0 "Example normally distributed frog"
```

The mean value can be any number, but the standard deviation must be greater than zero.

The following keyword would define a different stochastic variable than the one just entered because the identification string (Quote1) is case sensitive:

```
STOCHASTIC “BCFC14FROG” 7  125.0  5.0  “Example normally distributed frog”
```

The following keyword entry would define a lognormal (base 10) distribution where the mean and standard deviation (of the logarithms) are –2.0 and 0.5:

```
STOCHASTIC “Unique8”  8  -2  0.5
   “Example for a lognormal (base 10) variable”
```

The mean value can be any number, but the standard deviation must be greater than zero.

The following keyword entry would define a lognormal (base e) distribution where the mean and standard deviation (of the logarithms) are –2.0 and 0.5. In addition, the lognormal distribution will be truncated between the lower 0.025 and upper 0.99 probabilities.

```
STOCHASTIC “Unique9” 9 -2 .5 TRUNCATE 0.025 0.99
   “Example for a truncated lognormal variable”
```

The mean value can be any number, but the standard deviation must be greater than zero.

The following keyword entry illustrates the use of the user-defined distribution (distribution type 10). This example entry uses seven pairs of values. The first pair of numbers uses a probability of 0 to define the lower limit of the distribution at 8.4 E-7. The last pair of numbers uses a probability of 1 to define the upper limit of the distribution at 1.73E-6. The other values are associated with the probability levels of .025, .167, .5, .833, and .975. The probability data and distribution percentiles must be entered in strictly increasing order.

```
STOCHASTIC "Sr90Con" 10  7
   0  8.40E-7
   2.50E-02  9.20E-7
   1.67E-01  1.06E-6
   5.00E-01  1.21E-6
   8.33E-01  1.37E-6
   9.75E-01  1.58E-6
   1  1.73E-6
```

The first pair of numbers uses a probability of 0 to define the lower limit of the distribution. The last pair of numbers uses a probability of 1 to define the upper limit of the distribution. The intervening pairs define probability levels o and the associated data values. The probabilities and data values must be entered in strictly increasing order.

The following keyword entry would define a beta distribution with parameters 1.1 and 2.1 on the interval (0,1):

```
STOCHASTIC "Unique11-1"  11  1.1  2.1  0.0  1.0
   "Beta (1.1,2.1) on the interval 0,1"
```

Let the first parameter be denoted by α and the second parameter be denoted by β. The mean of the beta distribution would be α/(α+β) if the limits were 0 and 1. Both α and β must be greater than zero. The lower limit must be less than the upper limit.

The following keyword entry would define a beta distribution with parameters 1.1 and 2.1 but on the interval –2 to 4:

```
   STOCHASTIC "Unique11-2"  11  1.1  2.1  -2.0  4.0
      "Beta (1.1,2.1) on the interval (-2,4)"
```

The following keyword entry would define a log ratio distribution from a normal(-1.459,1.523) distribution on the interval -5.756 to 4.33.

```
   STOCHASTIC "Test1203" 12 -1.459  1.523 -5.756 4.330
      "Log ratio from Normal(-1.4,1.5) on (-5.756,4.330)"
```

The entry for the normal standard deviation (a value of 1.523 in this example) must be greater than zero. The last two numerical values define the interval for the generated values, so the lower limit must be smaller than the upper limit.

The following keyword entry would define a hyperbolic arcsine distribution from a normal(0.189,0.146) distribution.

```
   STOCHASTIC "Test1302" 13 0.189  0.146
      "Hyperbolic Arcsine from Normal(0.189,0.146)"
```

The entry for the normal standard deviation (a value of 0.189 in this example) must be greater than zero.

11.2 Probability Concepts

The distribution of a continuous random variable X (the term continuous indicates that the random variable is defined over a continuum of values) is completely described by its probability density function, f(x). The interpretation of the probability density function is that the area under f(x), for an interval a<x<b, equals the probability that the random variable, X, will fall in the interval (a,b), denoted P[a<X<b]. One cannot make the statement P[X=t] because there is no area under the probability density function at any given point t. Two axioms of probability theory (Mood et al. 1974, p. 22) are that the probability of any event is between zero and one, and the integral of the probability density function over the entire support (the interval [L,U]) of X equals 1. The integral of the probability density function from the lower bound L to some value x (suppose it is less than the upper bound U), represents the probability that X will be observed in the interval (L,x). This integral operation defines the cumulative distribution function for the random variable X. The cumulative distribution function is denoted by F(x) (the capital F for the cumulative distribution function corresponds to the lower case f for the probability density function) and mathematically is represented by

$$F(x) = \int_{L}^{U} f(s)ds$$

The inverse cumulative distribution function, $[F^{-1}(\bullet)]$, is single-valued if x is in the interval (L,U). Hence if p'= F(x') is known, in theory x'= F^{-1}(p') can be obtained.

11.3 Probability Integral Transform Method

Generation of a random variable from a given distribution typically involves the use of information either about f or F. There are two philosophical approaches to generating random numbers: exact methods and approximate methods. The algorithms embedded in SAC employ exact methods. Exact methods can be further categorized into probability integral transform methods and functional methods. The probability integral transform method is employed in SAC.

In the probability integral transform method, the random variable of interest is expressed as a function of a U(0,1) random variable, where U(0,1) denotes the continuous random variable ranging uniformly over the interval (0,1). The probability density function of the uniform random variable is g(u)=1 if u0(0,1) and is zero elsewhere. The cumulative distribution function for this random variable takes the exceedingly simple form G(u)=u. It can be shown that any cumulative distribution function evaluated at a random value X (instead of being evaluated at a known value x as in the previous discussion) is distributed uniformly over the interval (0,1) (Mood et al. 1974, p. 202). Therefore, given a realization u of the U(0,1) random variable and a selected statistical distribution (known cumulative distribution function), one can set u=F(x) and solve to obtain x=F^{-1}(u). The value x thus obtained is a random realization from the selected statistical distribution.

In principle, one can obtain an exact solution for x given any specific cumulative distribution function and value u. In reality, there exist some distributions, such as the normal and beta distributions, for which no closed-form analytical expression for F^{-1} exists, and hence approximation methods must be applied.

The probability integral transform method allows efficient sampling from a subregion of the interval (L,U), such as (c,d), where L<c<d<U. In this case, one would find the corresponding interval in the uniform domain, say (c',d'), and sample uniformly over that interval by sampling from the rescaled uniform distribution [for example, u'=(d-c)u+c] and then obtaining x as usual using x=F^{-1}(u'). The rescaled uniform distribution takes the form g(u)=1/(d'-c') for u0(c',d') and is zero elsewhere. For any distribution with probability density function f(x) and cumulative distribution function F(x), the probability density function, under truncation to the interval (c,d), is $f_T(x)$ = f(x)/[F(d)-F(c)]. The divisor ensures that $f_T(x)$ integrates to unity.

11.4 Stratified Sampling

Stratified sampling can easily be implemented when generating random deviates using the probability integral transform method. This is accomplished by dividing the uniform interval (0,1) into subintervals, or strata, and sampling a specified number of times within each stratum, each time obtaining the corresponding value of x. Within SAC, the strata intervals are assigned equal probability, and exactly one value is sampled within each stratum. The method generates samples from each stratum, then randomly shuffles the entire set of realizations using a variation of the Quicksort algorithm (Singleton 1969). The primary purpose of stratified sampling is to achieve more evenly spaced (in a probability sense) samples from the distribution of a random variable than would result from randomly sampling over the whole range of the distribution. Iman and Conover (1982) have shown that stratified sampling can result in more efficient estimation of simulation results for a variety of estimators than when using simple random sampling.

11.5 Generation Algorithms

Table 11.1 summarizes the statistical distributions available in the SAC codes. Descriptions of the generation algorithms are provided in the following paragraphs.

11.5.1 Algorithm for the Uniform Distribution

Algorithms that generate truly random uniform numbers do not exist, although many algorithms generate pseudo-random deviates (hereafter loosely referred to as random numbers). The selection of a random number generator is based on four considerations: 1) computer implementability, 2) degree of independence within a sequence of deviates, 3) periodicity or cyclic length of a sequence, and 4) uniform coverage of sequences (occurrence) over the interval (0,1), the square (0,1) X (0,1), etc., up to the hypercube (0,1) in k dimensions.

Commonly used random number generation techniques on digital computers involve feedback shift register and linear congruential methods (Kennedy and Gentle 1980, pp. 136, 150). The SAC codes use a linear congruential random number generator. The linear congruential generator generates random integers from an algorithm of the form $S_i = (A \bullet S_{i-1} + C) \bmod M$ where S_i is the ith generated random integer, A and C are constants, M is the modulus of the generated integers, and mod denotes the remainder function. These integers are converted to approximate uniform (0,1) numbers by the division $U_i = S_i / M$.

The period of a sequence $\{U_i\}$ of generated deviates is the minimal value k such that $U_i=U_{i+k}$ (this occurs independent of i for linear congruential generators). It can be shown that the period of any congruential generator does not exceed M. Therefore, if one is generating many uniform deviates, it is desirable that M be large. The performance of each congruential generator (each choice of A, C, and M) can thus be examined with respect to criteria proceeding from the four considerations given above. The SAC implementation uses A=16807, C=0, and M=2147483647. These choices yield a sequence $\{U_i\}$ that 1) is implementable on a 32-bit computer without machine language coding, 2) is sufficiently independent on an element-by-element basis, 3) possesses a long cycle (period), and 4) has a reasonable degree of coverage over all hypercubes of dimension less than k. These conclusions proceed from results from tests described in Fishman and Moore (1986).

Any value, x, generated from the uniform(a,b) distribution in the SAC codes makes use of a value, y, from the U(0,1) distribution. The value y is first generated, then x is evaluated as x=a+(b-a)y. The uniform(a,b) distribution will be denoted by U(a,b).

11.5.2 Algorithm for the Discrete Uniform Distribution

The probability density function for the discrete uniform distribution is f(x)=1/N for each of the N integers ranging in the interval L to U. The generation algorithm for the discrete uniform distribution is:

$$F^{-1}(u) = L + \text{int}[u(U - L + 1)]$$

where the int(·) function returns the integer portion of its argument.

11.5.3 Algorithm for the Loguniform Distribution

The probability density function for the loguniform random variable of base b is:

$$f(x) = \frac{I(b^c < x < b^d)}{x(d - c) \bullet \ln(b)}$$

for -4<c<d<4, where I is an indicator function (0 if false, 1 if true), b is the logarithm base (either 10 or the natural constant e), and ln(b) denotes the natural logarithm of b.

The inverse cumulative distribution function algorithm used to generate a value, x, from the loguniform distribution first generates a value, y, from the U(c,d) distribution and then evaluates the expression $x=b^u$.

11.5.4 Algorithms for the Triangular Distribution

The triangular distribution has probability density function:

$$f(x) = \begin{Bmatrix} 2(x - a)/[(b - a)(c - a)] & \text{for} & a < x \le b \\ 2(c - x)/[(c - b)(c - a)] & \text{for} & b \le x < c \end{Bmatrix}$$

and takes the value 0 elsewhere.

The generation algorithm for the triangular distribution is given in the following equation:

$$F^{-1}(u) = \begin{Bmatrix} a + \sqrt{u(c - a)(b - a)} & \text{for} & 0 \le u \le (b - a)/(c - a) \\ c - \sqrt{(1 - u)(c - a)(c - b)} & \text{for} & (b - a)/(c - a) \le u \le 1 \end{Bmatrix}$$

11.5.5 Algorithms for the Normal Distribution

A normal (μ,Φ^2) random deviate, y, is obtained by generating a N(0,1) deviate, x, then transforming that value using the equation $y=\mu+\Phi x$. The normally distributed random variable with mean μ and variance σ^2, denoted as $N(\mu,\Phi)$, has the probability density function:

$$f(x) = \frac{e^{-(x-\mu)^2/(2\sigma^2)}}{\sigma\sqrt{2\pi}}$$

for -4<x<4, -4<μ<4 and Φ>0.

The inverse cumulative distribution function for the N(0,1) random variable does not have a closed form expression. It is approximated by:

$$F^{-1}(p) = \begin{Bmatrix} q\,A(q^2)/B(q^2) & \text{for } |q| < 0.42 \\ \text{sgn}(q)\,C(r)/D(r) & \text{otherwise} \end{Bmatrix}$$

where q=p-0.5, and r=√ln(0.5-|q|). The quantity (0.5-|q|) is formed as p or, to avoid cancellation if p is small, as (1-p). The letters A, B, C, and D represent polynomials of order 3, 4, 3, and 2, respectively, whose coefficients are given in Beasley and Springer (1985), and sgn(q)=1 if q>0 and -1 if q<0.

11.5.6 Algorithms for the Lognormal Distribution

The logarithm of a random variable that is lognormally distributed is distributed as a normal $N(\mu,\Phi^2)$ random variable (thus the name lognormal). The probability density function of the lognormal distribution is:

$$f(x) = \frac{A}{x\sigma\sqrt{2\pi}} e^{-[\log(x)-\mu]^2/2\sigma^2}$$

for x>0 and σ>0. Because this distribution is available in both base 10 and natural logarithm base form, the constant A is $1/\log_e 10$ for base 10 and 1 for the natural logarithm base. The logarithm log(x) is also evaluated in terms of the chosen base.

A lognormal random variable, x, is generated using a two-step process. First, a value, y, is generated from the $N(\mu,\Phi^2)$ distribution. This value is then used in the expression $x=b^y$, where the base b is either 10 or the natural constant e as desired.

11.5.7 Algorithms for the User Defined Distribution

In addition to selecting from parametric families of distributions, the user may implement any other distribution by supplying a table of data pairs corresponding to the pairs [F(x),x]. Thus, the user provides the SAC code with discrete evaluations of the cumulative distribution function. The algorithm linearly interpolates between these points to solve for F^{-1} when generating a random deviate.

11.5.8 Algorithms for the Beta Distribution

The beta random variable, x, is described by the probability density function:

$$f(x) = \frac{x^{p-1}(1-x)^{q-1}}{B(p,q)}$$

for p>0, q>0, and 0<x<1. This variable can be transformed to the interval (a,b), and the resulting probability density function for the random variable, y, takes the form:

$$f(x) = \frac{(b-a)^{-(p+q+1)}(y-a)^{p-1}(b-y)^{q-1}}{B(p,q)}$$

for p>0, q>0, and a<y<b. The second expression for the probability density function can be obtained from the first by the change of variable y=(b-a)x+a.

A closed form expression for the beta inverse cumulative distribution function does not exist. The algorithm implemented is provided in Algorithm AS 64/AS 109 (Griffiths and Hill 1985, p. 121). Algorithm AS 64/AS 109 requires the logarithm of B(p,q). Utilizing the relationship between the beta and gamma functions (Mood et al. 1974, p.535), B(p,q)=G(p+q)/{G(p)G(q)}, where G(·) denotes the gamma function, the logarithm of B(p,q) is computed using Algorithm ACM 291 (Pike and Hill 1966). Algorithm AS 64/AS 109 uses approximate starting values and a Newton-Rhapson iterative method to achieve a final solution.

11.5.9 Algorithms for the Log Ratio Distribution

Let y denote a random variable from the log ratio distribution on the interval (a,b). The probability density function for y is:

$$f(y) = \frac{(b-a)e^{-(\ln\{(y-a)/(b-y)\}-\mu)^2/(2\sigma^2)}}{\sigma(y-a)(b-y)\sqrt{2\pi}}$$

where a<b and a<y<b.

A random variable, y, from the log ratio distribution is generated using a two-step process. First, a value, x, is generated from the N(μ,Φ^2) distribution. This value is then used in the expression:

$$F^{-1}(x) = \frac{a+be^x}{1+e^x}$$

11.5.10 Algorithms for the Hyperbolic Arcsine Distribution

Let the random variable x be a normally distributed random variable with mean μ and variance σ^2. Then, let y be a random variable defined as y = $\sinh^{-1}(x)$. The probability density function for y is:

$$f(y) = \frac{0.5(e^{u} + e^{-u})e^{-(\sinh(y)-\mu)^2/(2\sigma^2)}}{\sigma\sqrt{2\pi}}$$

for $-4<y<4$, $-4<\mu<4$ and $\Phi^2>0$.

A random variable, y, from the hyperbolic arcsine distribution is generated using a two-step process. First, a value, x, is generated from the $N(\mu,\Phi^2)$ distribution. This value is used in the expression:

$$F^{-1}(x) = \sinh(x)$$

12.0 References

Beasley JD, and SG Springer. 1985. "The Percentage Points of the Normal Distribution." Algorithm AS 111 in *Applied Statistics Algorithms*, eds. P Griffiths and ID Hill. Ellis Horwood Limited, Chichester, United Kingdom.

BHI. 1999. *Groundwater/Vadose Zone Integration Project: Preliminary System Assessment Capability Concepts for Architecture, Platform and Data Management*. CCN 0512242, Bechtel Hanford, Inc., Richland, WA. Online at: http://www.bhi-erc.com/projects/vadose/sac/sacdocs.htm.

CFEST. 2000. *Coupled Fluid, Energy, and Solute Transport (CFEST) Software User's Manual (DRAFT): Integrated 2D and 3D Fate and Transport Groundwater Technology for Monte Carlo Simulations.* Consultants for Environmental System Technologies Company, Irvine, CA.

Eslinger PW, C Arimescu, DW Engel, BA Kanyid, and TB Miley. 2002. *User Instructions for the Systems Assessment Capability, Rev. 0, Computer Codes. Volume 2: Impacts Modules*. PNNL-13932-Volume 2, Pacific Northwest National Laboratory, Richland, Washington.

Fishman GS, and LR Moore. 1986. "An Exhaustive Analysis of Multiplicative Congruential Random Number Generators with Modulus 2^{31}-1." *Siam Journal of Scientific and Statistical Computing* 7(1):24-44.

Griffiths P, and ID Hill, eds. 1985. *Applied Statistics Algorithms*. Ellis Horwood Limited, Chichester, England.

Iman RL, and WJ Conover. 1982. "A Distribution-Free Approach to Inducing Rank Correlations Among Input Variables." *Communications in Statistics* B11(3): 311-334.

Kennedy WJ Jr., and JE Gentle. 1980. *Statistical Computing*. Marcel Dekker, New York.

Kincaid CT, MP Bergeron, CR Cole, MD Freshley, NL Hassig, VG Johnson, DI Kaplan, RJ Serne, GP Streile, DL Strenge, PD Thorne, LW Vail, GA Whyatt, SK Wurstner. 1998. "Appendix B – Environmental Restoration Site Inventories" in *Composite Analysis for Low-Level Waste Disposal in the 200 Area Plateau of the Hanford Site,* PNNL-11800, Pacific Northwest National Laboratory, Richland, Washington.

Lewis PAW, AS Goodman, and JM Miller. 1969. "A Pseudo-Random Number Generator for the System/360." *IBM Systems Journal* 8(2) 136-145.

Mood AM, FA Graybill, and DC Boes. 1974. *Introduction to the Theory of Statistics*. Third Edition. McGraw-Hill Book Company, New York, New York.

Pike MC, and ID Hill. 1966. "Algorithm ACM 291." *Communications of the ACM* 9: 684.

Riley RG, and CA Lo Presti. 2001. *Data Catalog for Models Simulating Release of Contaminants from Hanford Site Waste Sources*. PNNL-13666, Pacific Northwest National Laboratory, Richland, Washington.

Singleton RC. 1969. Algorithm 347 in "An Efficient Algorithm for Sorting with Minimal Storage." *Communications of the Association for Computing Machinery* 12(3):185-187.

Strenge DL. 1997. "A General Algorithm for Radioactive Decay with Branching and Loss from a Medium." *Health Physics* 73(6):953-957.

Thomann RV. 1989. "Bioaccumulation Model of Organic Chemical Distribution in Aquatic Food Chains." *Environmental Science and Technology* 23:699-707.

Thomann RV, JP Connolly, and TF Parkerton. 1992. "An Equilibrium Model of Organic Chemical Accumulation in Aquatic Food Webs with Sediment Interaction." *Environmental Toxicology and Chemistry* 11: 615-629.

White MD, and M Oostrom. 2000. *STOMP Subsurface Transport Over Multiple Phases User's Guide, Version 2.0*. PNNL-12034, Pacific Northwest National Laboratory, Richland, Washington.

PNNL-13932-Volume 1

Distribution

No. of
Copies

ONSITE

DOE, Richland Operations Office

R. D. Hildebrand	A5-13

34 Pacific Northwest National Laboratory

C. Arimescu	K6-80	C. T. Kincaid	K9-33
M. P. Bergeron	K9-36	C. A. Lo Presti	K5-12
R. W. Bryce	K6-75	T. B. Miley	K6-80
D. W. Engel	K5-12	B. A. Napier	K3-54
P. W. Eslinger (21)	K6-80	W. E. Nichols	K9-33
L. H. Gerhardstein	K7-28	R. G. Riley	K6-96
B. A. Kanyid	K7-22	D. L. Strenge	K3-54