

---

**Pacific Northwest  
National Laboratory**

Operated by Battelle for the  
U.S. Department of Energy

**User Instructions for the Systems  
Assessment Capability, Rev. 1,  
Computer Codes**

**Volume 1: Inventory, Release, and  
Transport Modules**

P.W. Eslinger  
T.B. Miley  
D.W. Engel  
W.E. Nichols

L.H. Gerhardstein  
D.L. Strenge  
C.A. Lo Presti  
S.K. Wurstner

September 2004



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](mailto: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](mailto:orders@ntis.fedworld.gov)**

**online ordering: <http://www.ntis.gov/ordering.htm>**



This document was printed on recycled paper.

**User Instructions for the Systems  
Assessment Capability,  
Rev. 1, Computer Codes**

**Volume 1: Inventory, Release,  
and Transport Modules**

P. W. Eslinger  
T. B. Miley  
D. W. Engel  
W. E. Nichols

L. H. Gerhardstein  
D.L. Streng  
C. A. Lo Presti  
S. K. Wurstner

September 2004

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

Pacific Northwest National Laboratory  
Richland, Washington 99352

## **Contents**

1.0	Introduction and Background.....	1
1.1	Overview of the SAC Systems Code.....	1
1.2	Purpose of This Document .....	2
1.3	General Keyword Syntax .....	4
1.4	Stochastic Variable Generation .....	4
2.0	Environmental Settings Definition (ESD) Files.....	9
2.1	ESD Keywords.....	9
2.1.1	ANALYTE Keyword in the ESD Keyword File.....	9
2.1.2	BACKGROUND Keyword in the ESD Keyword File .....	12
2.1.3	BALANCE Keyword in the ESD Keyword File.....	12
2.1.4	CREATDIR Keyword in the ESD Keyword File.....	13
2.1.5	DEBUG Keyword in the ESD Keyword File.....	13
2.1.6	DILUTE Keyword in the ESD Keyword File .....	13
2.1.7	ECHO Keyword in the ESD Keyword File.....	14
2.1.8	END Keyword in the ESD Keyword File .....	14
2.1.9	EVAPORATE Keyword in the ESD Keyword File.....	15
2.1.10	EXEDIR Keyword in the ESD Keyword File .....	16
2.1.11	FILE Keyword in the ESD Keyword File .....	16
2.1.12	FILLECDA Keyword in the ESD Keyword File .....	18
2.1.13	INFILTRATION Keyword in the ESD Keyword File.....	19
2.1.14	IOONLY Keyword in the ESD Keyword File .....	21
2.1.15	IRRIGATE Keyword in the ESD Keyword File.....	21
2.1.16	KDSOIL Keyword in the ESD Keyword File .....	22
2.1.17	LOCATION Keyword in the ESD Keyword File .....	23
2.1.18	MODULE Keyword in the ESD Keyword File .....	27
2.1.19	OS Keyword in the ESD Keyword File .....	28
2.1.20	PERIOD Keyword in the ESD Keyword File .....	29
2.1.21	PROCESSOR Keyword in the ESD Keyword File.....	29
2.1.22	REALIZAT Keyword in the ESD Keyword File .....	30
2.1.23	REMEDIAT Keyword in the ESD File.....	30
2.1.24	REPORT Keyword in the ESD Keyword File .....	32
2.1.25	RESITE Keyword in the ESD Keyword File .....	33
2.1.26	RESTART Keyword in the ESD Keyword File.....	33
2.1.27	SITE Keyword in the ESD Keyword File .....	33
2.1.28	SPECIES Keyword in the ESD Keyword File.....	36
2.1.29	TIMES Keyword in the ESD Keyword File.....	40
2.1.30	TITLE Keyword in the ESD Keyword File .....	40
2.1.31	USER Keyword in the ESD Keyword File .....	40
2.2	Environmental Concentration Data Accumulator .....	41
2.2.1	Format of the ECDA Files.....	41
2.2.2	Format of the Record Map File for ECDA Data .....	42
2.2.3	ECDA Header file for SACView .....	44
2.3	Shared Environmental Stochastic Data .....	45

2.3.1	Format of the DILUTE Data File .....	46
2.3.2	Format of the KDSOIL Data File.....	46
2.3.3	Format of the INFILT Data File.....	47
3.0	ESP – Environmental Stochastic Preprocessor .....	49
3.1	Code Purpose.....	49
3.2	Algorithms and Assumptions .....	49
3.2.1	ESP Major Functions .....	50
3.2.2	Where Vadose Zone Flow and Transport Files Reside in SAC .....	54
3.3	Code Environment.....	57
3.3.1	How Code Is Invoked.....	57
3.3.2	Code Control and Keyword Descriptions.....	60
3.4	Data Files and Simulation Control .....	64
3.4.1	Input Files and Simulation Control .....	64
3.4.2	Output Files and Diagnostics .....	94
4.0	INVENTORY – Inventory Tracking and Disposition .....	99
4.1	Purpose99	
4.2	Algorithms and Assumptions .....	99
4.2.1	Radioactive Decay.....	99
4.2.2	Analysis Steps .....	100
4.3	Code Environment.....	101
4.3.1	Location in Processing Sequence .....	101
4.3.2	How INVENTORY Is Invoked.....	101
4.4	Keyword Descriptions for INVENTORY .....	101
4.4.1	DEBUG Keyword for INVENTORY .....	102
4.4.2	DECAY Keyword for INVENTORY .....	103
4.4.3	DISPOSAL Keyword for INVENTORY .....	103
4.4.4	END Keyword for INVENTORY.....	105
4.4.5	EXECUTE Keyword for INVENTORY .....	105
4.4.6	FILE Keyword for INVENTORY.....	105
4.4.7	MASSBALANCE Keyword for INVENTORY .....	106
4.4.8	NORMALIZE Keyword for INVENTORY.....	106
4.4.9	SEED Keyword for INVENTORY .....	107
4.4.10	SELECT Keyword for INVENTORY.....	107
4.4.11	TIMER Keyword for INVENTORY .....	108
4.4.12	TITLE Keyword for INVENTORY .....	108
4.4.13	USER Keyword for INVENTORY .....	108
4.4.14	WASTEMAP Keyword for INVENTORY .....	109
4.5	Data Files.....	110
4.5.1	Input Files.....	111
4.5.2	Output Files .....	117
4.5.3	Temporary Active Run Status File.....	126
4.6	Error Recovery Guidance.....	126
5.0	VADER – Vadose Zone Release Module .....	129
5.1	Purpose129	
5.2	Algorithms and Assumptions .....	129

5.2.1	Limitations .....	130
5.2.2	Waste Forms in VADER .....	130
5.2.3	Release Models .....	131
5.2.4	Waste Site Remediation .....	141
5.3	Code Environment.....	142
5.3.1	Location in the Processing Sequence .....	142
5.3.2	How the VADER Code Is Invoked .....	142
5.4	VADER Keyword Descriptions .....	143
5.4.1	ANALYTE Keyword for VADER.....	144
5.4.2	BYPASS Keyword for VADER.....	146
5.4.3	CONTROL Keyword for VADER.....	146
5.4.4	CUTOFF Keyword for VADER .....	147
5.4.5	DEBUG Keyword for VADER.....	147
5.4.6	DELTAT Keyword for VADER.....	148
5.4.7	END Keyword for VADER .....	149
5.4.8	FILE Keyword for VADER .....	149
5.4.9	INFILTRA Keyword for VADER .....	151
5.4.10	MAP Keyword for VADER.....	152
5.4.11	MODELS Keyword for VADER .....	153
5.4.12	NOCOMPUTE Keyword for VADER.....	159
5.4.13	PATH Keyword for VADER .....	159
5.4.14	PERIOD Keyword for VADER .....	159
5.4.15	REALIZATION Keyword for VADER.....	160
5.4.16	REMEDIAT Keyword for VADER.....	160
5.4.17	SITE Keyword for VADER .....	163
5.4.18	STEPS Keyword for VADER .....	164
5.4.19	TITLE Keyword for VADER.....	164
5.5	Data Files.....	165
5.5.1	VADER Input files.....	166
5.5.2	VADER Output Files .....	173
5.6	Consistency and Traceability Checks.....	180
6.0	VZDROP – Vadose Zone to Groundwater Mass Transfer Module.....	183
6.1	Code Purpose.....	183
6.2	Algorithms and Assumptions .....	183
6.2.1	Nearest CFEST Node Determination for Vadose Zone Sites.....	183
6.2.2	Distribution of Analyte Releases from Vadose Zone Sites to Groundwater ....	184
6.2.3	A Note About Time-Drift Effect.....	186
6.2.4	Major Assumptions .....	187
6.3	Code Environment.....	187
6.3.1	Location in Processing Sequence .....	188
6.3.2	How Code Is Invoked.....	188
6.3.3	Code Control and Keyword Descriptions.....	188
6.4	Data Files.....	190
6.4.1	Input Files.....	190
6.4.2	Output Files .....	192

7.0	GWDROP – Groundwater to River Mass Transfer Module .....	197
7.1	Code Purpose.....	197
7.2	Algorithms and Assumptions .....	197
7.2.1	CFEST Node Polygons Geometry .....	197
7.2.2	MASS2 Cell Geometry .....	199
7.2.3	POLYCEN Mode for Flux Transfer from CFEST to MASS2 .....	200
7.2.4	POLYINT Mode for Flux Transfer from CFEST to MASS2 .....	201
7.2.5	Balance Diagnostics .....	201
7.2.6	COV and TMS File Naming Algorithms .....	202
7.2.7	Time and Delta-Time Algorithms .....	203
7.2.8	Concentration Data Extraction .....	203
7.3	Code Execution Environment .....	204
7.3.1	Location in the Processing Sequence .....	204
7.3.2	How the GWDROP Code Is Invoked.....	204
7.3.3	Code Control and Keyword Descriptions.....	205
7.4	Data Files.....	207
7.4.1	Input Files.....	207
7.4.2	Output Files .....	212
7.4.3	Example POLYGONS.DAT File .....	215
8.0	AIRDROP – Atmospheric Concentration Extraction Module .....	217
8.1	Code Purpose.....	217
8.2	Algorithms and Assumptions .....	217
8.3	Code Environment.....	217
8.3.1	Location in Processing Sequence .....	217
8.3.2	How the AIRDROP Code Is Invoked .....	217
8.3.3	Code Control and Keyword Descriptions.....	218
8.4	Data Files.....	220
8.4.1	Input Files.....	221
8.4.2	Output Files .....	223
9.0	CRDROP – Columbia River Concentration Extraction Module .....	229
9.1	Code Purpose.....	229
9.2	Algorithms and Assumptions .....	229
9.3	Code Environment.....	229
9.3.1	Location in the Processing Sequence .....	229
9.3.2	How the CRDROP Code Is Invoked.....	229
9.3.3	Code Control and Keyword Descriptions.....	230
9.4	Data Files.....	232
9.4.1	Input Files.....	232
9.4.2	Output Files .....	236
9.5	CRDROP_INDEX Utility Program .....	238
9.5.1	Code Purpose.....	238
9.5.2	Algorithms and Assumptions .....	238
9.5.3	Code Environment.....	238
9.5.4	Keyword Descriptions for CRDROP_INDEX.....	239
9.5.5	Data Files.....	241

10.0	RIPSAC – Riparian Zone Module .....	245
10.1	Purpose	245
10.2	Algorithms and Assumptions .....	245
10.3	Code Execution Environment .....	245
10.3.1	Location in the Processing Sequence .....	245
10.3.2	How the Code Is Invoked.....	246
10.4	Keyword Descriptions for RIPSAC .....	246
10.4.1	ANALYTE Keyword for RIPSAC.....	247
10.4.2	DEBUG Keyword for RIPSAC.....	247
10.4.3	END Keyword for RIPSAC .....	247
10.4.4	FILE Keyword for RIPSAC .....	247
10.4.5	KDSOIL Keyword for RIPSAC.....	248
10.4.6	LOCATION Keyword for RIPSAC .....	248
10.4.7	REALIZATION Keyword for RIPSAC.....	249
10.4.8	REPORT Keyword for RIPSAC .....	250
10.4.9	TITLE Keyword for RIPSAC .....	250
10.4.10	USER Keyword for RIPSAC .....	250
10.5	Data Files.....	250
10.5.1	Input Files.....	250
10.5.2	Output Files .....	252
11.0	SOIL – Soil Accumulation Module .....	257
11.1	Purpose	257
11.2	Algorithms and Assumptions .....	257
11.3	Code Execution Environment .....	258
11.3.1	Location in the Processing Sequence .....	258
11.3.2	How the Code Is Invoked.....	258
11.4	Keyword Descriptions for SOIL .....	259
11.4.1	ANALYTE Keyword for SOIL.....	260
11.4.2	DEBUG Keyword for SOIL.....	260
11.4.3	END Keyword for SOIL .....	260
11.4.4	EXECUTE Keyword for SOIL .....	261
11.4.5	FILE Keyword for SOIL .....	261
11.4.6	KDSOIL Keyword for SOIL.....	261
11.4.7	LOCATION Keyword for SOIL .....	262
11.4.8	REALIZATION Keyword for SOIL.....	263
11.4.9	REPORT Keyword for SOIL .....	264
11.4.10	SOIL Keyword for SOIL.....	264
11.4.11	TITLE Keyword for SOIL .....	264
11.4.12	USER Keyword for SOIL .....	265
11.5	Data Files.....	265
11.5.1	Input Files.....	265
11.5.2	ESD Keyword File .....	265
11.5.3	SOIL Keyword File.....	266
11.5.4	KDSOIL Stochastic Data .....	266
11.5.5	INFILT Stochastic Data .....	267



11.5.6	ECDA Files .....	267
11.6	Output Files .....	267
11.6.1	SOIL Report File .....	267
11.6.2	“soil.run” File .....	270
11.6.3	“soil.done” File .....	270
11.6.4	Concentration Data Files .....	270
12.0	Keyword Language Syntax .....	271
12.1	Keyword Records .....	271
12.2	Continuation Records .....	271
12.3	Comment Records .....	272
12.4	Input Data Handling .....	272
12.4.1	Quote Strings .....	272
12.4.2	Data Separators .....	273
13.0	References .....	275

## **Figures**

Figure 1.1	Module Information Flow for the SAC Rev. 1 Systems Code .....	2
Figure 3.1	Structure of the /vadose directory .....	55
Figure 3.2	Example /vadose zone Subdirectory .....	56
Figure 3.3	Using the ESP to Generate STOMP Input Files .....	75
Figure 3.4	Sample Template Directory Structure .....	76
Figure 3.5	Structure of the /cfest Subdirectory .....	79
Figure 3.6	Using the ESP to Generate CFEST Control Files .....	82
Figure 6.1	Depiction of Hypothetical CFEST and STOMP to Describe the Backing Up Effect ....	187
Figure 7.1	Example Nodes and Elements Layout .....	198
Figure 7.2	Example with Nodal Polygons Shaded .....	199
Figure 7.3	MASS2 River Cell Layout .....	200

## **Tables**

Table 2.1	Modifiers for the ANALYTE Keyword in the ESD File .....	10
Table 2.2	Modifiers for the EVAPORATE Keyword in the ESD File .....	15
Table 2.3	Modifiers for the FILE Keyword in the ESD File .....	17
Table 2.4	Modifiers for the LOCATION Keyword in the ESD File .....	18
Table 2.5	Modifiers for the IRRIGATE Keyword in the ESD File .....	22
Table 2.6	Modifiers for the LOCATION Keyword in the ESD File .....	23
Table 2.7	MODULE Options in the ESD File .....	27
Table 2.8	Modifiers for the REMEDIAT Keyword in the ESD File .....	31
Table 2.9	Modifiers for the SITE Keyword in the ESD File .....	34
Table 2.10	Modifiers for the SPECIES Keyword in the ESD File .....	36
Table 2.11	Structure of an ECDA File .....	41
Table 2.12	Excerpts Record Index Map File .....	43
Table 2.13	Example ECDA Header File for use in SACView .....	44

Table 2.14 Example DILUTE File.....	46
Table 2.15 Example KDSOIL File .....	46
Table 2.16 Example INFILT File .....	47
Table 3.1 Batch Procedure ../processors/p121.com .....	58
Table 3.2 Batch Procedure ../assessment/processors/p002.com .....	58
Table 3.3 Batch Procedure ../assessment/processors/p005.com .....	59
Table 3.4 Example ESD Keyword File for Use by ESP .....	60
Table 3.5 Example File ../assessment/vadose/600-148/template_vader.key .....	65
Table 3.6 Example File ../assessment/vadose/600-148/stochastic_vader.key .....	65
Table 3.7 Example STOMP Control File input-esp.....	68
Table 3.8 Example File ../assessment/vadose/600-148/template_stomp.key .....	71
Table 3.9 Example File ../assessment/vadose/stochastic_stomp.key.....	72
Table 3.10 Sample CFEST Control File ../assessment/cfest/H3/cfest.key .....	78
Table 3.11 Sample Stochastic Definition File for CFEST ../assessment/cfest/stochastic.key.....	78
Table 3.12 Sample script file to populate an assessment with CFEST flow solutions .....	81
Table 3.13 Recommended Parameter Values for the cfest.ctl File for Flow Solutions and the cfest.key Template File for SAC Transport Solutions .....	83
Table 3.14 Example gwdrop.key file for Use in ESP .....	84
Table 3.15 Example gwdrop.inp file prepared by ESP.....	85
Table 3.16 Sample RATCHET Control File ../assessment/ratchet/metdata/ratchet.ctl .....	86
Table 3.17 Sample Stochastic Definition File for RATCHET ../assessment/ratchet/stochastic.key.....	86
Table 3.18 Example index.key File Used by ESP for CRDROP_INDEX .....	87
Table 3.19 Example File mass2.key Used by ESP in Preparing MASS2 Input Files.....	88
Table 3.20 Example biota.stoch File for Use in ESP.....	88
Table 3.21 Example biota.key File for Realization 1 of Tritium for Use by MASS2 .....	90
Table 3.22 Example CRDROP Keyword File Written by ESP .....	92
Table 3.23 Example File ripsac.key Used in the RIPSAC Code .....	92
Table 3.24 Example File soil.key Used in the SOIL Code .....	93
Table 4.1 Summary of Keywords Used by INVENTORY .....	102
Table 4.2 Modifiers for the DISPOSAL Keyword in INVENTORY .....	104
Table 4.3 Modifiers for the FILE Keyword in INVENTORY.....	106
Table 4.4 Files Used by INVENTORY Code.....	110
Table 4.5 Example inventory.key File.....	111
Table 4.6 ESD File Keyword Information Used by INVENTORY .....	112
Table 4.7 Example Records from a Disposal Action Keyword File.....	113
Table 4.8 Example Waste Stream Selection File for INVENTORY .....	115
Table 4.9 Format of Radionuclide Master Decay Data File .....	115
Table 4.10 Example Data Set for the Radionuclide Master Decay Data File.....	116
Table 4.11 Excerpted Records from the inventory.out File .....	118
Table 4.12 Example Mass Balance Result File from INVENTORY.....	122
Table 4.13 Format of Result Output Files (.RES) in INVENTORY .....	122
Table 4.14 Example Result File from INVENTORY.....	123
Table 4.15 Data Record Descriptions for the INVENTORY.ALL File.....	124
Table 4.16 Example Records for a, INVENTORY.ALL File.....	125
Table 4.17 Example Normalization Result File.....	125

Table 5.1 Summary of release models implemented in VADER .....	131
Table 5.2 Waste Forms Processed by VADER.....	133
Table 5.3 Summary of Keywords Used by VADER .....	143
Table 5.4 Modifiers for the ANALYTE Keyword in VADER.....	145
Table 5.5 Modifiers for the DEBUG Keyword in VADER.....	148
Table 5.6 Overview of Files Used by VADER.....	165
Table 5.7 Example vader.key File .....	166
Table 5.8 Example Inventory Results File.....	167
Table 5.9 WASTE Form to Release Model Mapping in VADER.....	168
Table 5.10 Example STOMP-built Remediation File.....	169
Table 5.11 Example VADER-Built Remediation File.....	169
Table 5.12 Example BYPASS file for VADER .....	170
Table 5.13 BYPASS file header records specifications for VADER .....	171
Table 5.14 BYPASS file data records description for VADER.....	171
Table 5.15 Excerpted Records from an input-esp File Modified by VADER .....	172
Table 5.16 Excerpted Records from a vader.table File.....	174
Table 5.17 Field Descriptions for the vader.table file.....	175
Table 5.18 Excerpted Records from a vader.river File .....	176
Table 5.19 Excerpted Records from a vader.log File.....	177
Table 5.20 Example vader.done File .....	180
Table 6.1 Example SAC Header for STOMP Input/Output Files.....	184
Table 6.2 VZDROP Keyword File Example 1 .....	189
Table 6.3 VZDROP Keyword File Example 2 .....	190
Table 6.4 Example STOMP Release File .....	191
Table 6.5 Example VZDROP Log File.....	194
Table 7.1 Definition of Fixed Content Lines in the GWDROP.LIS File.....	205
Table 7.2 Definition of Variable Context Lines for the GWDROP.LIS File.....	206
Table 7.3 Example GWDROP.LIS File for I129, Realization 21 .....	207
Table 7.4 First 21 Lines from a MASS2 River Cell File .....	211
Table 7.5 Excerpts from a GWDROP Log File .....	213
Table 7.6 Excerpted Records from a Water Flow TMS file for Serial Number 10727 .....	214
Table 7.7 Excerpted Records from a Flux TMS file for Serial Number 10727 .....	214
Table 7.8 Example COV File Written by GWDROP .....	215
Table 7.9 Example Nodal Polygons Output File from GWDROP .....	215
Table 8.1 Description and Order of Keywords Used by the AIRDROP Code.....	218
Table 8.2 AIRDROP Keyword Control File Example.....	219
Table 8.3 Files Accessed by the AIRDROP Code.....	220
Table 8.4 Example Release Location File .....	221
Table 8.5 Example STOMP Airborne Release File .....	223
Table 8.6 Example Log File for AIRDROP .....	223
Table 9.1 Summary of Keywords Used by the CRDROP Code.....	230
Table 9.2 Example Keyword File for CRDROP.....	233
Table 9.3 Content and Format of the Location Cross-Index File .....	233
Table 9.4 Example Cross-Index File for CRDROP .....	234
Table 9.5 Structure of MASS2 River Model Output Files.....	235

Table 9.6 Excerpted Records from a MASS2 River File.....	235
Table 9.7 Example Report File for CRDROP.....	236
Table 9.8 Files Accessed by the CRDROP_INDEX Code.....	241
Table 9.9 Example Keyword File for CRDROP_INDEX.....	242
Table 9.10 Excerpted Records from a River Model Grid Location File.....	243
Table 9.11 Example Report file from CRDROP_INDEX.....	243
Table 10.1 Summary of Keywords Used by RIPSAC.....	246
Table 10.2 Modifiers for the REALIZATION Keyword in RIPSAC.....	249
Table 10.3 Example RIPSAC Keyword File.....	251
Table 10.4 Excerpted Records from a RIPSAC Report File.....	252
Table 11.1 Summary of Keywords Used by SOIL.....	259
Table 11.2 Modifiers for the DEBUG Keyword in SOIL.....	260
Table 11.3 Modifiers for the KDSOIL Keyword in SOIL.....	262
Table 11.4 Modifiers for the LOCATION Keyword in SOIL.....	263
Table 11.5 Modifiers for the REALIZATION Keyword in SOIL.....	263
Table 11.6 ESD Keywords Used by SOIL.....	265
Table 11.7 Example SOIL Keyword File.....	266
Table 11.8 Excerpts from a SOIL Report File.....	267

## **1.0 Introduction and Background**

In 1999, the U.S. Department of Energy (DOE) initiated the development of an assessment tool that will enable users to model the movement of contaminants from all waste sites at Hanford through the vadose zone, groundwater and Columbia River and estimate the impact of contaminants on human health, ecology and the local cultures and economy. This tool was named the System Assessment Capability (SAC). This tool is an integrated system of computer models and databases used to assess the impact of waste remaining on the Hanford Site. The SAC will help decision makers and the public evaluate the cumulative effects of contamination from Hanford.

The design of the SAC resulted from extensive interactions with Hanford projects, regulators, Tribal Nations, and stakeholders. The approach taken in the assessment follows that advanced by regulatory agencies such as the U.S. Environmental Protection Agency (EPA) in their guidance on uncertainty analyses (Firestone et al. 1997) and ecological risk assessments (EPA 1998), and DOE in its radioactive waste management manual (DOE 1999a) and implementation guide (DOE 1999b), which support DOE Order 435.1 on radioactive waste management. The SAC also was designed with the intent to use it to perform the next composite analysis, an assessment first performed to satisfy Defense Nuclear Facility Safety Board recommendation 94-2. The approach taken is also consistent with the methods, characteristics, and controls associated with acceptable analyses as described by the Columbia River Comprehensive Impact Assessment (CRCIA) team (DOE 1998).

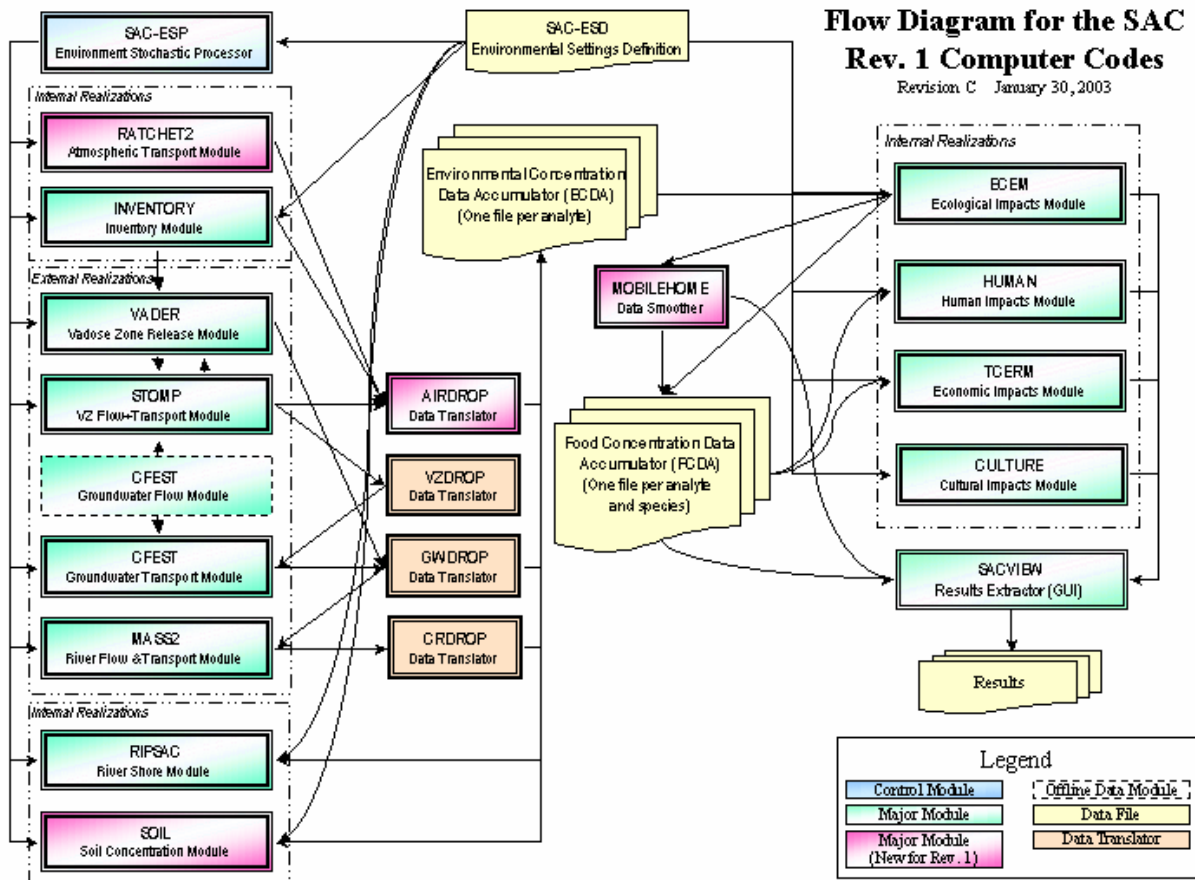
### **1.1 Overview of the SAC Systems Code**

The SAC Systems Code is a tool used to simulate the migration of contaminants (analytes) present on the Hanford Site and 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 1.1.

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, the proximity of contaminants to social and cultural resources.

The general approach to handling uncertainty in SAC, Rev. 1, 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*. The entire process is then repeated, 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 the 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.



**Figure 1.1** Module Information Flow for the SAC Rev. 1 Systems Code

## 1.2 Purpose of This Document

The SAC codes on the left side of Figure 1.1 address inventory tracking, release of contaminants to the environment, and transport of contaminants through the atmosphere, unsaturated zone, saturated zone, and the Columbia River. This document contains detailed user instructions for the computer codes described in Table 1.1. Instructions for some of the codes on the left side of Figure 1.1 and a number of utility codes are not provided in this document. The status of user instructions for these codes is shown in Table 1.2.

Table 1.1. Overview of Transport Codes

Name	Purpose
SAC ESP	Environmental stochastic processor. This processor controls the execution of all the codes on the left side of Figure 1.1.
INVENTORY	Inventory tracking and aggregation code.
VADER	Vadose zone release module. The function of this code includes release of contaminants from the waste form and 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.
AIRDROP	Utility code to extract results from the atmospheric transport module (RATCHET), scale them by inventory and vadose zone releases to the atmosphere, and save them 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.
SOIL	Soil concentration module. The function of this code is to calculate soil concentrations in the upper soil layer for locations outside waste disposal areas.

Table 1.2. Description of Codes Not Provided in this Document

Name	Purpose
STOMP	A standalone user's guide is provided in White and Oostrom (2000). Section 3.4.1.7 documents the few code changes made to integrate STOMP into the SAC framework.
CFEST96	A user's guide for the CFEST96 code has been published separately (CFEST 2000).
MASS2	Theory and numerical methods for the MASS2 code are provided in Perkins and Richmond (2004a), and a user's guide is provided in Perkins and Richmond (2004b).
RATCHET	A user's guide is provided in Ramsdell, et. al (1994). Section 3.4.1.15 documents the changes made to integrate RATCHET into the SAC framework.

The suite of computer codes for Rev. 1 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 regional economic impacts assessment. User instructions for the impacts codes are provided in Eslinger et al. (2004a). User instructions for a suite of utility codes are documented in Eslinger et al. (2004b).

The assumption is made that users of this document are knowledgeable computer users. In this case, the computer system runs under 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.

### **1.3 General Keyword Syntax**

Many of the programs for SAC Rev. 1 are controlled through the use of data files containing text entries called keywords. In the keyword descriptions, some data are optional for a particular problem definition and some are required. Data that are required are enclosed in square brackets. For example, 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 one or more 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. Quote strings must be enclosed in double quotation marks. For example:

```
FILE C_ECDA ANALYTE="U" NAME="/home/CPO/ecda/U_CPO.dat" CREATE
```

### **1.4 Stochastic Variable Generation**

Many of the codes in SAC, Rev. 1, 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.

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<sub>1</sub>"] [Dist\_Index Parameters] {TRUNCATE U1 U2} {"Quote<sub>2</sub>"}

The entry for Quote<sub>1</sub> 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 are needed. The entry for Quote<sub>2</sub> 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<sub>2</sub> 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 1.3 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 1.3. Statistical Distributions Available in All 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.

<b>Index</b>	<b>Distribution</b>	<b>Truncate</b>	<b>Parameters Required</b>
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 (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 100.E-7 to 10.0E-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 10.0E+3 to 10.0E+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 <sup>14</sup>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  $\alpha$  and the second parameter be denoted by  $\beta$ . The mean of the beta distribution would be  $\alpha/(\alpha+\beta)$  if the limits were 0 and 1. Both  $\alpha$  and  $\beta$  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.

## **2.0 Environmental Settings Definition (ESD) Files**

As seen in Figure 1.1, the SAC Rev. 1 systems code contains a number of component models that are executed independently. Some 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 contains 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 impact 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. 1, 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 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**

The ESD keyword file is read by a number of different programs. Data required by one program may not be used in another program; thus, specifications of data being optional or required are not made in this section.

#### **2.1.1 ANALYTE Keyword in the ESD Keyword File**

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

```
ANALYTE ID="quote 1" TYPE="quote 2" NAME="quote 3" AIR= "quote 4"  
HENRY=N2 DFIMM=N3 DFSED=N4} GAMMA=N5 HALFLIFE=N6 MOLWGT=N7  
SPECIFIC=N8 GASDIFF=N9 MOLDIFF=N1 COMPUTE
```

A separate ANALYTE keyword must be entered for every analyte to be included in the simulation. Table 2.1 describes the modifiers for the ANALYTE keyword.

**Table 2.1** Modifiers for the ANALYTE Keyword in the ESD File

<b>Modifier</b>	<b>Description</b>
AIR	<p>The quote string associated with the AIR modifier is a string up to 8 characters in length defining the analyte type for atmospheric transport and deposition. The following are the valid entries for this string:</p> <ul style="list-style-type: none"> <li>• NOBLE – if the analyte is to be treated as a noble gas in the atmospheric transport module (atmospheric transport only, no deposition)</li> <li>• PARTICLE – if the analyte is to be treated in the atmospheric transport module as one that deposits as a particle (atmospheric transport and deposition)</li> <li>• IODINE – if the analyte is iodine (atmospheric transport and deposition characteristics unique to iodine)</li> </ul>
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.
DFIMM	The numerical entry associated with the DFIMM modifier is the immersion dose factor for radioactive analytes. This value has units of mrad/yr per $\mu\text{Ci}/\text{m}^3$ . 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 $\text{Sv}\cdot\text{m}^3/\text{sec}\cdot\text{Bq}$ . 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. Entry of this modifier is optional. If it is not present, the value of GAMMA defaults to zero.
GASDIFF	The numerical entry associated with the GASDIFF modifier is the gas phase molecular diffusivity. This value has units of $\text{cm}^2/\text{sec}$ . Entry of this modifier is optional. If it is not present, the value of GASDIFF 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).
HENRY	The numerical entry associated with the HENRY modifier is the Henry's law coefficient for organic analytes. This value has units of $\text{Pa}\cdot\text{m}^3/\text{mole}$ . Entry of this modifier is optional. If it is not present, the value of HENRY defaults to zero.

Modifier	Description
ID	<p>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:</p> <ul style="list-style-type: none"> <li>• Only the first entry in the analyte identification string is capitalized</li> <li>• No embedded spaces or hyphens are used, even for radionuclides</li> <li>• Individual elements are defined using the standard element abbreviation</li> </ul>
MOLDIFF	<p>The numerical entry associated with the MOLDIFF modifier is the molecular diffusivity of the analyte. This value has units of cm<sup>2</sup>/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.</p>
MOLWGT	<p>The numerical entry associated with the MOLWGT modifier is the molecular weight of the analyte. This value has units of g/mole.</p>
NAME	<p>The quote string associated with the NAME modifier is an analyte name or description up to 72 characters in length.</p>
SPECIFIC	<p>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 but should be omitted for nonradioactive analytes.</p>
TYPE	<p>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:</p> <ul style="list-style-type: none"> <li>• NR – if the analyte is a radioactive element or an inorganic compound containing a radionuclide</li> <li>• NS – if the analyte is a stable (nonradioactive) element or inorganic compound</li> <li>• OR – if the analyte is an organic compound containing a radionuclide</li> <li>• OS – if the analyte is an organic compound, containing a stable (nonradioactive) elemental analyte or compound</li> </ul>

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* should 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 following ANALYTE keywords select the analytes carbon tetrachloride, strontium-90 and carbon-14 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=8.990774E+01   HALFLIFE=28.78   SPECIFIC=1.382778E+02
      DFIMM=1.703575E-03   DFSED=3.72E-21   GAMMA=0.0
      MOLDIFF=1.05E-05   GASDIFF=0.0   HENRY=0.0
```

```
ANALYTE ID="C14"   NAME="Carbon-14"   TYPE="NR"   AIR="NOBLE"   COMPUTE
      MOLWGT=1.400324E+01   HALFLIFE=5.715E+03   SPECIFIC=4.470906E+00
      DFIMM=5.122392E-05   DFSED=7.200000E-23   GAMMA=0.0
      MOLDIFF=1.05E-05   GASDIFF=0.0   HENRY=0.0
```

### **2.1.2 BACKGROUND Keyword in the ESD Keyword File**

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

```
BACKGROUND COMPUTE
```

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

### **2.1.3 BALANCE Keyword in the ESD Keyword File**

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

```
BALANCE [ALL | [year1] {year2} {year3} ... {yearn}]
```

If the modifier ALL is present, then mass balance information is saved for all years defined in the PERIOD keyword record (Section 2.1.20) from the start year through 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
```



### **2.1.4 CREATDIR Keyword in the ESD Keyword File**

The CREATEDIR keyword tells ESP to create the subdirectory structure for the analysis. The following is this 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 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 2.1.18), ANALYTE (see Section 2.1.1), and SITE (see Section 2.1.27).

### **2.1.5 DEBUG Keyword in the ESD Keyword File**

The DEBUG keyword initiates output of optional data. This keyword has two uses for two separate codes. The following is this keyword's syntax:

```
DEBUG STOCHASTIC="quote"  
DEBUG ECDA
```

The quote string associated with the STOCHASTIC modifier identifies a file name (up to 200 characters in length) in which to store the stochastic values generated by the 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. 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"
```

If the ECDA modifier is present it initiates output of extra information to the report file when the ECDA code is executed. The keyword entry for that action is as follows:

```
DEBUG ECDA
```

With the exception of the ECDA code, the debug keyword in the ESD keyword file is only used by the ESP program. Debug options in other codes are activated by specific inputs for those codes contained in other files.

### **2.1.6 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 used in the riparian zone water mixing model. The following is this keyword's syntax:

```
DILUTE [ID="quote1"] [Dist_Index Parameters] {TRUNCATE U1 U2}  
{LABEL="quote2"} [UNITS="quote3"]
```

The quote string associated with the ID modifier is a unique character string of up to 20 characters that will be used to identify this stochastic variable for subsequent uses. It is case sensitive and embedded spaces are significant. The quote string associated with the optional modifier LABEL contains a description for the stochastic variable that can be up to 64 characters long. An entry for quote2 is not required, although it is used for labeling purposes if present. However, if the modifier LABEL is present the associated quote string must be entered as well. The quote string associated with the UNITS modifier contains a units descriptor for the data. The strings "none" or "unitless" should be used if the variable is unitless. Section 1.4 contains further information about generating stochastic values.

More than one DILUTE keyword may be entered. The set of DILUTE keywords is used to generate a library of stochastic values that can be accessed by any SAC code. At this time only the RIPSAC code uses these values. The name of the library file is defined using the DILUTE modifier on the FILE keyword (Section 2.1.11).

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

```
DILUTE ID="ID#1" 6 0.2 0.5 0.99 LABEL="Example dilution factor"  
UNITS="none"
```

### **2.1.7 ECHO Keyword in the ESD Keyword File**

The ECHO keyword is used to initiate output of summary information by the ECDA utility code when processing the statistical distributions defined by KDSOIL and DILUTE keywords. The following is this keyword's syntax:

```
ECHO {KDSOIL} {DILUTE}
```

If the KDSOIL modifier is present then variable definitions and summary statistics on generated values will be written to the report file for every distribution specified on a KDSOIL keyword. If the DILUTE modifier is present then variable definitions and summary statistics on generated values will be written to the report file for every distribution specified on a DILUTE keyword. Example uses of this keyword are the following:

```
ECHO KDSOIL  
ECHO DILUTE  
ECHO DILUTE KDSOIL
```

### **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 following is this keyword's syntax:

```
END
```

### 2.1.9 EVAPORATE Keyword in the ESD Keyword File

The EVAPORATE keyword defines the evaporation rates applied to liquid disposal sites. The following is this keyword's syntax:

```
EVAPORATE [SITE="quote1"] [START="quote2"] [END="quote3"]
          [VOLUME=N1] [UNITS="quote4"] {COMPUTE}
```

The modifiers for this keyword are described in Table 2.2.

**Table 2.2** Modifiers for the EVAPORATE Keyword in the ESD File

Modifier	Description
SITE	The quote string associated with the SITE modifier contains the site ID for the specified site.
START	The quote string associated with the START modifier is the date (in format "mm/dd/yyyy") when evaporation begins.
END	The quote string associated with the END modifier is the date (in format "mm/dd/yyyy") when evaporation ends.
VOLUME	The numerical entry associated with the VOLUME modifier is the volume to be evaporated from the liquid disposal. The units of this value are those declared in the UNITS modifier.
UNITS	The quote string associated with the UNITS modifier contains a units descriptor for the data.
COMPUTE	The optional modifier COMPUTE can be entered. If COMPUTE is not present, the evaporation will not be included in the run.

Multiple EVAPORATE keywords can be entered. They can define different time periods for the same site or they can define data for different sites. The order of record dates must be arranged in ascending order for multiple keywords defining different time periods for a specific site. Example uses of this keyword are the following:

```
EVAPORATE SITE="216-B-3" START="01/01/1945" END="01/02/1945"
          VOLUME=14881.09 UNITS="m^3" compute
EVAPORATE SITE="216-B-3" START="01/02/1945" END="01/03/1945"
          VOLUME=21125.85 UNITS="m^3" compute
EVAPORATE SITE="216-B-3" START="01/03/1945" END="01/04/1945"
          VOLUME=36644.92 UNITS="m^3" compute
EVAPORATE SITE="216-B-3" START="01/04/1945" END="01/05/1945"
          VOLUME=51527.04 UNITS="m^3" compute
EVAPORATE SITE="216-B-3" START="01/05/1945" END="01/06/1945"
          VOLUME=66307.8 UNITS="m^3" compute
```

In the event that the declared quantity to evaporate exceeds the available water from infiltration and liquid disposals, the quantity will be truncated to match the available water.

### **2.1.10 EXEDIR Keyword in the ESD Keyword File**

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

```
EXEDIR ["quote"]
```

The EXEDIR keyword is used to identify the directory where executable programs reside that get called from the main processor ESP or from a script prepared by ESP. Example programs include the following:

- ecda-1.exe** for setup of the concentration data files for later use by the impacts codes
- ripsac-1.exe** for calculation of riparian zone concentrations
- soil-1.exe** for calculation of upper soil layer soil concentrations
- back\_mass2-1.exe** for setup of background concentration data files for use in the river transport model
- inventory-1.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 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 ESP. For instance, on a Unix system, the user can specify a path name in the user's definition file (.cshrc).

### **2.1.11 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 following is this keyword's syntax:

```
FILE [NAME="quote1"] {ANALYTE="quote2"} {SEED=N1}  
  [ HEADER | KDSOIL | DILUTE | INFILT | C_ECDA | I_ECDA ] {CREATE}
```

The file names must be entered with complete path information. Every file definition requires the entry of a separate FILE keyword. One FILE keyword is required for every analyte for which concentrations are to be generated. The modifiers for this keyword are described in Table 2.3.

**Table 2.3** Modifiers for the FILE Keyword in the ESD File

Modifier	Description
ANALYTE	If the file name is associated with the C_ECDA modifier, the ANALYTE keyword must also be entered. The quote string associated with the ANALYTE modifier contains the ID for the analyte associated with the ECDA concentration data file (see the ANALYTE keyword in Section 2.1.1).
CREATE	If this modifier is present the file will be created during a code execution. If the CREATE modifier is not present no file creation actions occur for the file.
C_ECDA	This modifier indicates that the FILE keyword is defining an ECDA file. Data 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 2.1.1).
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.
INFILT	This modifier indicates that the FILE keyword is defining a file to contain stochastic realizations of all of the random variables defined using the INFILTRATION keyword.
I_ECDA	This modifier indicates that the FILE keyword is defining a record index file for mapping into all ECDA concentration files.
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.
NAME	The quote string associated with the NAME modifier contains a file name (including path information) up to 200 characters in length.
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 a suite of analytes:

```
FILE C_ECDA ANALYTE="H3" NAME="/test/ecda/H3_median.dat" CREATE
FILE C_ECDA ANALYTE="C14" NAME="/test/ecda/C14_median.dat" CREATE
FILE C_ECDA ANALYTE="I129" NAME="/test/ecda/I129_median.dat" CREATE
FILE C_ECDA ANALYTE="Tc99" NAME="/test/ecda/Tc99_median.dat" CREATE
```

If present, the CREATE flag causes the following actions by the ECDA code:

- Deletion of any existing file by that name and creation of a new file.
- If the file is associated with the HEADER, DILUTE, KDSOIL, C\_ECDA or I\_ECDA modifiers, new data are written to the file.

The following example entries define the files for soil-water Kd values, the water dilution factors for the river-shore module, and the groundwater infiltration rates for the soil module:

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

The following example entries define the SACView Header file and ECDA record number index files:

```
FILE HEADER NAME="/test/ecda/Sacview_median.hdr" CREATE
FILE I_ECDA NAME="/test/ecda/ECDA_median.map" CREATE
```

### 2.1.12 FILLECDA Keyword in the ESD Keyword File

The FILLECDA keyword tells the ECDA code which values to use for initialization when an ECDA file is created. The following is this keyword's syntax:

```
FILLECDA [GWAT | SWAT | AIRC | AIRD | PWAT | SEDI | SORP | SEEP | SODR | SOGW | SOSW]
[ FIXED=N1 | RANDOM]
```

The modifiers for this keyword are described in Table 2.4.

**Table 2.4** Modifiers for the LOCATION Keyword in the ESD File

Modifier	Description
AIRC	Presence of this optional modifier indicates that atmospheric concentrations will be initialized with the specified value.
AIRD	Presence of this optional modifier indicates that atmospheric deposition rates will be initialized with the specified value.
FIXED	The value $N_1$ associated with the FIXED modifier is the value to be used to initialize the specified media in the ECDA file. Concentrations are typically initialized to zero, or an invalid (-1.0) value.
GWAT	Presence of this optional modifier indicates that groundwater concentrations will be initialized with the specified value.
PWAT	Presence of this optional modifier indicates that river bottom pore water concentrations will be initialized with the specified value.
RANDOM	The RANDOM option is used only for testing purposes. When RANDOM is selected the media concentrations are all filled with values from a uniform distribution on the (0,1) interval.
SEDI	Presence of this optional modifier indicates that sediment concentrations (on the river bottom) will be initialized with the specified value.
SEEP	Presence of this optional modifier indicates that seep water concentrations (on the land surface) will be initialized with the specified value.

Modifier	Description
SODR	Presence of this optional modifier indicates that upland soil concentrations from non-irrigated soils will be initialized with the specified value.
SOGW	Presence of this optional modifier indicates that upland soil concentrations using groundwater for irrigation will be initialized with the specified value.
SORP	Presence of this optional modifier indicates that riparian zone soil concentrations (on the land surface) will be initialized with the specified value.
SOSW	Presence of this optional modifier indicates that upland soil concentrations using surface water for irrigation will be initialized with the specified value.
SWAT	Presence of this optional modifier indicates that surface water concentrations will be initialized with the specified value.

The following example entries assign values to all 11 media for initialization of the ECDA files:

```
FILLECDA GWAT FIXED = 0.0
FILLECDA PWAT FIXED = 0.0
FILLECDA SWAT FIXED = 0.0
FILLECDA SEEP FIXED = -1.0
FILLECDA SEDI FIXED = 0.0
FILLECDA SORP FIXED = -1.0
FILLECDA SODR FIXED = -1.0
FILLECDA SOGW FIXED = -1.0
FILLECDA SOSW FIXED = -1.0
FILLECDA AIRC FIXED = 0.0
FILLECDA AIRD FIXED = 0.0
```

### 2.1.13 INFILTRATION Keyword in the ESD Keyword File

The INFILTRATION keyword defines the stochastic distribution for groundwater infiltration rates. If the vadose zone is being modeled, an infiltration rate or rates must be specified for each site (as specified by the SITE keyword) sufficient to cover all simulated time. Infiltration classes are also defined for impact locations with this keyword as well. The keyword may have one of two general syntaxes; the first syntax is used to define an “infiltration class” and an associated infiltration rate or stochastic distribution of rates. The second general syntax is used to apply an infiltration class to a specific vadose zone site.

The two syntaxes are distinguished by the presence of the modifier “DEFINE” or the modifier “SITE”. One, and only one, of these two modifiers must be used with the INFILTRATION keyword. There are two ways to define an infiltration class. The first way to define an infiltration class explicitly declares an infiltration class and rate using the following syntax:

```
INFILTRATION DEFINE [CLASS="quote 1"] [UNITS="quote 2"]
[Dist_Index Parameters] {TRUNCATE U1 U2} {"quote 3"}
```

The quote string associated with the CLASS modifier contains a user specified ID for the infiltration class. The data for this class will be referenced by the ID. The quote string associated with the UNITS modifier must contain the data units. Typically, units of mm/yr are used. The quote string identified as quote3 contains a descriptive label for this infiltration class and must be the third quote string in the

keyword entry. The remaining entries for this syntax are used to define a statistical distribution for the infiltration rate. Additional information about these entries is provided in Section 1.4. Some example uses of this syntax are the following:

```
INFILTRATION DEFINE UNIT="mm/yr" CLASS="Hanford" 1 0.1 "Hanford Barrier"
INFILTRATION DEFINE UNIT="mm/yr" CLASS="Hs-dn" 1 55
  "Hanford Sand (Hs), disturbed (d) - with no (n) vegetation"
INFILTRATION DEFINE UNIT="mm/yr" CLASS="RCRA C" 1 0.1 "Modified RCRA C"
```

The second way to define an infiltration class specifies the rate as a function of two infiltration class rates already defined using the following syntax:

```
INFILTRATION DEFINE [CLASS="quote 1"] [FRACTION=N1]
  [PRIOR="quote2"] [NEXT="quote 3"] "quote 4"
```

The quote string associated with the CLASS modifier contains a user specified ID for the infiltration class. The data for this class will be referenced by the ID. The quote string associated with the PRIOR modifier must contain the ID of an infiltration class already defined. The quote string associated with the NEXT modifier must also contain the ID of an infiltration class already defined. The quote string identified as quote4 contains a descriptive label for this infiltration class and must be the fourth quote string in the keyword entry. The numerical value associated with the FRACTION modifier is used in computing the infiltration rate as a function of the PRIOR and NEXT classes as follows:

$$\text{RATE} = \text{MIN}(\text{PRIOR}, \text{NEXT}) + \text{FRACTION} \times \text{ABS}(\text{NEXT} - \text{PRIOR})$$

This alternative definition allows barrier degradation to be modeled in steps from full barrier effectiveness to a post-degradation long-term rate. For example, if a barrier period uses infiltration class S3 with a rate of 1.5 mm/yr, and the long-term post-barrier infiltration rate uses infiltration class S7 with a rate of 4.0 mm/yr, we can simulate the barrier degradation in four steps as follows:

```
INFILTRATION DEFINE CLASS="S3" UNITS="mm/yr" 1 1.5 "Barrier"
INFILTRATION DEFINE CLASS="S7" UNITS="mm/yr" 1 4.0 "Long-Term"
INFILTRATION DEFINE CLASS="S4" FRACTION=0.25 PRIOR="S3" NEXT="S7"
  "Degrade 1"
INFILTRATION DEFINE CLASS="S5" FRACTION=0.50 PRIOR="S3" NEXT="S7"
  "Degrade 2"
INFILTRATION DEFINE CLASS="S6" FRACTION=0.75 PRIOR="S3" NEXT="S7"
  "Degrade 3"
```

By way of example, the rate for dependent classes S4, S5, and S6 will be computed as follows:

$$\begin{aligned} S4 &= \text{MIN}(S3, S7) + \text{FRACTION} \times \text{ABS}(S7 - S3) = 1.5 + 0.25 \times \text{ABS}(4.0 - 1.5) = 2.125 \\ S5 &= \text{MIN}(S3, S7) + \text{FRACTION} \times \text{ABS}(S7 - S3) = 1.5 + 0.50 \times \text{ABS}(4.0 - 1.5) = 2.750 \\ S6 &= \text{MIN}(S3, S7) + \text{FRACTION} \times \text{ABS}(S7 - S3) = 1.5 + 0.75 \times \text{ABS}(4.0 - 1.5) = 3.375 \end{aligned}$$

The second general syntax applies an infiltration class to a specific vadose zone site for a specific period of time:

```
INFILTRATION [SITE="quote 1"] [CLASS="quote 2"] [START=Year1] [END=Year2]
  {ENHANCE=N1} {" quote 3" }
```



The quote string associated with the modifier SITE must match a site specified by the SITE keyword. The SITE keywords must be placed before the INFILTRATION keywords in the ESD keyword file. The numerical values associated with the START and END modifiers identify the time period (range of integer calendar years) for which this recharge rate will be used. The entire time period specified by the PERIOD keyword (Section 2.1.20) must be covered. The quote string associated with the CLASS modifier must contain the ID of an infiltration class specified using an INFILTRATION DEFINE keyword that precedes the INFILTRATION SITE keyword in the ESD keyword file. The quote string identified as quote3 contains a descriptive label for this infiltration class and must be the third quote string in the keyword entry. The optional ENHANCE modifier indicates the user wishes to apply a recharge rate in vadose zone transport model that differs from that applied in the vadose zone release model by a constant factor, N1. The ENHANCE factor is provided to allow treatment of barrier side-slopes that may serve to enhance recharge for the vadose zone as a whole, but not in the waste disposal area.

For example, one might define an infiltration data set for the “S” Hydrogeologic Province with an operational period cover (unvegetated gravel), which we identify as class “S2”. The INFILTRATION keyword might then be entered as follows:

```
INFILTRATION DEFINE Class="S1" Units="mm/yr" 6 0.2 0.5 0.99 "Example class"
```

In this example, infiltration class S1 is declared and will have infiltration rates sampled from a triangular distribution with minimum 0.2 mm/yr, mode 0.5 mm/yr, and maximum 0.99 mm/yr. One might want to apply the infiltration class just declared to the vadose zone site 241-SX-104 for the period from 1955 through 2040. In this case, the INFILTRATION keyword will be entered as follows:

```
INFILTRATION SITE "241-SX-104" CLASS="S1" FROM 1955 TO 2040
```

### **2.1.14 IOONLY Keyword in the ESD Keyword File**

The IOONLY keyword instructs the main processor ESP to create input files without running any analysis. The following is this 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 2.1.18).

### **2.1.15 IRRIGATE Keyword in the ESD Keyword File**

The IRRIGATE keyword defines the irrigation rates to be used by the SOIL model. The following is this keyword’s syntax:

```
IRRIGATE [SPRING=N1] [FALL=N2] [RATE=N3] [NET=N4] [START=N5]  
[THETAIRG=N6] [THETADRY=N7]
```

The modifiers for this keyword are defined in Table 2.5. The same irrigation period and water application rate applies to all locations in the model domain. No irrigation occurs before the start year. Once irrigation starts, it continues for every subsequent year in the simulation period.

**Table 2.5** Modifiers for the IRRIGATE Keyword in the ESD File

Modifier	Description
FALL	The numerical entry associated with the FALL modifier is last day of the year (in julian days) that irrigation occurs.
NET	The numerical entry associated with the NET modifier is the net fraction of irrigation that percolates below 15 cm in the soil.
RATE	The numerical entry associated with the RATE modifier is the irrigation rate that is applied. This value has units of cm/yr.
SPRING	The numerical entry associated with the SPRING modifier is the day of the year (in julian days) that irrigation begins in the spring.
START	The numerical entry associated with the START modifier is the calendar year that irrigation begins. The assumption is that no irrigation occurs before that date, and that once started, irrigation continues every year after that through the end of the simulation.
THETADRY	The numerical entry associated with the THETADRY modifier is the surface soil volumetric water content during the dry (non-irrigated) period. This value has units of ml/cm <sup>3</sup> .
THETAIRG	The numerical entry associated with the THETAIRG modifier is the surface soil volumetric water content during the irrigation period. This value has units of ml/cm <sup>3</sup> .

The following example begins yearly irrigation in 2004 at a rate of 75 cm per year with 20% of the water infiltrating below 15 cm in soil depth.

```
IRRIGATE SPRING=121 FALL=256 RATE=75.0 NET=0.20 START=2004 THETAIRG=0.5
      THETADRY=0.2
```

### 2.1.16 KDSOIL Keyword in the ESD Keyword File

The KDSOIL keyword is used to enter statistical distribution definitions for solid-aqueous distribution coefficients ( $K_d$ ) to be used for calculating soil and sediment concentrations from water concentrations. The following is this keyword's syntax:

```
KDSOIL [ID="quote1"] [Dist_Index Parameters] {TRUNCATE U1 U2}
      {LABEL="quote2"} [UNITS="quote3"]
```

The quote string associated with the ID modifier is a unique character string of up to 20 characters that will be used to identify this stochastic variable for subsequent uses. It is case sensitive and embedded spaces are significant. The quote string associated with the optional modifier LABEL contains a description for the stochastic variable that can be up to 64 characters long. An entry for quote2 is not

required, although it is used for labeling purposes if present. If the modifier LABEL is present the associated quote string must be entered as well. The quote string associated with the UNITS modifier contains a units descriptor for the data. The strings “none” or “unitless” should be used if the variable is unitless. Further information about defining statistical distributions is provided in Section 1.4.

More than one KDSOIL keyword may be entered. The set of KDSOIL keywords is used to generate a library of stochastic values that can be accessed by any SAC code. The name of the library file is defined using the KDSOIL modifier on the FILE keyword (Section 2.1.11).

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

```
KDSOIL ID="ID#1" 6 0.2 0.5 0.99 LABEL="Example Kd" UNITS= "L/g"
```

The data entered by this keyword are used in the river model MASS2, the riparian zone model RIPSAC, and the soil concentration model SOIL.

### 2.1.17 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 this keyword’s syntax:

```
LOCATION [ID="quote1"] [EASTING=N1] [NORTHING=N2] {AREA=N3}
  {COMPUTE} {POP=N4} {GWAT} {SWAT} {AIRC} {AIRD} {PWAT} {SEDI} {SORP}
  {SEEP} {SODR} {SOGW} {SOSW} [NAME="quote2"] [TYPE="quote3"]
  [LOCUS="quote4"] {APSD=N5} {POROSITY= N6} {FOC=N7} {VEGCOV=N8}
  {NECF=N9} {RHOS=N10} {SRH=N11} {TEMP=N12} {MSWIND=N13} {MZWIND=N14}
  {COXYGEN=N15} {MILE=N16} {IRG_SWAT="quote5"}
```

The modifiers and associated data for this keyword are described in Table 2.6.

**Table 2.6** Modifiers for the LOCATION Keyword in the ESD File

Modifier	Description
AIRC	Presence of this optional modifier indicates that atmospheric concentrations will be stored in the ECDA files for this location.
AIRD	Presence of this optional modifier indicates that atmospheric deposition rates will be stored in the ECDA files for this location.
APSD	The numerical value associated with the APSD modifier is the aggregate particle size. This value has units of mm and is used only in the ECEM code. It is valid only for land locations.
AREA	The numerical value associated with the AREA modifier identifies the land surface area associated with the location. This value has units of m <sup>2</sup> .
COXYGEN	The numerical value associated with the COXYGEN modifier is the concentration of oxygen in surface water. This value has units of g/L and is used only in the ECEM code. It is valid only for aquatic locations.

<b>Modifier</b>	<b>Description</b>
EASTING	The numerical value associated with the EASTING modifier is the easting coordinate for this location. These coordinates are defined in terms of the Lambert projection of the Washington State Plane North American Datum of 1983, expressed in meters.
FOC	The numerical value associated with the FOC modifier is the fraction of organic carbon content in the soil. This value is unitless and is used only in the ECEM code. It is valid only for land locations.
GWAT	Presence of this optional modifier indicates that groundwater concentrations will be stored in the ECDA files for this location. This option can only be used for upland or aquatic locations.
ID	The quote string associated with the ID modifier contains the ID for this location. This string is limited to 15 characters and must be unique. It is used to associate other data with a specific location.
IRG_SWAT	The quote string associated with the modifier indicates the source of surface water for use in computing soil concentrations. The quote string must contain the ID of a location where a surface water solution is computed.
LOCUS	The quote string associated with the LOCUS modifier is used to limit the home range calculations for a mobile species to one side of the Columbia River. This modifier is used by the MOBILEHOME code. It has two valid values, "HANFORD" and "FARSHORE".
MILE	The numerical value associated with the MILE modifier is the river mile for the location. This value has units of miles and is used only for riparian or aquatic locations.
MSWIND	The numerical value associated with the MSWIND modifier is the mean annual wind speed. This value has units of m/sec and is used only in the ECEM code. It is valid only for land locations.
MZWIND	The numerical value associated with the MZWIND modifier is the mixing zone wind speed. This value has units of m/sec and is used only in the ECEM code. It is valid only for land locations.
NAME	The quote string associated with the NAME modifier is a name or description up to 72 characters in length for this location. This value is used for labeling purposes only.
NECF	The numerical value associated with the NECF modifier is the non-erodible elements correlation factor. This value is unitless and is used only in the ECEM code. It is valid only for land locations.
NORTHING	The numerical value associated with the NORTHING modifier is the northing coordinate for this location. These coordinates are defined in terms of the Lambert projection of the Washington State Plane North American Datum of 1983, expressed in meters.

<b>Modifier</b>	<b>Description</b>
POROSITY	The numerical value associated with the POROSITY modifier is the soil porosity. This value is unitless and is used only in the ECEM code. It is valid only for land locations.
POP	The numerical values associated with the POP modifier is the human population at this location. This value is used only in the HUMAN code for calculating population dose.
PWAT	Presence of this optional modifier indicates that river bottom pore water concentrations will be stored in the ECDA files for this location. This option can only be used for aquatic locations.
RHOS	The numerical value associated with the RHOS modifier is the soil density. This value has units of gm/cm <sup>3</sup> and is used in the ECEM and SOIL codes. It is valid only for land locations.
SEDI	Presence of this optional modifier indicates that sediment concentrations (on the river bottom) will be stored in the ECDA files for this location. This option can only be used for aquatic locations.
SEEP	Presence of this optional modifier indicates that seep water concentrations (on the land surface) will be stored in the ECDA files for this location. This option can only be used for upland or aquatic locations.
SODR	Presence of this optional modifier indicates that soil concentrations without irrigation will be stored in the ECDA files for this location. This option can only be used for upland locations.
SOGW	Presence of this optional modifier indicates that soil concentrations calculated by using groundwater for irrigation will be stored in the ECDA files for this location. This option can only be used for upland locations.
SORP	Presence of this optional modifier indicates that riparian zone soil concentrations (on the land surface) will be stored in the ECDA files for this location. This option can only be used for riparian locations.
SOSW	Presence of this optional modifier indicates that soil concentrations calculated by using surface water for irrigation will be stored in the ECDA files for this location. This option can only be used for upland locations.
SRH	The numerical value associated with the SRH modifier is the surface roughness height. This value has units of m and is used only in the ECEM code. It is valid only for land locations.
SWAT	Presence of this optional modifier indicates that surface water concentrations will be stored in the ECDA files for this location. This option can only be used for aquatic locations.
TEMP	The numerical value associated with the TEMP modifier is the temperature. This value has units of Kelvin and is used only in the ECEM model. It is valid only for land locations.

*User Instructions for the Systems Assessment Capability, Rev. 0, Computer Codes  
Volume 1: Inventory, Release, and Transport Modules*

Modifier	Description
TYPE	The quote string associated with the TYPE modifier is the location type. The following are the valid entries for this string: UPLAND, RIPARIAN, or AQUATIC.
VEGCOV	The numerical value associated with the VEGCOV modifier is the fraction of vegetation cover at the location. This value is unitless and is used only in the ECEM model. It is valid only for land locations.

The following example keywords define three upland locations:

```
LOCATION ID="UH0001" NAME="UnsuitableForAgricul" EASTING=576022
NORTHING=154367 GWAT SOGW SOSW SODR AIRC AIRD, TYPE = "UPLAND",
LOCUS = "HANFORD" APSD = 4.00E-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 IRG_SWAT ="QHP025" AREA = 1457376.1
```

```
LOCATION ID="UH0042" NAME="AgExclusionBuffer" EASTING=573022
NORTHING=149867 GWAT SOGW SOSW SODR AIRC AIRD, TYPE = "UPLAND",
LOCUS = "HANFORD" APSD = 4.00E-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 IRG_SWAT ="QHP025" AREA = 562500.0
```

```
LOCATION ID="UH0015" NAME="Upland" EASTING=576022 NORTHING=152117
GWAT SOGW SOSW SODR AIRC AIRD, TYPE = "UPLAND", LOCUS = "HANFORD"
APSD = 4.00E-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 IRG_SWAT ="QHP025" AREA = 562500.0
```

The following example keywords define three aquatic locations:

```
LOCATION ID="Q02I01" NAME="ISLAND 02,BLM" EASTING=574295.6
NORTHING=155058.0 SWAT SEDI PWAT TYPE="AQUATIC" LOCUS="HANFORD"
MILE=375.49 COXYGEN = 1.10E-02 AREA = 2.5E+03
```

```
LOCATION ID="Q06I04" NAME="ISLAND,06,LOCKE ISLAND,UPLANDS-DOE,SHORE-BLM"
EASTING=577962.1 NORTHING=154231.9 SWAT SEDI PWAT TYPE="AQUATIC"
LOCUS="HANFORD" MILE=373.09 COXYGEN = 1.10E-02 AREA = 2.5E+03
```

```
LOCATION ID="QTV008" NAME="Transect VERNITA-1 HRM 0.3" EASTING=559027
NORTHING=146042 SWAT SEDI PWAT TYPE="AQUATIC" LOCUS="FARSIDE"
MILE=387.83 COXYGEN = 1.10E-02 AREA = 2.5E+03
```

The following example keywords define three riparian locations:

```
LOCATION ID="RFP021" NAME="RiparianGrantUpriver100BC" EASTING=564431.8
NORTHING=145847.4 GWAT SEEP SORP AIRC AIRD TYPE="RIPARIAN"
LOCUS="FARSIDE" MILE=384.40 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    = 9360
```

```
LOCATION ID="RFP022" NAME="RiparianGrant100BC" EASTING=564841.7
NORTHING=145818.1 GWAT SEEP SORP AIRC AIRD TYPE="RIPARIAN"
LOCUS="FAR SIDE" MILE=384.10 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    = 9360
```

```
LOCATION ID="RFP031" NAME="RiparianGrantCoyoteRapids" EASTING=568198.5
NORTHING=147140.3 GWAT SEEP SORP AIRC AIRD TYPE="RIPARIAN"
LOCUS="FAR SIDE" MILE=381.70 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    = 9360
```

### 2.1.18 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 this keyword's syntax:

```
MODULE [ID="quote"] {FLOW|TRAN} {HIRES} {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 modifiers for this keyword are defined in Table 2.7.

**Table 2.7** MODULE Options in the ESD File

Quote String	Description
AIRDROP	The quote string "AIRDROP" identifies the program that processes INVENTORY, STOMP, and RATCHET output and inserts air concentrations and deposition rates into the ECDA files.
CFEST	The quote string "CFEST" identifies the groundwater transport module. The modifier HIRES indicates that the high resolution groundwater transport model is to be run. Groundwater flow runs are performed outside the control of ESP.
CRDROP	The quote string "CRDROP" identifies the program that processes MASS2 outputs and inserts surface water, groundwater and sediment concentrations into the ECDA files.
ECDA	The quote string "ECDA" identifies the module that sets up the environmental concentration data files for later use by other modules.
GWDROP	The quote string "GWDROP" identifies the utility program to convert VADER and CFEST outputs into inputs for the river module. It also processes CFEST outputs to insert groundwater concentrations into the ECDA files.
INVENTORY	The quote string "INVENTORY" identifies the inventory module.
MASS2	The quote string "MASS2" identifies the river flow and transport module.

Quote String	Description
RATCHET	The quote string "RATCHET" identifies the atmospheric transport module.
RIPSAC	The quote string "RIPSAC" identifies the riparian zone concentration module. It computes riparian zone seep water, and wetted soil concentrations and inserts them into the ECDA files.
SOIL	The quote string "SOIL" identifies the upland soil concentration module. It computes soil concentrations under three irrigation assumptions (no irrigation, irrigation using groundwater, and irrigation using surface water) and inserts them into the ECDA files.
STOMP	The quote string "STOMP", when accompanied by the modifier FLOW, identifies a flow-only run for the vadose zone module. The optional modifier HIRES indicates that the high resolution groundwater model is to be run.  The quote string "STOMP", when accompanied by the modifier TRAN, identifies a transport-only run for the vadose zone module. The optional modifier HIRES indicates that the high resolution groundwater model is to be run.
VADER	The quote string "VADER" identifies the vadose zone release module.

The following set of keywords executes all available modules:

```

MODULE ID="INVENTORY"      compute
MODULE ID="VADER"          compute
MODULE ID="STOMP" FLOW HIRES compute
MODULE ID="STOMP" TRAN HIRES compute
MODULE ID="CFEST" HIRES    compute
MODULE ID="GWDROP"         compute
MODULE ID="MASS2"          compute
MODULE ID="RIPSAC"         compute
MODULE ID="ECDA"           compute
MODULE ID="SOIL"           compute
MODULE ID="CRDROP"         compute
MODULE ID="RATCHET"        compute
MODULE ID="AIRDROP"        compute

```

### 2.1.19 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 this 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"
```



### **2.1.20 PERIOD Keyword in the ESD Keyword File**

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

```
PERIOD [START=year1] [STOP=year2] [CLOSURE=year3]
```

The modifier START and the associated value year<sub>1</sub> identify the calendar year the simulation starts. The start of the simulation period must be 1944 or later or the inventory code will error terminate. The modifier STOP and the associated value year<sub>2</sub> identify the calendar year the simulation ends. 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 associated value year<sub>3</sub> identify the year that site closure occurs (and all inventory actions have been completed). 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
```

### **2.1.21 PROCESSOR Keyword in the ESD Keyword File**

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

```
PROCESSOR [MACHINE="quote 1"] [RUNDIR="quote 2"] {TYPE="quote 3"}  
{PRIORITY=N1} {COMPUTE}
```

The main processor 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 ESP is being run. The COMPUTE modifier must be present for the PROCESSOR keyword to have any effect.

The quote string associated with the MACHINE modifier must be a unique character string of up to 20 characters that will be used to identify the computational node. The names of the computational nodes are the names by which the computers are identified on the network. Dual-CPU computers should be identified with two identical PROCESSOR keywords in order to access both CPU's.

The quote string associated with the RUNDIR modifier identifies the base run directory for this particular analysis. The entry should include a complete directory path and can be up to 200 character in length.

The quote string associated with the optional TYPE modifier identifies the processor type for the remote machine. The entry should be "P3" for a Pentium 3 CPU and "P4" for a Pentium 4 CPU. This modifier allows ESP to select executable codes that are optimized for different types of processors. The type defaults to "P3" if this entry is not provided.

The numerical entry associated with the optional PRIORITY modifier identifies a priority for the ESP to use in assigning processors. Allowable values are 1, 2, and 3. When assigning jobs, the ESP will check all priority 1 CPU's for availability before checking priority 2 CPU's. Similarly, priority 3 CPU's will not have jobs assigned unless all priority 1 and 2 CPU's are busy. The priority defaults to 1 if this entry is not entered.

The SAC project runs a Linux cluster containing 128 Intel Pentium III 1.0-GHz processors, 2 Intel Pentium IV 2.0-GHz Processors and 13 Intel Pentium IV 2.7-GHz Processors. Each machine is a dual processor, so each machine can be declared twice. The following set of keywords defines a suite of processors to use for a simulation:

```
PROCESSOR MACHINE="c1-0" PRIORITY=1 TYPE="P4" compute
RUNDIR="/home/ANALYSIS4/CA2_stoch_1ky"
PROCESSOR MACHINE="c1-0" PRIORITY=1 TYPE="P4" compute
RUNDIR="/home/ANALYSIS4/CA2_stoch_1ky"
PROCESSOR MACHINE="c1-1" PRIORITY=1 TYPE="P4" compute
RUNDIR="/home/ANALYSIS4/CA2_stoch_1ky"
PROCESSOR MACHINE="c2-0" PRIORITY=2 TYPE="P4" compute
RUNDIR="/home/ANALYSIS4/CA2_stoch_1ky"
PROCESSOR MACHINE="c2-1" PRIORITY=2 TYPE="P4" compute
RUNDIR="/home/ANALYSIS4/CA2_stoch_1ky"
PROCESSOR MACHINE="c0-0" PRIORITY=3
RUNDIR="/home/ANALYSIS4/CA2_stoch_1ky" compute
PROCESSOR MACHINE="c0-1" PRIORITY=3
RUNDIR="/home/ANALYSIS4/CA2_stoch_1ky" compute
```

### **2.1.22 REALIZAT Keyword in the ESD Keyword File**

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

```
REALIZAT N1
```

The number of realizations is given by the single numerical entry N<sub>1</sub>. 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 REALIZAT keyword requests 25 realizations:

```
REALIZAT 25
```

### **2.1.23 REMEDIAT Keyword in the ESD File**

The REMEDIAT keyword identifies remediation actions between waste sites. These actions are computed by VADER and STOMP so both codes use this keyword. The following is this keyword's syntax:

```
REMIAT [YEAR=N1] [FROM="quote1"] [TO="quote2"] {UPPER=N2} {LOWER=N3}
{SOIL=N4} {CAKE=N5} {RIVER=N6} {CEMENT=N7} {CORE=N8} {LIQUID=N9}
{DEBRIS=N10} {REACTOR=N11} {FORM="quote3"}
```

All information needed by STOMP or VADER exists on one keyword although each code may use different data from the keyword. The VADER code remediates waste in the VADER working site inventory and sends it to another site for use in STOMP. The STOMP code remediates soil from the vadose zone and sends it to another site for later use in VADER. This creates the possibility of transfers to sites of quantities from previous runs of both VADER and STOMP. Table 2.8 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 2.8** Modifiers for the REMEDIAT Keyword in the ESD File

<b>Modifier</b>	<b>Description</b>
CAKE	The numerical value associated with the CAKE modifier identifies the fraction of saltcake/sludge waste (nominally found only in waste tanks) 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.
CEMENT	The numerical value associated with the CEMENT modifier identifies the fraction of cement waste to be exported by VADER. Valid values are in the range 0 to 1. The CMNT modifier can be used in place of the CEMENT 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.
DEBRIS	The numerical value associated with the DEBRIS modifier identifies the fraction of debris to be exported by VADER. Valid values are in the range 0 to 1.
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.
LOWER	The numerical value associated with the LOWER modifier identifies the bottom of the column of soil to be excavated from the FROM site by STOMP and moved to the TO site. This value has units of m. STOMP uses a convention that the soil surface is location 0 and locations below the surface are larger than 0. This value must be larger than the value associated with the UPPER modifier.
FORM	(optional; default is SOIL DEBRIS if not specified) the waste form the remediated mass will be assigned by VADER at the TO site receiving the remediated mass.
FROM	The quote string associated with the FROM modifier identifies the release site from which waste or soil will be exported.
REACTOR	The numerical value associated with the REACTOR modifier identifies the fraction of reactor/component waste to be exported by VADER. Valid values are in the range 0 to 1. The CORE modifier can be used in place of the REACTOR modifier.
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.
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.

<b>Modifier</b>	<b>Description</b>
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.
UPPER	The numerical value associated with the UPPER modifier identifies the top of the column of soil to be excavated from the FROM site by STOMP and moved to the TO site. This value has units of m. STOMP uses a convention that the soil surface is location 0 and locations below the surface are larger than 0. This value must be larger than the value associated with the UPPER modifier.
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.

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:

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

The following example keywords cause STOMP to excavate soil from 4 different sites send all the to site 600-148:

```
REMIATION YEAR=1991 FROM="618-9" TO="600-148" UPPER=0.0 LOWER=4.5
SOIL=1.0 DEBRIS=1.0 CAKE=1.0 SLUDGE=1.0 GLASS=1.0 CEMENT=1.0 compute
REMIATION YEAR=1995 FROM="4843" TO="600-148" UPPER=0.0 LOWER=1 SOIL=1.0
DEBRIS=1.0 CAKE=1.0 SLUDGE=1.0 GLASS=1.0 CEMENT=1.0 compute
REMIATION YEAR=1997 FROM="116-B-4" TO="600-148" UPPER=0.0 LOWER=4.5
SOIL=1.0 DEBRIS=1.0 CAKE=1.0 SLUDGE=1.0 GLASS=1.0 CEMENT=1.0 compute
REMIATION YEAR=1997 FROM="116-B-5" TO="600-148" UPPER=0.0 LOWER=4.5
SOIL=1.0 DEBRIS=1.0 CAKE=1.0 SLUDGE=1.0 GLASS=1.0 CEMENT=1.0 compute
```

### **2.1.24 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 ESP. The following is this 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 ESP program will error terminate. Programs other than ESP do not use the REPORT keyword in the ESD keyword file. The name of the report file is entered in a quote string. File names up to 200 characters long are supported, and path information can be included. The following is an example REPORT keyword:

```
REPORT "/home/ANALYSIS/Initial2/ESD_Initial2.rpt"
```

### **2.1.25 RESITE Keyword in the ESD Keyword File**

The RESITE keyword defines a subset of the sites defined by the SITE keyword (see Section 2.1.27) to be processed under the control of ESP. The following is this 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 SITE keyword are the source of data for entry in the quote strings "site<sub>1</sub>", "site<sub>2</sub>", etc. The following RESITE keyword identifies four sites to be calculated in the run of ESP:

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

### **2.1.26 RESTART Keyword in the ESD Keyword File**

The RESTART keyword defines a subset of the realizations defined by the REALIZATION keyword (see Section 2.1.22) to be processed under the control of ESP. The following is this 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.

### **2.1.27 SITE Keyword in the ESD Keyword File**

The SITE keyword identifies the disposal sites where disposal actions will be modeled in SAC. The following is this keyword's syntax:

```
SITE [ID="quote1"] {COMPUTE} [TITLE= "quote2"] {OFFSITE}  
[POINTS=N1 (N2,N3) ... (N4,N5)] {AREAX=N6}  
{AREAXMIN=N7} {UPSCALE=N8} {STACK=N9}
```

The modifiers and associated data for the SITE keyword are described in Table 2.9.

**Table 2.9** Modifiers for the SITE Keyword in the ESD File

<b>Modifier</b>	<b>Description</b>
AREAX	The numerical values associated with the AREAX modifier is the multiplier (unitless) 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.
AREAXMIN	The numerical value associated with the AREAXMIN modifier is the minimum multiplier (unitless) allowed when the $K_s$ -dependent vadose zone modeling area method is used. This value is used only when the numerical value associated with the AREAX modifier is negative.
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 quote string associated with the ID modifier contains a location ID. This string is limited to 30 characters and must be unique. This ID is used to associate other data with a specific waste site. The ESP, inventory code and the pre- and post-processors for the inventory code can handle site IDs up to 30 characters in length. However, some of the other codes can only handle waste site IDs up to 15 characters in length. The net result is that site IDs for sites with the OFFSITE modifier can contain up to 30 characters while site IDs for sites within the modeling domain are limited to 15 characters.
OFFSITE	If the optional OFFSITE modifier is present then the site denotes a location outside the modeling domain. These sites are useful to account for waste that is exported from the modeling domain. Only the site ID and TITLE need to be defined if the OFFSITE modifier is present.
POINTS	The first numerical value (N1) associated with the POINTS modifier is the number of coordinate pairs needed to define the corners or ends of the waste disposal site. This value is followed by N1 coordinate pairs (Easting, Northing) defining the site vertices. Easting and northing coordinates are defined in terms of the Lambert projection of the Washington State Plane North American Datum of 1983, expressed in meters. This modifier is not used when the OFFSITE modifier is used.
STACK	The optional STACK modifier identifies that the waste site is a stack for atmospheric calculations. The associated numerical gives the stack height in meters. If stack height is greater than 20 m the site is modeled as a stack release in RATCHET, otherwise, it is modeled as a surface release. If the specified stack height is less than 20 m the STACK modifier has no effect.
TITLE	The quote string associated with the TITLE modifier contains a descriptive title for the waste site. This quote string is limited to 72 characters. It is used only for output labeling purposes.

Modifier	Description
UPSCALE	The numerical value associated with the UPSCALE modifier is a scaling factor (unitless, value of 1 or larger) that allows the ESP to try different input values if STOMP does not converge. This factor acts as a multiplier on the AREAX multiplier and has the effect of increasing the cross-sectional area of the transport column in STOMP in an effort to achieve a stable solution. If not specified, UPSCALE defaults to 1 (no upscaling).

The coordinates define the locations of the corners of the waste site and are usually 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 an area polygon with vertices defined by the coordinates specified.

The result files of INVENTORY (.res files) will include an entry for sites that have both COMPUTE and OFFSITE in the SITE 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  $A_x$  = the multiplier of the facility footprint area (dimensionless)  
 $Q_{\max}$  = the maximum artificial liquid discharge rate (m/s)  
 $K_{s\min}$  = the minimum hydraulic conductivity of all layers for the given site (m/s) and realization  
 $A_0$  = the nominal area (m<sup>2</sup>) defined by the coordinates in the SITE keyword.

AREAX is constrained by ESP to be equal to or greater than AREAXMIN. This may sometimes be smaller than the facility footprint area, but in order to accurately represent certain tank leaks this is appropriate. 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 SITE keyword for specification of an offsite disposition to the Waste Isolation Pilot Plant (WIPP) to be included in the analyses is as follows:

```
SITE ID="WIPP" TITLE="Waste Isolation Pilot Plant" OFFSITE COMPUTE
```

Three example SITE keywords for specification of onsite disposals are the following:

```
SITE ID="100-B-15" POINTS=4 (565437.787,145477.787),
(565482.213,145477.787), (565482.213,145522.213),
(565437.787,145522.213) TITLE="Radioactive Process Sewer"
AreaX=2.0 COMPUTE
SITE ID="200-E-77" POINTS=4 (575273.468,135644.468),
(575274.532,135644.468), (575274.532,135645.532),
(575273.468,135645.532) TITLE="Injection/Reverse Well"
AreaX=10000.0 COMPUTE
SITE ID="241-A-101" POINTS=4 (575312.245,136033.167),
(575332.505,136033.167), (575332.505,136053.427),
(575312.245,136053.427) TITLE="Single-Shell Tank"
AreaX=-1.0 upScale=9.600 AreaXMin=0.01 compute
```

### 2.1.28 SPECIES Keyword in the ESD Keyword File

The SPECIES keyword is used to enter definitions for ecological species to be simulated. The ECER code uses species from this master list and passes computed information on to the HUMAN, TCERM, and MOBILEHOME codes. The following is this keyword's syntax:

```
SPECIES [ID="quote 1"] [TYPE="quote 2"] [NAME="quote 3"]
{Modifier=N1} ... {Modifier=N34} {EMERGENT}
```

A separate SPECIES keyword must be entered for every species being defined. Table 2.10 describes the modifiers associated with the SPECIES keyword.

**Table 2.10** Modifiers for the SPECIES 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.
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.



<b>Modifier</b>	<b>Description</b>
AE	The numerical entry associated with the AE modifier is the assimilation efficiency of the species. This value is unitless. This modifier applies only to terrestrial plants and animals. 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. This modifier applies only to aquatic plants and animals. If it is not present, the value of AWD defaults to zero.
DIFFHT	The numerical entry associated with the DIFFHT modifier is the diffusion height of the species. This value has units of meters. This modifier applies only to terrestrial plants and animals. 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. This modifier applies only to terrestrial plants and animals. If it is not present, the value of ETWATER defaults to zero.
FLIPID	The numerical entry associated with the FLIPID modifier is the fraction of the mass of the species that is lipid. This value has units of g lipid/g wet. This modifier applies only to aquatic plants and animals. If it is not present, the value of FLIPID defaults to zero.
FMR	The numerical entry associated with the FMR modifier is the free metabolic rate of predator species. This value has units of kcal/day. This modifier applies only to terrestrial animals. 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. This modifier applies only to aquatic plants and animals. 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. This modifier applies only to terrestrial plants. 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. This modifier applies only to terrestrial plants. 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. This modifier applies only to terrestrial plants. If it is not present, the value of FPW defaults to zero.

<b>Modifier</b>	<b>Description</b>
FW	The numerical entry associated with the FW modifier is the water weight fraction of plant tissue for the species. This value is unitless. This modifier applies only to terrestrial plants. 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. This modifier applies only to terrestrial plants and animals. 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. This modifier applies only to terrestrial animals. If it is not present, the value of FDW defaults to zero.
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. This modifier applies only to terrestrial plants and animals. 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 <sup>3</sup> /day. This modifier applies only to terrestrial animals. 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. This modifier applies only to aquatic animals. 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. This modifier applies only to terrestrial animals. 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. This modifier applies only to terrestrial animals. 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. This modifier applies only to terrestrial animals. 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. This modifier applies to all ecological species. 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 <sup>3</sup> . This modifier applies only to terrestrial plants. 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 <sup>2</sup> . This modifier applies only to terrestrial animals. If it is not present, the value of SA defaults to zero.

Modifier	Description
SADHER	The numerical entry associated with the SADHER modifier is the skin adherence factor for the species. This value has units of mg/cm <sup>2</sup> . This modifier applies only to terrestrial animals. If it is not present, the value of SADHER defaults to zero.
THETA	The numerical entry associated with the THETA modifier is the area use factor for the species. This value is unitless. This modifier applies only to terrestrial animals. 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. This modifier applies only to terrestrial animals. 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. This modifier applies only to aquatic plants and animals. 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). This modifier applies only to terrestrial plants and animals. If it is not present, the value of WEIGHT defaults to zero.

The following six examples illustrate the use of the SPECIES keyword for a variety of locations and species characteristics:

```

! Aquatic plant
SPECIES ID="QPERIP" TYPE="QP" NAME="periphyton" HABITAT="AQUATIC"
  AWD=10 FLIPID=0.0418 FOC=0.35 RADIUS=1.4 WBMASS=0.000035

! Aquatic animal
SPECIES ID="QCARPS" TYPE="QA" NAME="CARP" HABITAT="AQUATIC"
  CAR=8.0000E-01 AWD=5.0000E+00 FLIPID=7.4000E-02 FOC=4.5000E-01
  BMASS=1.0500E+03 RADIUS=7.0000E+00

! Terrestrial animal - riparian location
SPECIES ID="RACOOT" TYPE="TA" NAME="American coot" HABITAT="RIPARIAN"
  AE=8.1000E-01 DIFFHT=2.0000E-01 ETWATER=2.0000E+01
  FMR=1.6309E+02 FWATER=5.0000E-01 FDW=0.4
  GE=1.9000E+03 INHRATE=6.1593E-01 PCS=2.5000E-01
  PCW=5.0000E-01 PSI=1.0000E+00 RADIUS=5.0000E+00
  SADHER=1.4500E+00 THETA=1.0000E+00 SA=7.8406E+02
  WATERING=4.6102E-02 WEIGHT=6.9200E-01

! Terrestrial animal - upland location
SPECIES ID="UBDGER" TYPE="TA" NAME="American badger" HABITAT="UPLAND"
  AE=8.1000E-01 DIFFHT=2.0000E-02 ETWATER=0.0000E+00
  FMR=8.2073E+02 FWATER=0.0000E+00 FDW=0.4
  GE=1.7000E+03 INHRATE=6.0032E+00 PCS=2.2000E-01
  PCW=0.0000E+00 PSI=1.0000E+00 RADIUS=1.0000E+01
  SADHER=1.4500E+00 THETA=1.0000E+00 SA=4.1972E+03
  WATERING=7.2063E-01 WEIGHT=9.0000E+00

```

```
! Terrestrial plant - riparian location
SPECIES ID="RCTWOD" TYPE="TP" NAME="black cottonwood" HABITAT="RIPARIAN"
  SURFACE=7.8050E+01 AE=3.4000E-01 DIFFHT=5.0000E+00 FPA=5.0000E-01
  FPL=1.0000E-02 FPW=4.0000E-01 FW=5.5000E-01 GE=3.2000E+03
  FWATER=0.0000E+00 RADIUS=3.0000E+01 RHOP=1.0000E+03
  WEIGHT=2.0000E+02 ETWATER=0.0000E+00
```

```
! Terrestrial plant - upland location
SPECIES ID="UFUNGI" TYPE="TP" NAME="fungi" HABITAT="UPLAND"
  SURFACE=1.0000E-03 AE=7.3000E-01 DIFFHT=5.0000E-02 FPA=5.0000E-01
  FPL=1.0000E-02 FPW=4.0000E-01 FW=9.1000E-01 GE=6.3000E+02
  FWATER=0.0000E+00 RADIUS=1.4000E+00 RHOP=1.0000E+03
  WEIGHT=3.0000E-03 ETWATER=0.0000E+00
```

### **2.1.29 TIMES Keyword in the ESD Keyword File**

The TIMES keyword identifies the times at which concentration data will be saved in the ECDA files to support the calculations by the impacts codes. The following is this keyword's syntax:

```
TIMES [T1] {T2} ... {Tn}
```

The numerical entries T<sub>1</sub>, T<sub>2</sub>, ..., T<sub>n</sub> are the times (whole number calendar years) when concentration data are to be saved. The following example TIMES keyword requests concentration data for the three years 2020, 2075, and 3014:

```
TIMES 2020 2075 3014
```

### **2.1.30 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 200 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."
```

### **2.1.31 USER Keyword in the ESD Keyword File**

The USER keyword is used to identify the user of the ESP program. The user name will be written to output files. The following is this keyword's syntax:

```
USER ["quote"]
```

The user name is entered in a quote string. 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"
```

## 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 impact calculations.

### 2.2.1 Format of the ECDA Files

The file structure for the ECDA files has the following characteristics:

- Data storage for different analytes are provided in separate files (9 analytes would result in 9 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).

After the first 12 header lines, each data record in a ECDA file contains the following information: Year, Location ID, Media ID, media data (realizations 1 to the maximum). The relative order in which media data appear in the file for a given time and location is always the following:

- GWAT: concentrations in groundwater ( $\text{Ci}/\text{m}^3$  or  $\text{kg}/\text{m}^3$ )
- SEEP: concentrations in seep water ( $\text{Ci}/\text{m}^3$  or  $\text{kg}/\text{m}^3$ )
- SWAT: concentrations in surface water (river) ( $\text{Ci}/\text{m}^3$  or  $\text{kg}/\text{m}^3$ )
- PWAT: concentrations in river bottom pore water ( $\text{Ci}/\text{m}^3$  or  $\text{kg}/\text{m}^3$ )
- SEDI: concentrations in river bottom sediment ( $\text{Ci}/\text{kg}_{\text{sediment}}$  or  $\text{kg}_{\text{analyte}}/\text{kg}_{\text{sediment}}$ )
- SORP: concentrations in riparian zone soil (land surface) ( $\text{Ci}/\text{kg}_{\text{soil}}$  or  $\text{kg}_{\text{analyte}}/\text{kg}_{\text{soil}}$ )
- SODR: concentrations in upland soil (land surface) with no irrigation ( $\text{Ci}/\text{kg}_{\text{soil}}$  or  $\text{kg}_{\text{analyte}}/\text{kg}_{\text{soil}}$ )
- SOGW: concentrations in upland soil (land surface) with groundwater irrigation ( $\text{Ci}/\text{kg}_{\text{soil}}$  or  $\text{kg}_{\text{analyte}}/\text{kg}_{\text{soil}}$ )
- SOSW: concentrations in upland soil (land surface) with surface water irrigation ( $\text{Ci}/\text{kg}_{\text{soil}}$  or  $\text{kg}_{\text{analyte}}/\text{kg}_{\text{soil}}$ )
- AIRC: concentrations in air ( $\text{Ci}/\text{m}^3$  or  $\text{kg}/\text{m}^3$ )
- AIRD: air deposition rates ( $\text{Ci}/\text{m}^2/\text{yr}$  or  $\text{kg}/\text{m}^2/\text{yr}$ )

Table 2.11 provides an overview of the structure of a ECDA file assuming that 25 realizations were used.

**Table 2.11** Structure of an ECDA File

<b>Analyte ID</b>		
11 lines, each containing the <b>units</b> for one medium as a character string		
Time 1	Location 1	Media L1-1, 25 realizations
		...
Location 2	Media L1-L1MAX, 25 realiz.	
	Media L2-1, 25 realizations	
		...
		Media L2-L2MAX, 25 realiz.

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

### 2.2.2 Format of the Record Map File for ECDA Data

The record map file is an ASCII file that provides indexing information for the storage locations in the binary ECDA files. One record map file is used for all ECDA files for a given simulation problem because the structure of the files are 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.
- **Block Size.** The number of records with data in a time block in the binary data file.

- **Record Length.** Record length for the binary data file.
- **Number of Realizations.** The number of realizations requested for this run.
- **Number of Times.** The number of solution times requested for this run.
- **List of Times.** A list of solution times with one time per line.
- **Number of Locations.** The number of locations requested for this run.
- **List of Locations.** A list of locations IDs, with one ID 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 2.12 provides excerpts from a record map file for a large data set.

**Table 2.12** Excerpts Record Index Map File

```

"2004 Composite Analysis 10,000-year (Median Inputs) Assessment"
"ECDA"
"2.00.A.4"
"16 Oct 2003"
"Eslinger-DWE-WEN"
"20040802132252"
23155,"Records in a time block"
18,"Record length in the ECDA file"
1,"Number of realizations"
274,"Number of times"
1945
1950
<deleted lines>
12000
12050
4351,"Number of locations"
"UH0001"
"UH0002"
<deleted lines>
"QTV010"
"QHRWI1"
11,"Number of media"
"GWAT"
"SEEP"
"SWAT"
"PWAT"
"SEDI"
"SORP"
"SODR"
"SOGW"
"SOSW"
"AIRC"
"AIRD"
"UH0001",13,-1,-1,-1,-1,-1,14,15,16,17,18
"UH0002",19,-1,-1,-1,-1,-1,20,21,22,23,24
<deleted lines>
"QTV010",-1,-1,23162,23163,23164,-1,-1,-1,-1,-1,-1
"QHRWI1",-1,-1,23165,23166,23167,-1,-1,-1,-1,-1,-1

```

### 2.2.3 ECDA Header file for SACView

The SACView header file is an output file containing information used by the interactive SACView program to allow easy extraction of subsets of data stored in the ECDA files. This file is written by the ECDA code. The file contains the following sections of information:

- A header section of basic run information.
- Number of realizations selected.
- Media used.
- Times used.
- Locations used.
- Analytes used and analyte data file names.
- Record map file name.

Excerpts from a SACView header file is given in Table 2.13. The user only specifies the name of this file; users should not modify the file after it is generated by the ECDA code. See the instructions in (Eslinger et al. (2004a)) regarding the use of the SACView program. Because this file typically has at most a few thousand entries, the file size is always smaller than 200 Kb. File names for options not selected are set to the string "null".

**Table 2.13** Example ECDA Header File for use in SACView

```
type: "ECDA"
title: "2004 Composite Analysis 10,000-year (Median Inputs) Assessment"
user: "Eslinger-DWE-WEN"
name: "ECDA"
version: "2.00.A.4"
date: "16 Oct 2003"
id: "20040802132252"
envfile: "ESD_CA1_median.key"
realizations: 1
media: 11
"GWAT","Groundwater concentrations"
"SEEP","Seep water (riparian zone) concentrations"
"SWAT","Surface water (river) concentrations"
"PWAT","Pore water (river bottom) concentrations"
"SEDI","Sediment (river bottom) concentrations"
"SORP","Soil (riparian zone) concentrations"
"SODR","Non-irrigated upland soil concentrations"
"SOGW","Groundwater irrigated upland soil concentrations"
"SOSW","Surface water irrigated upland soil concentrations"
"AIRC","Air concentrations"
"AIRD","Air deposition rates"
times: 274
  1945
  1950

<deleted lines>

12000
```



```
12050
locations: 4351
"UH0001", "UnsuitableForAgricul"
"UH0002", "UnsuitableForAgricul"

<deleted lines>

"QTV010", "Transect VERNITA-1 HRM 0.3"
"QHRWI1", "City of Richland Municipal Water Intake"
analytes: 15
"C14", "NR", "Carbon-14"
"/home/ANALYSIS4/CA1_median/ecda/C14_CA1_median.dat"
"Cs137", "NR", "Cesium-137"
"/home/ANALYSIS4/CA1_median/ecda/Cs137_CA1_median.dat"
"Cl36", "NR", "Chlorine-36"
"/home/ANALYSIS4/CA1_median/ecda/Cl36_CA1_median.dat"
"I129", "NR", "Iodine-129"
"/home/ANALYSIS4/CA1_median/ecda/I129_CA1_median.dat"
"Np237", "NR", "Neptunium-237"
"/home/ANALYSIS4/CA1_median/ecda/Np237_CA1_median.dat"
"Pa231", "NR", "Protactinium-231"
"/home/ANALYSIS4/CA1_median/ecda/Pa231_CA1_median.dat"
"Ra226", "NR", "Radium-226"
"/home/ANALYSIS4/CA1_median/ecda/Ra226_CA1_median.dat"
"Se79", "NR", "Selenium-79"
"/home/ANALYSIS4/CA1_median/ecda/Se79_CA1_median.dat"
"Sr90", "NR", "Strontium-90"
"/home/ANALYSIS4/CA1_median/ecda/Sr90_CA1_median.dat"
"Tc99", "NR", "Technetium-99"
"/home/ANALYSIS4/CA1_median/ecda/Tc99_CA1_median.dat"
"H3", "NR", "Tritium"
"/home/ANALYSIS4/CA1_median/ecda/H3_CA1_median.dat"
"Eu152", "NR", "Europium-152"
"/home/ANALYSIS4/CA1_median/ecda/Eu152_CA1_median.dat"
"U233", "NR", "Uranium-233"
"/home/ANALYSIS4/CA1_median/ecda/U233_CA1_median.dat"
"U235", "NR", "Uranium-235"
"/home/ANALYSIS4/CA1_median/ecda/U235_CA1_median.dat"
"U238", "NR", "Uranium-238"
"/home/ANALYSIS4/CA1_median/ecda/U238_CA1_median.dat"
recordmap: "/home/ANALYSIS4/CA1_median/ecda/ECDA_CA1_median.map"
```

### **2.3 Shared Environmental Stochastic Data**

Some stochastic data are shared among several programs, including that defined by the KDSOIL (Section 2.1.16) and DILUTE (Section 2.1.6) 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 2.1.6) 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 nine 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 – 3 in this example. Each succeeding line contains data for a single DILUTE definition in the form of index, identification string, data units, 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 2.14** Example DILUTE File

"DILUTE"
"2004 Composite Analysis 1,000-year (Stochastic Inputs) Assessment"
"ECDA"
"2.00.A.4"
"16 Oct 2003"
"20040617150724"
"Eslinger-DWE-WEN"
1
3
1, "DF5m", "None", 6.01588E-01, 5.10553E-01, 5.12033E-01

### 2.3.2 Format of the KDSOIL Data File

Entry of one or more KDSOIL keywords in the ESD keyword file (see Section 2.1.16) causes the ECDA code to generate a data file of stochastic KDSOIL values for use in other codes. Table 2.15 provides an example of this file for 10 KDSOIL keywords and three realizations. The file is a text file that starts with 9 header lines. The entry on line 8 is the number of KDSOIL definitions – 10 in this example. The next line contains the number of realizations – 3 in this example. Each succeeding line contains data for a single KDSOIL definition in the form of index, identification string, data units, 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 2.15** Example KDSOIL File

"KDSOIL"
"2004 Composite Analysis 1,000-year (Stochastic) Assessment"
"ECDA"
"2.00.A.4"
"16 Oct 2003"
"20040617150724"
"Eslinger-DWE-WEN"
10
3
1, "KDC", "L/g", 8.51504E-02, 1.70702E-02, 5.09645E-02
2, "KDC1", "L/g", 0.00000E+00, 0.00000E+00, 0.00000E+00
3, "KDCs", "L/g", 6.42750E-01, 3.89112E+00, 2.62510E+00
4, "KDH", "L/g", 0.00000E+00, 0.00000E+00, 0.00000E+00

5, "KDI", "L/g", 1.37567E-05, 7.58985E-05, 6.64624E-05
6, "KDNp", "L/g", 1.39180E-02, 2.26965E-02, 1.23003E-02
7, "KDSe", "L/g", 5.29678E-03, 4.82762E-03, 5.18776E-03
8, "KDSr", "L/g", 2.35305E-02, 2.12109E-02, 2.08005E-02
9, "KDTc", "L/g", 6.34344E-06, 0.00000E+00, 2.57418E-05
10, "KDU", "L/g", 8.38553E-04, 3.77616E-04, 9.35109E-04

### 2.3.3 Format of the INFILT Data File

Entry of one or more INFILTRATION keywords with the DEFINE modifier in the ESD keyword file (see Section 2.1.13) causes the ESP code to generate a data file of stochastic infiltration values for use in other codes. Table 2.15 provides an example of this file for 11 INFILTRATION keywords and three realizations. The file is a text file that starts with 9 header lines. The entry on line 8 is the number of INFILTRATION definitions – 11 in this example. The next line contains the number of realizations – 3 in this example. Each succeeding line contains data for a single INFILTRATION definition in the form of index, identification string, data units, 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 2.16** Example INFILT File

"INFILT"
"2004 Composite Analysis 1,000-year (Stochastic Inputs) Assessment"
"ESP-Unix"
"1.5"
"21 Jul 2004"
"20040809075542"
"Eslinger-DWE-WEN"
11
3
1, "River", "mm/yr", 1.00000E+00, 1.00000E+00, 1.00000E+00
2, "Eb-s", "mm/yr", 9.53286E-01, 2.35171E+00, 2.12358E+00
3, "El-s", "mm/yr", 1.61586E+00, 1.94937E+00, 2.24883E+00
4, "Ba-s", "mm/yr", 3.84200E+00, 3.99799E+00, 3.55278E+00
5, "Rpe-s", "mm/yr", 9.72041E-01, 1.31674E+00, 1.05688E+00
6, "Rp-s", "mm/yr", 6.07450E+00, 2.90303E+00, 3.05800E+00
7, "Eb-ds", "mm/yr", 5.34899E+00, 5.12679E+00, 3.06532E+00
8, "Eb-dg", "mm/yr", 8.38464E+00, 1.12903E+01, 9.14490E+00
9, "Eb-dn", "mm/yr", 1.68621E+01, 1.96773E+01, 1.75756E+01
10, "El-ds", "mm/yr", 6.43085E+00, 4.41790E+00, 3.39158E+00
11, "El-dg", "mm/yr", 8.03304E+00, 1.19927E+01, 8.45288E+00



## **3.0 ESP – Environmental Stochastic Preprocessor**

### **3.1 Code Purpose**

All of the environmental codes shown on the left side of Figure 1.1 are run as deterministic models except INVENTORY, RIPSAC, and SOIL. This means they require a single set of inputs and produce a single set of outputs for every realization. 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, RIPSAC and SOIL 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 7.3). Many of the other codes shown in Figure 1.1 have been developed to run under either a Windows environment or a Unix environment. However, the ESP has been optimized for the Linux operating system under and, thus, was not developed to run under both systems.

The 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 1.4.

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. A directory structure is developed for the assessment to keep all this information in a logical order. 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/inventory
../assessment/mass2/analyte/realization
../assessment/processors
../assessment/ripsac
../assessment/vadose/site/analyte/realization
../assessment/ratchet/realization/analyte
```

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 600-148), two analytes (H3 and C14) 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/C14/1
.../assessment/vadose/216-H-8/C14/2
.../assessment/vadose/600-148/H3/1
.../assessment/vadose/600-148/H3/2
.../assessment/vadose/600-148/C14/1
.../assessment/vadose/600-148/C14/2
```

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 600-148, H3, and the second realization would look like the following:

```
.../assessment/vadose/600-148/H3/002
```

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

ESP will execute INVENTORY only once. The INVENTORY module can also be executed in a standalone mode external to control by ESP. The RIPSAC and SOIL modules only require one execution. They 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 keyword file (see Section 2.1.21).

### **3.2.1 ESP Major Functions**

ESP is the central control processor of the environmental modules. As such, ESP is the main program for scheduling models of inventory, release, and transport in the atmosphere, the 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 `.../assessment/ecda` directory and run the ECDA module to set up the ECDA, KDSOIL, and DILUTE files.
3. Create the directory structure (run ESP with the CREATEDIR keyword enabled, and all of MODULE, ANALYTE, and SITE keywords enabled). These keywords are enabled by entering a COMPUTE modifier on the keyword line.
4. Run the INVENTORY module (external) and ensure that the inventory result files (`inv1.res`, `inv2.res`, ...) are located in the inventory subdirectory (`.../assessment/inventory`).
5. 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/ratchet/stochastic.key` – RATCHET stochastic definitions
  - `.../assessment/ratchet/metdata/metYYYY.dat` – RATCHET metadata definitions
  - `.../assessment/ratchet/metdata/def_mix.file` – RATCHET mixing layer height definitions
  - `.../assessment/ratchet/metdata/z0_10cm.file` – RATCHET surface roughness definitions
  - `.../assessment/ratchet/metdata/ratchet.ctl` – RATCHET control 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
  - `.../assessment/ripsac/soil.key` – SOIL control and stochastic definitions
6. Create the following realization-specific input files (run ESP with CREATEDIR keyword without COMPUTE, IOONLY keyword enabled, and all of the MODULE, ANALYTE, and SITE keywords enabled):
- `.../assessment/vadose/site/analyte/realization/vader.key`
  - `.../assessment/vadose/site/analyte/realization/input-esp`
  - `.../assessment/ratchet/realization/analyte/airdrop.ctl`
  - `.../assessment/ratchet/metYYYY.dat/surf/location/ratchet.ctl`
  - `.../assessment/ratchet/metYYYY.dat/stack/location/ratchet.ctl`
  - `.../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`.
7. 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`.
8. Check the vadose results (run VZGRAB for all realizations).
9. 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).
10. Model the transport through the atmosphere (run ESP, disable RESITE keyword, enable realizations of interest on the RESTART keyword, enable only the RATCHET module of the MODULES keyword, and enable all ANALYTE keywords), creating the following result files:
- `.../assessment/ratchet/metYYYY.dat/surf/location/particle.bin`
  - `.../assessment/ratchet/metYYYY.dat/surf/location/noble.bin`
  - `.../assessment/ratchet/metYYYY.dat/surf/location/iodine.bin`
11. Model the analyte/realization specific transport through the atmosphere (run ESP, disable RESITE keyword, enable realizations of interest on the RESTART keyword, enable only the AIRDROP module of the MODULES keyword, and enable all ANALYTE keywords), and place air concentration and air deposition data in the ECDA files.



12. 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.
13. Check the groundwater results (run GRGRAB for all realizations).
14. 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).
15. 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.
16. Model the river/groundwater interface zone (run ESP, enable realizations of interest in the RESTART keyword, enable only the RIPSAC module of the MODULES keyword) , and place concentrations in the ECDA files.
17. Model the contaminant accumulation in soil (run ESP, enable realizations of interest in the RESTART keyword, enable only the SOIL module of the MODULES keyword), and place concentrations in the ECDA files.

An example of a typical background-only analysis is provided to illustrate the above sequence of instructions. 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 and first realization of the groundwater concentration file. ESP will then create a new file (.tab file) containing zero concentrations. However, the flow results from CFEST for all time steps and realizations will be used in the calculation of the background river transport (specified on the CFDIRECT keyword of gwdrop.key).

The following is a typical background-only analysis:

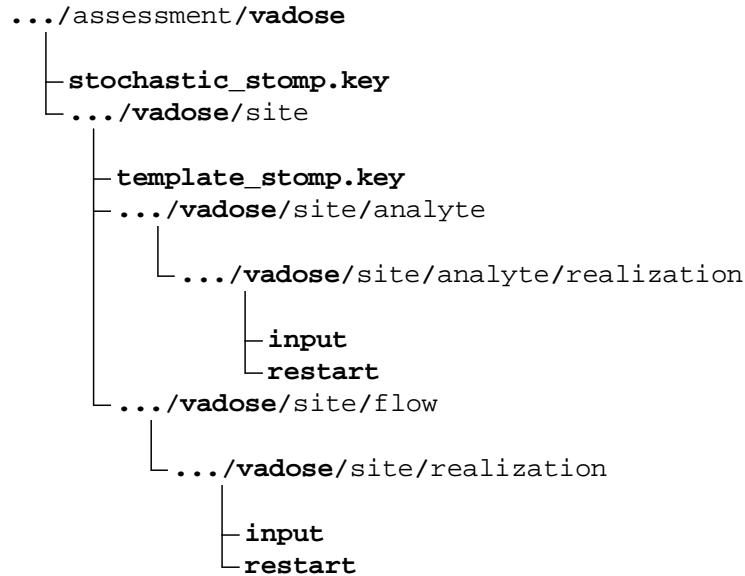
1. Create a new main directory (../assessment/bg).
2. Create the ESD keyword file.

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 existing directories containing previously modeled CFEST result files (i.e., *hheldm001.tab* and *hheldq001.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/hanfnd83m-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/groundwater interface zone (run ESP, enable realizations of interest in the RESTART keyword, enable only the RIPSAC module of the MODULES keyword).
9. Model the soil contaminant accumulation (run ESP, enable realizations of interest in the RESTART keyword, enable only the SOIL module of the MODULES keyword).

### **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 SITE 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.



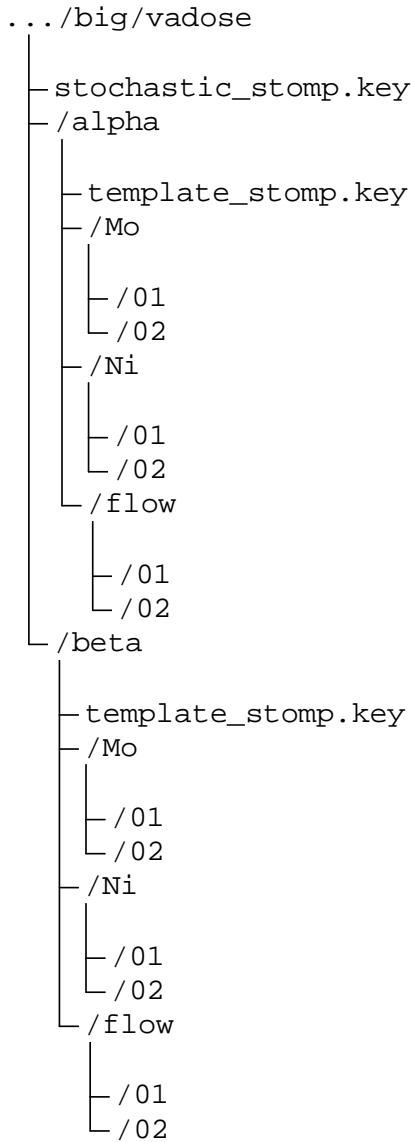
**Figure 3.1** Structure of the /vadose directory

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 SITE keyword in the ESD keyword input file. Similarly, “/analyte” could be /H3, /CCl4, or /Cs137, 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
SITE "alpha" ... compute
SITE "beta" ... compute
```

In this assessment, the /vadose zone subdirectory would appear as shown in Figure 3.2.



**Figure 3.2** Example /vadose zone Subdirectory

This very simple example shows 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 2.1.4). The second pass runs the entire simulation. Alternatively, each process can be run separately using the MODULE keyword of the ESD file (see Section 2.1.18).

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 2.1.21). 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 121<sup>st</sup> 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 ../bin/cfest.* bin/cf
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

```
! SAC Rev.1 2004 Composite Analysis 1,000-year (Stochastic) Assessment
!
!-----| Simulation Control
!
REPORT "ESD_CA2_stoch_lky-CrDir.rpt"
TITLE "2004 Composite Analysis 1,000-year (Stochastic Inputs) Assessment"
USER "Eslinger-DWE-WEN"
!
!- Declare operating system
OS "Unix"
!
!- Declare where executables reside
EXEDIR "/home/CODES/bin"
!
!- Output simulated stochastic values to file
DEBUG STOCHASTIC="stoch/esp_stoch.out"
!
!- Number of realizations to run
REALIZAT 100
!
PERIOD START=1944 STOP=3050 CLOSURE=2070 !
BALANCE 2000 2035 2070 2100 2150 2550 3050
!
!- Create problem subdirectory structure (1st time)
CREATEDIR compute
!
!- Create input files only - don't simulate
!IOONLY compute
!
!- Restart mode only (to preserve 2-pass STOMP setup)
```



```

!RESTART 1 compute
!
!- Resite mode (for individual VZ site handling)
!RESITE "216-B-14" compute

!-----| SAC Module Options |
!BACKGROUND compute
!
!MODULE ID="INVENTORY"      compute
MODULE ID="VADER"           compute
MODULE ID="STOMP" FLOW HIRES compute
MODULE ID="STOMP" TRAN HIRES compute
MODULE ID="CFEST" HIRES     compute
MODULE ID="GWDROP"         compute
MODULE ID="MASS2"          compute
MODULE ID="RIPSAC"         compute
MODULE ID="ECDA"           compute
MODULE ID="SOIL"           compute
MODULE ID="CRDROP"         compute
MODULE ID="RATCHET"        compute
MODULE ID="AIRDROP"        compute

!-----| Distributed Processors
PROCESSOR MACHINE="c0-0"  PRIORITY=3
  RUNDIR="/home/ANALYSIS4/CA2_stoch_lky" compute
PROCESSOR MACHINE="c0-1"  PRIORITY=3
  RUNDIR="/home/ANALYSIS4/CA2_stoch_lky" compute
PROCESSOR MACHINE="c0-62" PRIORITY=3
  RUNDIR="/home/ANALYSIS4/CA2_stoch_lky" compute
PROCESSOR MACHINE="c0-63" PRIORITY=3
  RUNDIR="/home/ANALYSIS4/CA2_stoch_lky" compute

!-----| Analytes
ANALYTE ID="C14"  NAME="Carbon-14"  TYPE="NR"  AIR="NOBLE"      COMPUTE
  MOLWGT=1.400324E+01  HALFLIFE=5.715000E+03  SPECIFIC=4.470906E+00
  DFIMM=5.122392E-05  DFSED=7.200000E-23  GAMMA=0.000000E+00
  MOLDIFF=1.050000E-05  GASDIFF=0.000000E+00  HENRY=0.000000E+00
...
ANALYTE ID="U238"  NAME="Uranium-238"  TYPE="NR"      COMPUTE
  MOLWGT=2.380508E+02  HALFLIFE=4.470000E+09  SPECIFIC=3.362507E-07
  DFIMM=9.276314E-04  DFSED=5.520000E-22  GAMMA=1.000000E-03
  MOLDIFF=1.050000E-05  GASDIFF=0.000000E+00  HENRY=0.000000E+00

!-----| ECDA Solution Times
TIMES
  1945  1950  1955  1960  1965  1970  1975  1980  1985  1990
  2960  2980  3000  3050

!-----| SORP-Water Partition Coefficients (Kd's)
ECHO KDSOIL ! Echo SORP-water Kd values to the ECDA report file
FILE KDSOIL
  NAME="/home/ANALYSIS4/CA2_stoch_lky/ecda/KDSOIL_CA2_stoch_lky.dat"
  SEED=232323.0 CREATE
!   Kd's for bed sediment are from the stomp stochastic file, 4G class
KDSOIL ID="KDC"    2    0.0    0.1    LABEL="Soil-water Kd" UNITS="L/g"
KDSOIL ID="KDU"    9   -7.131  0.596 LABEL="Soil-water Kd" UNITS="L/g"
  TRUNCATE    0.010  0.990
!
!   Kd's for suspended sediment are 10 larger than bed sediment Kd's
KDSOIL ID="KDSUSC" 2    0.0    1.0
  LABEL="Soil-water Kd - suspended sediment" UNITS="L/g"

```

*User Instructions for the Systems Assessment Capability, Rev. 0, Computer Codes*  
*Volume 1: Inventory, Release, and Transport Modules*

---

```
KDSOIL ID="KDSUSU"      9   -4.828    0.596
  LABEL="Soil-water Kd - suspended sediment" UNITS="L/g"
  TRUNCATE    0.010    0.990

!-----| Dilution Factor Data for RIPSAC
ECHO DILUTE ! Echo dilution data to the ECDA report file
FILE DILUTE
  NAME="/home/ANALYSIS4/CA2_stoch_lky/ecda/DILUTE_CA2_stoch_lky.Dat"
  SEED=123445.0 CREATE
DILUTE ID="DF5m" 6   0.36 0.47 0.63
  LABEL="Dilution factor for 5 meters from the river" UNITS="None"
!
!-----| Infiltration Data
FILE INFILT NAME="/home/ANALYSIS4/CA2_stoch_lky/ecda/infilt.dat"
!-----| ECDA Concentration Related Files
! FILLECDA keyword initializes the ECDA files
FILLECDA GWAT FIXED = 0.0
FILLECDA PWAT FIXED = 0.0
FILLECDA SWAT FIXED = 0.0
FILLECDA SEEP FIXED = -1.0
FILLECDA SEDI FIXED = 0.0
FILLECDA SORP FIXED = -1.0
FILLECDA SODR FIXED = -1.0
FILLECDA SOGW FIXED = -1.0
FILLECDA SOSW FIXED = -1.0
FILLECDA AIRC FIXED = 0.0
FILLECDA AIRD FIXED = 0.0
!
! SACVIEW header file for ECDA concentration files
FILE HEADER NAME="/home/ANALYSIS4/CA2_lky/ecda/Sacview_CA2_lky.hdr" CREATE
!
! ECDA record number index map file for concentration data
FILE I_ECDA  NAME="/home/ANALYSIS4/CA2_stoch_/ecda/ECDA_CA2_lky.map" CREATE
!
! ECDA Concentration data files for each analyte
FILE C_ECDA ANALYTE="Cl4"
  NAME="/home/ANALYSIS4/CA2_stoch_lky/ecda/Cl4_CA2_stoch_lky.dat" create
FILE C_ECDA ANALYTE="U238"
  NAME="/home/ANALYSIS4/CA2_stoch_lky/ecda/U238_CA2_stoch_lky.dat" !create

!-----| Irrigation Control
IRRIGATE SPRING=121 FALL=256 RATE=75.0 NET=0.20 START=2004
  THETAIRG=0.5 THETADRY=0.2

!-----| Vadose Zone Release and Transport Sites
SITE ID="100-B-5" POINTS=4 (565432.181,144563.181),
  (565441.819,144563.181), (565441.819,144572.819),
  (565432.181,144572.819) TITLE="Trench"
  AreaX=-1.0 compute
SITE ID="100-F-23" POINTS=4 (580870.5,147790.281), (580871.5,147790.281),
  (580871.5,147791.281), (580870.5,147791.281) TITLE="French Drain"
  AreaX=-1.0 compute
SITE ID="600-148" POINTS=4 (568708.787,134376.6), (569023.337,134376.6),
  (569023.337,134691.15), (568708.787,134691.15)
  TITLE="Landfill (Lined)" AreaX=-1.0 compute
SITE ID="600-211" POINTS=4 (566414.412,138066.896), (566460.838,138066.896),
  (566460.838,138113.322), (566414.412,138113.322)
  TITLE="Drain/Tile Field" AreaX=-1.0 compute
SITE ID="US_Ecology" POINTS=4 (571880.117,134000.117),
  (572519.883,134000.117), (572519.883,134639.883),
  (571880.117,134639.883) TITLE="Burial Ground" AreaX=-1.0 compute
```

*User Instructions for the Systems Assessment Capability, Rev. 0, Computer Codes*  
*Volume 1: Inventory, Release, and Transport Modules*

---

```
!-----| Vadose Zone Infiltration Classes
INFILTRATION SEED=2591.67
!
! declare all unique infiltration classes for vadose zone release sites
INFILTRATION DEFINE UNIT="mm/yr" CLASS="River" 1 1
  "Columbia River outfall locations (placeholder data)"
INFILTRATION DEFINE UNIT="mm/yr" CLASS="Wa-s" 6 0.055 0.11 0.22
  "Warden Silt Loam (Wa) - (Represents final degradation of barrier top)"
!
!- Declare all transitional classes for vadose zone release sites
INFILTRATION DEFINE CLASS="Eb-ds_Eb-s_1" FRACTION=0.8
  PRIOR="River" NEXT="Wa-s" "Barrier Degradation 1"
INFILTRATION DEFINE CLASS="Wa-s_Rp-s_4"
  FRACTION=0.2 PRIOR="Wa-s" NEXT="Eb-ds_Eb-s_1" "Barrier Degradation 4"
!
! assign infiltration classes to all vadose zone release sites
INFILTRATION SITE="100-B-5" CLASS="River" START=1944 END=2000 compute
INFILTRATION SITE="100-B-5" CLASS="Wa-s" START=2001 END=3050 compute
!
!-----| Vadose Zone Site Evaporation
EVAPORATE SITE="216-B-3" START="01/01/1945" END="01/02/1945"
  VOLUME=14881.09 UNITS="m^3" compute
EVAPORATE SITE="216-U-10" START="01/12/1985" END="01/01/1986"
  VOLUME=7559.96 UNITS="m^3" compute
!
!-----| Vadose Zone Site Remediations
REMEDIATION YEAR=1991 FROM="618-9" TO="600-148" UPPER=0.0 LOWER=4.5 SOIL=1.0
  DEBRIS=1.0 CAKE=1.0 SLUDGE=1.0 GLASS=1.0 CEMENT=1.0 compute
REMEDIATION YEAR=1995 FROM="4843" TO="600-148" UPPER=0.0 LOWER=1 SOIL=1.0
  DEBRIS=1.0 CAKE=1.0 SLUDGE=1.0 GLASS=1.0 CEMENT=1.0 compute
REMEDIATION YEAR=1997 FROM="116-B-4" TO="600-148" UPPER=0.0 LOWER=4.5
  SOIL=1.0 DEBRIS=1.0 CAKE=1.0 SLUDGE=1.0 GLASS=1.0 CEMENT=1.0 compute
!
!-----| Impacts Locations
LOCATION ID="UH0001" NAME="UnsuitableForAg" EASTING=576022 NORTHING=154367
  GWAT SOGW SOSW SODR AIRC AIRD, TYPE = "UPLAND", LOCUS = "HANFORD"
  APSD = 4.00E-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
  IRG_SWAT = "QHP025" AREA = 1457376.1
LOCATION ID="RHP234" NAME="RiparianBenton1100Area"
  EASTING=595288.4 NORTHING=111893.7
  GWAT SEEP SORP AIRC AIRD
  TYPE="RIPARIAN" LOCUS="HANFORD" MILE=341.80
  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 = 9360
LOCATION ID="QFP003" NAME="AquaticGrantUpriverVernitaBridge"
  EASTING=557964.6 NORTHING=145457.0 SWAT SEDI PWAT
  TYPE="AQUATIC" LOCUS="FAR SIDE" MILE=388.60
  COXYGEN = 1.10E-02 AREA = 2.5E+03
!-----| Ecological Models Additional Data
RADIUS ID="Radius" 1.400E+00 2.000E+00 3.000E+00 5.000E+00
  7.000E+00 1.000E+01 2.000E+01 3.000E+01
RADIUS ID="C14" 5.000E-02 5.000E-02 5.000E-02 5.000E-02 5.000E-02
  5.000E-02 5.000E-02 5.000E-02 5.000E-02
RADIUS ID="U238" 4.300E+00 4.300E+00 4.300E+00 4.300E+00
  4.300E+00 4.300E+00 4.300E+00 4.300E+00
```

```
!-----| Biotic Species Information
! Type QP: Aquatic plants
SPECIES ID="QPERIP" TYPE="QP" NAME="periphyton" HABITAT="AQUATIC"
AWD=10 FLIPID=0.0418 FOC=0.35 RADIUS=1.4 WBMASS=0.000035
! Type QA: Aquatic animals
SPECIES ID="QCARPS" TYPE="QA" NAME="CARP " HABITAT="AQUATIC"
OCAR=8.0000E-01 AWD=5.0000E+00 FLIPID=7.4000E-02 FOC=4.5000E-01
WBMASS=1.0500E+03 RADIUS=7.0000E+00
! Type TA: Terrestrial animals - riparian locations
SPECIES ID="RACOOT" TYPE="TA" NAME="American coot" HABITAT="RIPARIAN"
AE=8.1000E-01 DIFFHT=2.0000E-01 ETWATER=2.0000E+01 FMR=1.6309E+02
FWATER=5.0000E-01 FDW=0.4 GE=1.9000E+03 INHRATE=6.1593E-01
PCS=2.5000E-01 PCW=5.0000E-01 PSI=1.0000E+00 RADIUS=5.0000E+00
SADHER=1.4500E+00 THETA=1.0000E+00 SA=7.8406E+02 WATERING=4.6102E-02
WEIGHT=6.9200E-01
! Type TA: Terrestrial animals - upland locations
SPECIES ID="UBDGER" TYPE="TA" NAME="American badger" HABITAT="UPLAND"
AE=8.1000E-01 DIFFHT=2.0000E-02 ETWATER=0.0000E+00 FMR=8.2073E+02
FWATER=0.0000E+00 FDW=0.4 GE=1.7000E+03 INHRATE=6.0032E+00
PCS=2.2000E-01 PCW=0.0000E+00 PSI=1.0000E+00 RADIUS=1.0000E+01
SADHER=1.4500E+00 THETA=1.0000E+00 SA=4.1972E+03 WATERING=7.2063E-01
WEIGHT=9.0000E+00
! Type TP: Terrestrial plants - riparian locations
SPECIES ID="RCTWOD" TYPE="TP" NAME="black cottonwood" HABITAT="RIPARIAN"
SURFACE=7.8050E+01 AE=3.4000E-01 DIFFHT=5.0000E+00 FPA=5.0000E-01
FPL=1.0000E-02 FPW=4.0000E-01 FW=5.5000E-01 GE=3.2000E+03
FWATER=0.0000E+00 RADIUS=3.0000E+01 RHOP=1.0000E+03
WEIGHT=2.0000E+02 ETWATER=0.0000E+00
! Type TP: Terrestrial plants - upland locations
SPECIES ID="UFUNGI" TYPE="TP" NAME="fungi" HABITAT="UPLAND"
SURFACE=1.0000E-03 AE=7.3000E-01 DIFFHT=5.0000E-02 FPA=5.0000E-01
FPL=1.0000E-02 FPW=4.0000E-01 FW=9.1000E-01 GE=6.3000E+02
FWATER=0.0000E+00 RADIUS=1.4000E+00 RHOP=1.0000E+03 WEIGHT=3.0000E-03
ETWATER=0.0000E+00
!-----| End of ESD File
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

The creation of the ECDA files can be controlled by ESP or they can be created outside the main processing stream. The typical method is to create the ECDA files outside SAC-ESO control. The

program ecda-1.exe creates the ECDA files for all analytes (see Section 2.2) and a header file used by the SACView program when viewing the contents of the ECDA files. The ecda-1.exe program relies solely on the ESD keyword file for problem control.

### 3.4.1.2 Input Files for INVENTORY

The inventory module typically will be modeled external to ESP control. Section 1.0 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 3.5 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<sup>3</sup> and 1.3700E+06 Ci/cm<sup>3</sup>, 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<sup>2</sup> 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).
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 in Table 3.9:

**Table 3.9** Example File .../assessment/vadose/stochastic\_stomp.key

```

!=====
! Title and Random Number Generator Seed.
!=====

```

*User Instructions for the Systems Assessment Capability, Rev. 0, Computer Codes*  
*Volume 1: Inventory, Release, and Transport Modules*

```

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

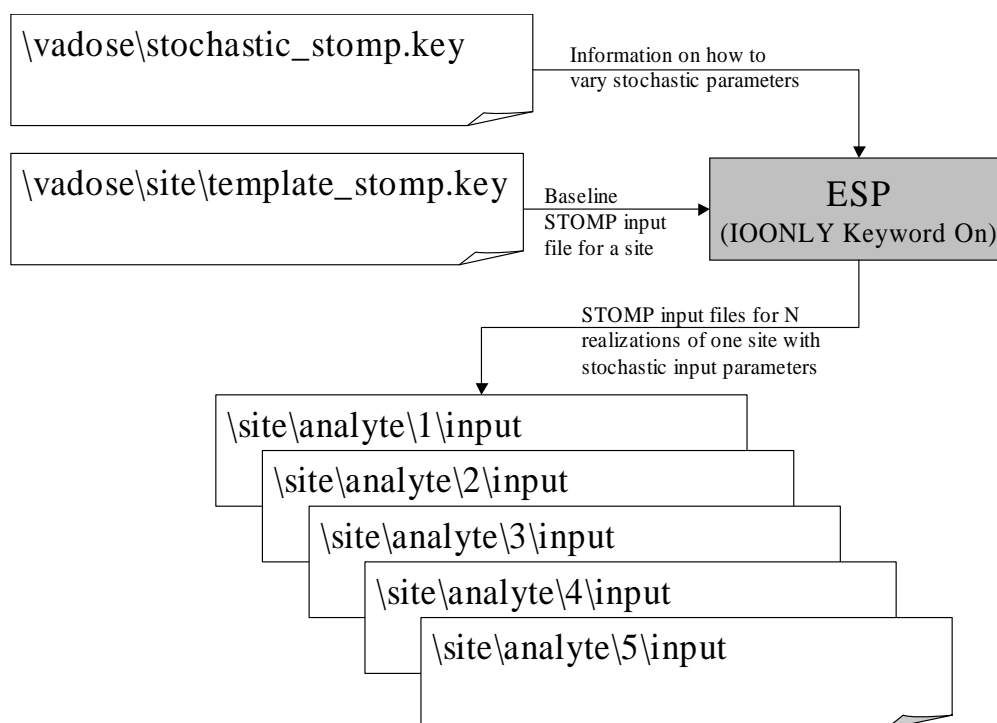
The user who prepares the vadose zone flow and transport inputs for SAC must be a competent user of the STOMP simulator (White and Oostrom 2000). To prepare for vadose zone flow and transport calculations in SAC, the user must take the following steps:

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.



**Figure 3.3** Using the ESP to Generate STOMP Input Files

The ESP is used to combine 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.

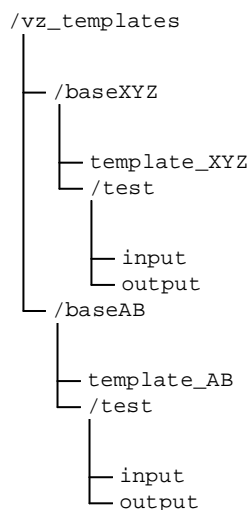
Ideally, a STOMP template is simulated in a standalone fashion. In fact, it is highly recommended that a test liquid volume and dilute species source keyword card be used to test any template before it is actually included in a SAC assessment. Such testing will ensure that the template is fully functional (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 directory structure shown in Figure 3.4:



**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

<pre>'cfest_001', 0,nprint_ctl 'hanford.qa' 'bincf/cfest' input.lpl input.l3i input.hbc concbc.dat MKD 0, 0, 0, 1, 0, 0, 1, 1, 1,0, 0,0, 0, 1, 1, 0, 2, 4, 1.00, 1.00, 1.0,4492.47, 1, 3=cross deriv. to 0 2,0.00100, 1.E-06,1.E-05, 1, 15,0.750000E-01,0.090000,2, 0, 1,0,0.100000E-05,0.100000E-05, 1.00000, cu. meters Kg day 1.00000, sq. meters * * NSPGC ACCELATOR (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,</pre>	<pre>Important this version number should match code version Line 1 - path and name for binary &amp; ascii output files Line 2 - LPROG1 input file name Line 3 - LPROG3I input filename Line 4 - time dependent head B.C. file name Line 5 - time dependent head B.C. file name Line 6 - Model units (valid options are "FPS" or "MKS") Line 7 - print options: Line 8 - NTSUBD,INTERM Time step sub-division Line 9 - ITT (initial or restart time step),ITSTOP(0=ALL) Line 10 - NHEAD_NEW,ILOPT_HEAD,NCYES,NDYES(1=y,0=n) Line 11 - NDIGH,NDIGC ** digits in head &amp; conc output Line 12 - Implicit (=1.0) CRANK-NICOLSON (0.5-1.0) Line 13 - UPSTRM (0.0 to 1.0) Upstream factor Line 14 - RETARD,HALF-LIFE in simulation units Line 15 - NCROSS (0=norm,1=enhanced,2=same as 1 except exit, Line 16 - ITRHO (Max. rho iter.), RHO convergence (0.01-.0001) Line 17 - RPARM1_H, RPARM1_C iterative convergence factors Line 18 - ITRANS (0=confined, 1=unconfined, 2=both) Line 19 - IWT_TIMES,ERR_TRANS,THICK_MIN,IWRITE_DRY Line 20 - IOPT_K (0=normal,1=unsat_k.tab) Line 21 - IOPT_M,IM_ERR,FL_NODAL_ERRMIN, SOL_NODAL_ERRMIN Line 22 - FACT_VOL factor for printing fluid volume Line 23 - FUNITS_VOL text string for fluid volume output Line 24 - Mass units Line 25 - time units(e.g. sec, days) Line 26 - FACT_AREA used for printing only Line 27 - area Units used for printing only ----- * NSPGC ACCELATOR (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) * -----</pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

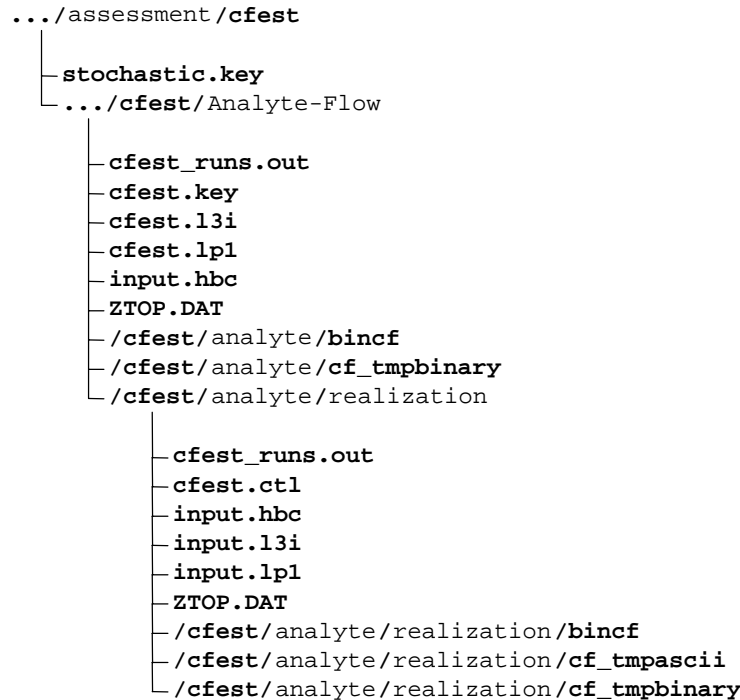
The only variable being simulated for CFEST is the retardation coefficient for each analyte. Table 3.11 provides a sample stochastic definition file.

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

<pre>! 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="CC14"   9  0.9517  0.2609  compute RETARDATION ANALYTE="CrVI"   1  1.0          compute ! END</pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 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**.



**Figure 3.5** Structure of the `/cfest` Subdirectory

In the Figure 3.5 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 `/CCI4` 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-Flow/cfest.key
cp /home/DATA/Database.2/groundwater/Template.Fast/cf_runs.out
/home/ANALYSIS/Median2/cfest/H3-Flow /cf_runs.out
cp /home/DATA/Database.2/groundwater/Template.Fast/cfest.lp1
/home/ANALYSIS/Median2/cfest/H3-Flow /cfest.lp1
cp /home/DATA/Database.2/groundwater/Template.Fast/cfest.l3i
/home/ANALYSIS/Median2/cfest/H3-Flow /cfest.l3i
cp /home/DATA/Database.2/groundwater/Template.Fast/input.hbc
/home/ANALYSIS/Median2/cfest/H3-Flow /input.hbc
cp /home/DATA/Database.2/groundwater/Template.Fast/ZTOP.DAT
/home/ANALYSIS/Median2/cfest/H3-Flow /ZTOP.DAT
cp /home/DATA/Database.2/groundwater/Template.Fast/bincf/*
/home/ANALYSIS/Median2/cfest/H3-Flow /bincf/
cp /home/DATA/Database.2/groundwater/Template.Fast/cf_tmpbinary/*
/home/ANALYSIS/Median2/cfest/H3-Flow /cf_tmpbinary/
#
```

### 3.4.1.12 Typical Steps to Prepare CFEST Input for SAC

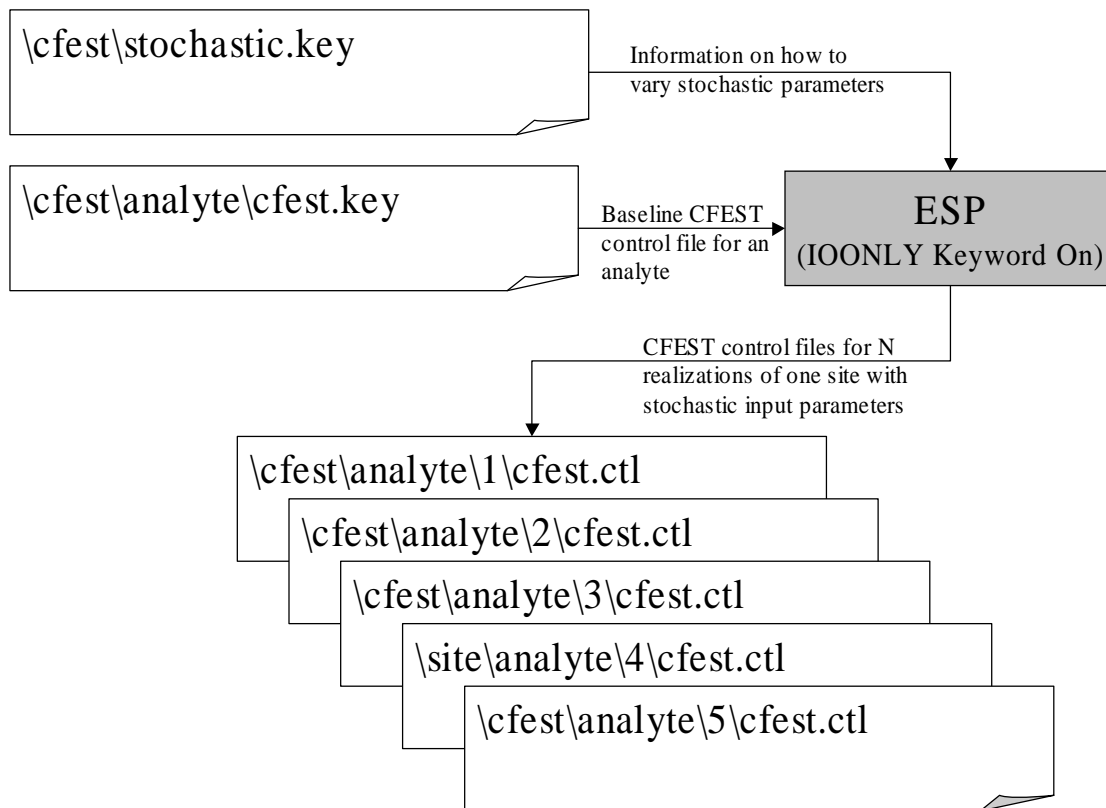
To prepare for groundwater transport calculations in SAC, the user must take the following steps:

- 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).
- 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).

- 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 'cfestctl' 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 ('cfestctl') 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 cfestctl control files for each analyte and realization. Figure 3.6 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.



**Figure 3.6** Using the ESP to Generate CFEST Control Files

Table 3.13 provides recommended parameter values for the cfestctl 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	bincl/cfest	bincl/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
RPARAM1_H	17	1.0E-6	1.0E-6
RPARAM1_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 1.0 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.ct1
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 and Control for Atmospheric Transport Using RATCHET**

Once INVENTORY and STOMP have completed their modeling of the release and transport for all sites, analytes, and realizations, the atmospheric transport module (RATCHET) can be run. RATCHET requires many input files, which need to be placed in the appropriate subdirectories. The user who

prepares the atmospheric inputs for SAC must be a competent user of the RATCHET simulator (RATCHET2). ESP will simulate RATCHET stochastic variables and place them in the appropriate files. Table 3.16 provides a sample RATCHET control file.

**Table 3.16** Sample RATCHET Control File .../assessment/ratchet/metdata/ratchet.ctl

```
RATCHET2 test for Shakedown4 of SAC
01 01 1998
01
01 01 1999
01
8760
46.3333 119.4167
4000.0
N 1000
met.dat
station.dat
z0_10cm.file
def_mix.file
iodine.bin
particle.bin
noble.bin
9.25 20.25 0.0
0.00000
```

The only variable being simulated for RATCHET is the metdata file for each realization. Table 3.17 provides a sample stochastic definition file.

**Table 3.17** Sample Stochastic Definition File for RATCHET .../assessment/ratchet/stochastic.key

```
!=====!
! RATCHET stochastic keyword file (stochastic.key)
! For use in Shakedown2 Assessments
!=====!
TITLE "RATCHET Stochastic Definition Keyword File"
!
SEED 11211.112
!
DEBUG STOCHASTIC="stoch/rat_stoch.out"
!
! define Ratchet grid
GRID EAST=583401.7605 NORTH=111718.4758 ESTEP=2000 NSTEP=2000 NEAST=41 NNORTH=53
!
! define met data partitioning
!METDATA FILE="met1993.dat" STATION="met1993.sta" YEAR=1993 PERCENT=100
METDATA FILE="met1983.dat" STATION="met1983.sta" YEAR=1983 PERCENT=5
METDATA FILE="met1984.dat" STATION="met1984.sta" YEAR=1984 PERCENT=5
METDATA FILE="met1985.dat" STATION="met1985.sta" YEAR=1985 PERCENT=5
METDATA FILE="met1986.dat" STATION="met1986.sta" YEAR=1986 PERCENT=5
METDATA FILE="met1987.dat" STATION="met1987.sta" YEAR=1987 PERCENT=5
METDATA FILE="met1988.dat" STATION="met1988.sta" YEAR=1988 PERCENT=5
METDATA FILE="met1989.dat" STATION="met1989.sta" YEAR=1989 PERCENT=5
METDATA FILE="met1990.dat" STATION="met1990.sta" YEAR=1990 PERCENT=5
METDATA FILE="met1991.dat" STATION="met1991.sta" YEAR=1991 PERCENT=5
METDATA FILE="met1992.dat" STATION="met1992.sta" YEAR=1992 PERCENT=5
METDATA FILE="met1993.dat" STATION="met1993.sta" YEAR=1993 PERCENT=5
METDATA FILE="met1994.dat" STATION="met1994.sta" YEAR=1994 PERCENT=5
METDATA FILE="met1995.dat" STATION="met1995.sta" YEAR=1995 PERCENT=5
METDATA FILE="met1996.dat" STATION="met1996.sta" YEAR=1996 PERCENT=5
METDATA FILE="met1997.dat" STATION="met1997.sta" YEAR=1997 PERCENT=5
METDATA FILE="met1998.dat" STATION="met1998.sta" YEAR=1998 PERCENT=5
METDATA FILE="met1999.dat" STATION="met1999.sta" YEAR=1999 PERCENT=5
METDATA FILE="met2000.dat" STATION="met2000.sta" YEAR=2000 PERCENT=5
```

```
METDATA FILE="met2001.dat" STATION="met2001.sta" YEAR=2001 PERCENT=5
METDATA FILE="met2002.dat" STATION="met2002.sta" YEAR=2002 PERCENT=5
!
END
```

### 3.4.1.16 Input Files and Control of AIRDROP

The AIRDROP code (see Section 1.0) extracts data from files written by RATCHET (atmospheric transport) and writes it to the environmental concentration data files (see Section 2.2).

### 3.4.1.17 Input Files and Control of CRDROP\_INDEX

The CRDROP code (see Section 1.0) 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 3.18) is needed by CRDROP\_INDEX and must be located in the mass2 subdirectory.

**Table 3.18** 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.

### 3.4.1.18 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 3.19 shows the index of the Kd value (as simulated earlier when creating the ECDA files (see Section 2.1.16)) 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 3.19** 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 3.20 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 3.20** 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
!
! Bottom 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
!
! Bioconcentration factors are needed for analyte 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)
```

*User Instructions for the Systems Assessment Capability, Rev. 0, Computer Codes*  
*Volume 1: Inventory, Release, and Transport Modules*

```

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 3.21 provides an example file.

**Table 3.21** 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
```

### 3.4.1.19 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 ECDA files (see Section 2.2.1). CRDROP reads the ESD keyword file and a `crdrop.key` file (created by ESP) to determine where data are located. Section 9.4.1.1 describes the entries for the `crdrop.key` file. Table 3.22 provides an example file.

**Table 3.22** 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
```

### 3.4.1.20 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 3.23. Section 1.0 further describes RIPSAC. RIPSAC can be run once (all analytes and all realizations), or it can be run multiple times where each run processes a subset of analytes and realizations.

**Table 3.23** 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

```

### 3.4.1.21 Inputs and Control of SOIL

The SOIL code is used to model the accumulation of contaminants in soil at upland locations. SOIL reads the ESD keyword file and a control keyword file (.../assessment/soil/soil.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 soil.key file, as illustrated in Table 3.24. Section 1.0 further describes SOIL. SOIL can be run once (all analytes and all realizations), or it can be run multiple times where each run processes a subset of analytes and realizations.

**Table 3.24** Example File soil.key Used in the SOIL Code

```

!=====!
!
!           SAC Rev.1 2004 Composite Analysis 10,000-year (Median Inputs) Assessment
!
!=====!
REPORT "Soil_CA1_median.rpt"
FILE ESD  "/home/ANALYSIS4/CA1_median/ESD_CA1_median.key"
USER  "Paul W. Eslinger"
TITLE  "2004 CA - SOIL keyword file - 10,000-year median inputs case"
REALIZAT ALL
!
EXECUTE
!
! Analytes to process
ANALYTE ID="H3"
ANALYTE ID="C14"

! Debug options
!DEBUG CONCENTR
!DEBUG LOOP

```

```
!DEBUG COMPUTE
!-----| Soil/Water Kd Data |-----!
KDSOIL TERSE ANALYTE="H3"    KDSOIL="KDH"
...
KDSOIL TERSE ANALYTE="C14"    KDSOIL="KDC"
!-----| Infiltration Control |-----!
LOCATION ID="UH0001"  INFILT="Wa-s"
LOCATION ID="UH0002"  INFILT="Ba-s"
...
LOCATION ID="UHR122"  INFILT="Wa-s"
LOCATION ID="UHR123"  INFILT="Wa-s"

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 1.1. 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 RATCHET Code**

RATCHET will create atmospheric transport files for each metadata file. This will be used by AIRDROP to calculate the atmospheric transport for each analyte-realization.

If RATCHET completes its analysis without any problems, it creates a ratchet.done file. However, if RATCHET fails for any reason, RATCHET will create a ratchet.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.

#### **3.4.2.5 Results Used by ESP from the AIRDROP Code**

AIRDROP reads the inventory file, a file created by STOMP (airborne), and the RATCHET result file for a given metadata file (year). AIRDROP will write the time-dependent analyte mass (chemicals) or activity (radionuclides) being released into the atmosphere into the ECDA file, as well as the time-dependent analyte atmospheric deposition rate.

If AIRDROP completes its analysis without any problems, it creates a airdrop.done file. However, if AIRDROP fails for any reason, AIRDROP will create a airdrop.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.

#### **3.4.2.6 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.

### **3.4.2.7 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.

### **3.4.2.8 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.

### **3.4.2.9 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.

### **3.4.2.10 Results Used by ESP from the CRDROP Code**

CRDROP reads the MASS2 result river file and inserts concentration results into the ECDA files. 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.

### **3.4.2.11 Results Used by ESP from the RIPSAC Code**

RIPSAC 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.

#### **3.4.2.12 Results Used by ESP from the SOIL Code**

SOIL is the last module to run under control of the ESP. It runs only after the air, 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 soil subdirectory – which is specified in the soil.key file). When SOIL is finished, it writes a soil.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 INVENTORY code provides a description of the annual movement of contaminants into the environment, thereby becoming available for liquid release or waste form dissolution and transport. The information is provided annually for every contaminant for 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 inventory code does not track contaminants before they become releasable into the environment. For instance, the contents of an intact Hanford waste tank are not tracked in the inventory database or the inventory code. However, past leaks from tanks are tracked at the year of the leak and future residual waste in tanks after cleanout actions have been performed is tracked once the cleanout actions are declared to be complete.

The primary analyses performed by the INVENTORY code 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 ID, 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 2.1.22) in the ESD keyword file.

#### **4.2.1 Radioactive Decay**

Radioactive decay is performed using a general decay algorithm to describe chain decay with branching (Streng 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)  
 $\lambda_c$  = 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:

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

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

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

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

where  $A_c(t)$  = quantity of chain member  $c$  at time  $t$  (atoms)  
 $\lambda_c$  = 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  $\lambda_c$ .

## 4.2.2 Analysis Steps

Preparation of consistent inputs for the INVENTORY code can be complicated because it involves integration of several data sets. The following major steps are used in developing the data for an analysis:

1. Develop the ESD keyword file to set the scope for the entire analysis. This will define the analytes to be used, the waste sites to be modeled, the time period of the analysis and the number of realizations to simulate.
2. Obtain a download from the inventory database (Eslinger et al. (2004b), Section 21) identifying the recorded disposal actions for the desired suite of analytes.
3. Prepare inputs for the INPROC code (Eslinger et al. (2004b), Section 19) and run it using the download from the inventory database. This run will prepare a disposal action file containing

stochastic definitions and WASTEMAP keywords. The INPROC code also implements fill-in rules for missing analyte information and surrogate rules for sites with no recorded information.

4. If desired, convert the stochastic disposal action file to a deterministic disposal action file that replaces all the variables with their median values. This step can be automated using the Inv\_Med codes (Eslinger et al. (2004b), Section 20).
5. Prepare the inventory.key file (including the WASTEMAP keywords generated during the INPROC run) and run the INVENTORY code to build the set of inventory results files for each realization that are used by the rest of the SAC codes.

### **4.3 Code Environment**

This section describes how to run INVENTORY.

#### **4.3.1 Location in Processing Sequence**

The INVENTORY code is executed one time before the execution of all the other codes in the processing sequence (see Figure 1.1). 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**

The INVENTORY code can run either under the Windows operating system or the Linux operating system. Under Windows, INVENTORY executes in a DOS box. To start INVENTORY, enter the following command line:

```
INVENTORY "ESDfilename"
```

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

```
Inventory-1.exe "ESDfilename"
```

Under Linux, file names are case sensitive; under Windows, they are not. For these command lines, INVENTORY.EXE or inventory-1.exe is the name of the executable program and ESDfilename is the name of the ESD keyword file. The ESD keyword file name may contain path information in addition to the file name. Nominally the inventory input files will 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 named "inventory.key", the Master Waste Stream Disposal Action file, and the ESD keyword file. The ESD



keywords are described in Section 2.1. The keywords used in INVENTORY are summarized in Table 4.1. The following sections describe the keywords used in inventory.key and the Master Waste Stream Disposal Action files.

**Table 4.1** Summary of Keywords Used by INVENTORY

<b>Keyword</b>	<b>Information Used by INVENTORY</b>
DEBUG	Selection of options for writing of intermediate stochastic analysis information to an output file.
DECAY	Specification of handling of radioactive decay in the analysis.
DISPOSAL	Specification of the volume and analyte concentrations for each disposal action.
END	Signifies end of a keyword file.
EXECUTE	Causes complete execution of INVENTORY in addition to performing data consistency checks.
FILE	Specification of file names for input and output.
MASSBALANCE	Specifies that a mass balance analysis will be performed.
NORMALIZE	Specification of expected total quantities for analytes.
SEED	Specification of random number seeds for generation of stochastic values.
SELECT	Selection of waste types and waste streams to include.
TIMER	Causes execution time information to be generated and written to the output file.
TITLE	Title of inventory.key keyword input file.
USER	User responsible for inventory.key keyword input file.
WASTEMAP	Specification of mapping of waste streams to aggregation sites.

#### **4.4.1 DEBUG Keyword for INVENTORY**

The optional DEBUG keyword is used to specify generation of summary information related to stochastic variables used in the analysis. The following is this keyword's syntax:

```
DEBUG {DEFINITI} {STOCHAST} {STATISTI}
```

The optional modifier DEFINITI causes the stochastic variable definitions to be written to the stochastic output file. The optional modifier STOCHAST causes the stochastic values generated for use in the analysis to be written to the stochastic output file. The optional modifier STATISTI causes some summary statistics for the generated values for each stochastic variable 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 DEBUG keyword should be used only when a small data set is being used because a large volume of output can be generated.

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

```
DEBUG STOCHAST
DEBUG DEFINITI STATISTI
```

#### **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. The following is this keyword's syntax:

```
DECAY [NONE | N1]
```

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. An example keyword entry for this option is the following:

```
DECAY NONE
```

When the modifier NONE is absent, a numerical value must be provided giving the base year to which all radionuclide activity quantities (e.g., concentrations) are decay corrected. The value must be 1944 or greater or an error message is written and execution is terminated. Decimal years are allowed so 1955.5 would represent July 1, 1955 as follows:

```
DECAY 1955.5
```

#### **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. The following is this keyword's syntax:

```
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}...}
```

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. Each of the modifiers other than the CONTAM<sub>i</sub> modifier must appear only once on the keyword record. At least one CONTAM<sub>i</sub> modifier must be present for each DISPOSAL keyword. The syntax for the stochastic parameter entries are described in Section 1.4.

**Table 4.2** Modifiers for the DISPOSAL Keyword in INVENTORY

Modifier	Description
CONTAMi	This modifier is composed of the letters <i>CONTAM</i> 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. Following the modifier is the ID of the analyte (up to six characters in a quote strong) and numbers defining the concentration of the analyte (in stochastic parameter representation) for the current disposal action.
TYPE	The quote string associated with the TYPE modifier is the waste type (up to six characters in length) for the current disposal action.
VOLUME	This modifier is followed by the volume (in stochastic parameter representation) for the current disposal action.
WASTEID	The quote string associated with the WASTEID modifier is the waste site ID (up 15 characters long) for the current disposal action. If the site has the OFFSITE property on the SITE keyword in the ESD keyword file the site ID may contain 30 characters, otherwise the site ID must be limited to 15 characters.
YEAR	The numerical value associated with the YEAR modifier is the calendar year (integer) that the current disposal action occurs.

Three example DISPOSAL keywords are provided here. The first two example records define analyte concentrations using the lognormal (base e) stochastic distribution model (number 9) and volumes using a triangular distribution (number 6). The third example defines volumes and concentrations using user-specified (tabular) distributions.

```
DISPOSAL WASTEID "100-H-5" TYPE "soil" YEAR 1953
VOLUME 6 5.87600E+03 7.34500E+03 8.81400E+03 TRUNCATE 0.000 1.000
CONTAM1 "H3" " 9 -9.85732E+00 1.26864E+00 TRUNCATE 0.010 0.990
CONTAM2 "C14" " 9 -8.97413E+00 1.26864E+00 TRUNCATE 0.010 0.990
CONTAM6 "Sr90" " 9 -1.08459E+01 1.26864E+00 TRUNCATE 0.010 0.990
CONTAM8 "Cs137" " 9 -8.95037E+00 1.26864E+00 TRUNCATE 0.010 0.990
CONTAM9 "Eu152" " 9 -5.40365E+00 1.26864E+00 TRUNCATE 0.010 0.990
CONTAM14 "U235" " 9 -2.95175E+01 1.26864E+00 TRUNCATE 0.010 0.990
CONTAM15 "U238" " 9 -2.47347E+01 1.26864E+00 TRUNCATE 0.010 0.990
```

```
DISPOSAL WASTEID "100-K-2" TYPE "soil" YEAR 1955
VOLUME 6 5.87600E+03 7.34500E+03 8.81400E+03 TRUNCATE 0.000 1.000
CONTAM1 "H3" " 9 -9.85732E+00 1.26864E+00 TRUNCATE 0.010 0.990
CONTAM2 "C14" " 9 -8.97413E+00 1.26864E+00 TRUNCATE 0.010 0.990
CONTAM6 "Sr90" " 9 -1.08459E+01 1.26864E+00 TRUNCATE 0.010 0.990
CONTAM8 "Cs137" " 9 -8.95037E+00 1.26864E+00 TRUNCATE 0.010 0.990
CONTAM9 "Eu152" " 9 -5.40365E+00 1.26864E+00 TRUNCATE 0.010 0.990
CONTAM14 "U235" " 9 -2.95175E+01 1.26864E+00 TRUNCATE 0.010 0.990
CONTAM15 "U238" " 9 -2.47347E+01 1.26864E+00 TRUNCATE 0.010 0.990
```

```
DISPOSAL WASTEID "200-E-100" TYPE "liquid" YEAR 1945
VOLUME 10 21 0.00E+00 2.40E+00 5.00E-02 2.88E+00 1.00E-01 3.16E+00
1.50E-01 3.38E+00 2.00E-01 3.57E+00 2.50E-01 3.73E+00 3.00E-01
```

3.88E+00	3.50E-01	4.02E+00	4.00E-01	4.14E+00	4.50E-01	4.26E+00			
5.00E-01	4.38E+00	5.50E-01	4.49E+00	6.00E-01	4.61E+00	6.50E-01			
4.73E+00	7.00E-01	4.87E+00	7.50E-01	5.02E+00	8.00E-01	5.18E+00			
8.50E-01	5.37E+00	9.00E-01	5.59E+00	9.50E-01	5.87E+00	1.00E+00			
6.35E+00									
CONTAM1	"H3	"	10	21	0.00E+00	8.58E-08	5.00E-02	2.73E-07	1.00E-01
3.84E-07	1.50E-01	4.71E-07	2.00E-01	5.44E-07	2.50E-01	6.08E-07			
3.00E-01	6.66E-07	3.50E-01	7.20E-07	4.00E-01	7.73E-07	4.50E-01			
8.28E-07	5.00E-01	8.87E-07	5.50E-01	9.48E-07	6.00E-01	1.01E-06			
6.50E-01	1.08E-06	7.00E-01	1.16E-06	7.50E-01	1.24E-06	8.00E-01			
1.32E-06	8.50E-01	1.43E-06	9.00E-01	1.54E-06	9.50E-01	1.70E-06			
1.00E+00	1.97E-06								

#### 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 this keyword's syntax:

```
END
```

#### 4.4.5 EXECUTE Keyword for INVENTORY

The EXECUTE keyword (if present) causes a full analysis to be attempted. The following is this keyword's syntax:

```
EXECUTE
```

If the EXECUTE keyword is not present the analysis will terminate after reading the input data and performing consistency checks. Running the code without the EXECUTE keyword only tests the input data set. The EXECUTE keyword must be present or the analysis will not proceed correctly when INVENTORY is exercised under the control of the ESP.

#### 4.4.6 FILE Keyword for INVENTORY

The FILE keyword is used to specify file names for selected input and output files. The following is this keyword's syntax:

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

The files that can be specified are defined in Table 4.3. File names may include path information and can be up to 200 characters long.

**Table 4.3** Modifiers for the FILE Keyword in INVENTORY

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 optional output file (see Section 4.5.2.3) contains the total release by radionuclide for each year for all realizations.
STOCHOUT	Stochastic data file	This optional output file (see the DEBUG keyword in Section 4.4.1 and the file description in Section 4.5.2.2) contains summary information on generated stochastic variables.
WASTE	Master waste stream disposal action file	This required input file contains waste stream disposal actions (see Section 4.5.1.3).

Multiple FILE keywords must be entered. However, only one modifier and associated file name may be entered on a single FILE keyword record. Examples of the FILE keyword for each file modifier are the following:

```
FILE DECAY "RMDLIB.DAT"      ! Input  - Radioactive decay library
FILE WASTE "CAL-median.daf"  ! Input  - Disposal actions (from INPROC)
FILE MASS "mass_out_inv.dat" ! Output (optional) - Mass balance info
FILE STOCHOUT "stochvar.out" ! Output (optional) - Stochastic value info
```

#### 4.4.7 MASSBALANCE Keyword for INVENTORY

The optional MASSBALANCE keyword is used to specify that a mass balance analysis will be performed. The following is this keyword's syntax:

```
MASSBALANCE
```

The details for the mass balance analysis are specified on the BALANCE keyword (section 2.1.3) in the ESD keyword file. When a mass balance calculation is to be performed, an output file name must be provided on a FILE keyword (see Section 4.4.6). If the MASSBALANCE keyword is not entered, then no mass balance calculations will be performed.

#### 4.4.8 NORMALIZE Keyword for INVENTORY

The optional NORMALIZE keyword is used to normalize the total amount of an analyte from all disposal actions (the units are curies for radionuclides, kilograms for other analytes). The amount is specified as a stochastic distribution. The following is this keyword's syntax:

```
NORMALIZE ["analyte ID" N1 P1 {P2} {P3} ... {Pn}]
```

The "analyte ID" is the ID of an analyte as specified on the ANALYTE keyword in the ESD file (see Section 2.1.1), N1 is the index of the stochastic distribution to be used to define the total amount of the

analyte. The parameters P1, P2, P3, etc. are the parameters necessary to define the statistical distribution. See Section 1.4 for details of the stochastic distribution options. The normalization is applied at the time of site closure as given on the PERIOD keyword in the ESD file (see Section 2.1.20). All radioactive waste streams are decay corrected to the time at which the normalization is applied before the normalization occurs. 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.

#### **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 this keyword's syntax:

```
SEED [V1] ["analyte1" S1] {"analyte2" S2} ... {"analyten" Sn}
```

The first numerical value, V<sub>1</sub>, provides the seed for volume generation. The remainder of the entries are presented in data pairs with an analyte name (in a quote string) followed by the random seed value to be used for that analyte. Values for all analytes must be provided on the SEED keyword or the program will error terminate. Only one SEED keyword should be provided in the keyword file.

An example SEED keyword is as follows:

```
SEED 199044 "Pu239" 23923 "Sr90" 9090
```

This SEED keyword 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 and should all be unique.

#### **4.4.10 SELECT Keyword for INVENTORY**

The SELECT keyword 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 following is this keyword's syntax:

```
SELECT [TYPE N1 "type1" "type2" ... "typeN" | WASTE "filename"]
```

If the TYPE modifier is present, then the number of types to be specified as "N1" is also given and N1 type ID's are also provided in quote strings. The TYPE and WASTE modifiers cannot be specified on the same SELECT keyword or in the same run of the code. 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. The format of this file is described in Section 4.5.1.4. The following example SELECT keyword identifies a file:

```
SELECT WASTE "waste.inp"
```

#### **4.4.11 TIMER Keyword for INVENTORY**

The optional TIMER keyword causes run timing information to be written to the report file (inventory.out). If the TIMER keyword is present, then timing information is generated and printed; otherwise, it is not. The following is this keyword's syntax:

```
TIMER
```

#### **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 file. 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 and can contain up to 72 characters. The following example defines a title for a run of the code.

```
TITLE "Example title line for the inventory keyword file."
```

#### **4.4.13 USER Keyword for INVENTORY**

This keyword is used to identify the individual responsible for preparing the keyword file. If the user name is not supplied, the program will error terminate. The following is this keyword's syntax:

```
USER "quote"
```

The user name is supplied in a quote string and can contain up to 16 characters. The following is an example USER keyword:

```
USER "John Doe"
```

#### **4.4.14 WASTEMAP Keyword for INVENTORY**

The WASTEMAP keyword is used to specify aggregation rules for each of the disposal actions. The following is this keyword's syntax:

```
WASTEMAP [AGGREGATE "aggregation site ID"] [FRACTION N1 P1 {P2} {P3} ...]
          [STREAMS "stream-type-year-string1" {"stream-type-year-string2"} ... ]
```

The WASTEMAP 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 ID (216-A-11), a waste type (liquid, for example), and a year of disposal (1956, for example). The WASTEMAP keyword must also specify the fraction of the disposal action quantity that goes to the named aggregation site. This allows a disposal action to be split between more than one aggregation site. A maximum of three aggregation sites may be used for each disposal action.

The modifier AGGREGATE is always required and is followed by the aggregation site name, declared as a string of up to 30 characters. If the site has the OFFSITE property on the SITE keyword in the ESD keyword file the site name may contain 30 characters, otherwise the site name must be limited 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 1.4. This value is followed by the parameters necessary to define the statistical 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 quote string identifying a waste disposal action. The construction of the string must be performed as follows: the waste stream name is in character positions 1-30 (left justified); the waste type name is in character positions 31-46 (left justified); and the year is in character positions 47-50. Three example WASTEMAP keywords are the following:

```
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 4.4 summarizes these files. The following sections describe the details of these files.

**Table 4.4** 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).
Mass balance results	Output	If mass balance was selected to be performed (MASSBALA 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 always 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 always "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, a file named inv.done is generated. Otherwise, the file named 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 always 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. Table 4.5 contains an example keyword file.

**Table 4.5** 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

The INVENTORY code uses some information provided in the ESD keyword file. The keywords for the ESD keyword file are described in Section 2.1. Table 4.6 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.

**Table 4.6** ESD File Keyword Information Used by INVENTORY

Keyword	Modifiers/Parameters Used	Description of Use
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 <i>windows</i> or <i>unix</i> ; default is <i>unix</i> .
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.
SITE	COMPUTE	This analyte is ignored if the COMPUTE modifier is not present.
	ID	Provides the site ID.
	OFFSITE	This analyte is ignored if the OFFSITE modifier is present.
	TITLE	Title of the release site.
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, 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 are defined by the SITE keywords 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.

Each DISPOSAL keyword provides the information for one disposal action (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. The syntax for providing the volume and concentration information as stochastic distributions is described in Section 1.4. The file must end with an END keyword record. Table 4.7 provides excerpted records from a disposal action keyword file.

**Table 4.7** 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 keyword (see Section 4.4.10) providing the file name for the waste stream selection file.

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 ID of one waste site to be included in the analysis. If the site has the OFFSITE property on the SITE keyword in the ESD keyword file the site ID may contain 30 characters, otherwise the site ID must be limited to 15 characters. The IDs must be unique. This list of waste site IDs is compared against the IDs provided on the DISPOSAL keyword for waste streams in the master disposal waste stream disposal action file. The ID provided in the DISPOSAL keyword must match a ID provided in the Waste Stream Selection file or an error message will be generated. The test for a match of waste stream IDs is only performed if a Waste Stream Selection file has been identified on a SELECT keyword in the inventory.key file. If not specified, then the analysis is performed for all waste streams defined on the DISPOSAL keywords in the master disposal waste stream disposal action file. Table 4.8 provides an example of the Waste Stream Selection file.

**Table 4.8** Example Waste Stream Selection File for INVENTORY

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 4.9. 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 4.9** 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.

Record	Parameter type	Description
		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 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 4.10 provides an example data set for a parent and progeny.

**Table 4.10** 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).

The INVENTORY code 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 at the time when the decay file is opened. 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 types of files are generated for a run of INVENTORY. Table 4.4 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 4.11 provides excerpted records from an analysis summary file. Section 4.6 describes major error messages.



**Table 4.11** Excerpted Records from the inventory.out File

III	N	N	VV	VV	EEEE	N	N	TTTT	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	T	O	O	RRRR	Y	Y
I	N	N	N	V	V	EEE	N	N	T	O	O	RRRR	Y	Y
I	N	N	N	V	V	E	N	N	T	O	O	RR	Y	Y
I	N	NN	V	V	E	N	NN	T	O	O	R	R	Y	Y
III	N	N	V	V	EEEE	N	N	T	OO	R	R	Y	Y	Y

INVENTORY      Version 2.0.063  
Last Modified on 3 Jul 2004

-----  
Generation of Waste Disposal Quantities  
-----

Developed By Battelle Memorial Institute  
Pacific Northwest National Laboratories  
Richland, Washington

Current Run ID = 20040805075910      User Name = Paul W. Eslinger

System Date = 08-05-2004      System Time = 07:59:10.553

Review Signatures

Input Prepared By: \_\_\_\_\_ Date: \_\_\_\_\_

Input Reviewed By: \_\_\_\_\_ Date: \_\_\_\_\_

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

Summary information from the Keyword Control File

Environmental Settings Data file is: /home/ANALYSIS4/CA1\_median/ESD\_CA1\_median.key  
Radionuclide chain decay file is: RMDLIB.DAT  
Mass Balance Results Output file name is:  
Stochastic data output file name is:  
Master Waste Stream Disposal Action file name is: CA1-median.daf  
Execution time progress will be output  
Input activities defined at year of disposal  
Complete execution of this analysis will be attempted

The number of realizations specified for this analysis is: 1

PERIOD Keyword results. Start time: 1944 Stop time: 2070

*User Instructions for the Systems Assessment Capability, Rev. 0, Computer Codes*  
*Volume 1: Inventory, Release, and Transport Modules*

---

Closure: 2070                      Number of years:     127

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: 2070, Found: 2100

The following analytes are included in the analysis

ID	Type	Name
C14	Rad.	Carbon-14
Cs137	Rad.	Cesium-137
Cl36	Rad.	Chlorine-36
I129	Rad.	Iodine-129
Np237	Rad.	Neptunium-237
Pa231	Rad.	Protactinium-231
Ra226	Rad.	Radium-226
Se79	Rad.	Selenium-79
Sr90	Rad.	Strontium-90
Tc99	Rad.	Technetium-99
H3	Rad.	Tritium
Eu152	Rad.	Europium-152
U233	Rad.	Uranium-233
U235	Rad.	Uranium-235
U238	Rad.	Uranium-238

The following aggregation release sites are defined for the analysis

Name	Location
WIPP	Onsite
Yucca	Onsite
SNM	Onsite
offsite	Onsite
WIPP_118-KE-1	Onsite
WIPP_218-E-12B	Onsite
WIPP_218-W-3AE	Onsite
WIPP_218-W-3A	Onsite
WIPP_218-W-4C	Onsite
WIPP_218-W-4B	Onsite

<lines deleted>

Random seed provided for volume and fraction data is:                      332233.

Random seeds provided for the following analytes

Analyte	Seed	Analyte	Seed	Analyte	Seed
C14	3344333.	Cs137	9092349.	Cl36	3312373.
I129	7744558.	Np237	1587129.	Pa231	2340909.
Ra226	9090234.	Se79	2394194.	Sr90	2464234.
Tc99	1414141.	H3	7774449.	Eu152	9234909.
U233	1291290.	U235	1294444.	U238	1444129.

*User Instructions for the Systems Assessment Capability, Rev. 0, Computer Codes*  
*Volume 1: Inventory, Release, and Transport Modules*

---

The following waste types are selected for this analysis

cake  
capsul  
cement

<lines deleted>

TRUSLG  
TRUsf  
WIPP

Waste streams included in the analysis are as follows

4843	100-B-15	100-B-3
100-B-5	100-B-8	100-C-3
100-C-6	100-D-23	100-D-24
100-D-29	100-D-3	100-D-32

<lines deleted>

Normalization will be performed for the following contaminants

None

Title from Keyword Control File:

2004 Composite Analysis 10,000-year Baseline Assessment (Median-Inputs Case)

Title from Environmental Settings File:

2004 Composite Analysis 10,000-year (Median Inputs) Assessment

Title from Chain Decay Data file:

MEPAS

Title from Waste Disposal Action file:

2004 Composite Analysis Baseline Inventory

User from Waste Stream Disposal Action file:

Paul W. Eslinger

Title from Waste Stream Selection File:

Site

Number of contaminants included in the analysis: 15

Number of waste volumes and concentrations: 77509

Number of aggregation data sets in the analysis: 6663

Total number of stochastic variables defined: 84187

Number of table entries for contaminants: 0

Number of table entries for volumes/conc.: 0

Number of table entries for fractions: 0

Total number of table entries: 0

Starting analysis for realization number: 1

Normalization factors for realization 1

Analyte Factor

C14 1.00000E+00

Cs137 1.00000E+00

C136	1.00000E+00
I129	1.00000E+00
Np237	1.00000E+00
Pa231	1.00000E+00
Ra226	1.00000E+00
Se79	1.00000E+00
Sr90	1.00000E+00
Tc99	1.00000E+00
H3	1.00000E+00
Eu152	1.00000E+00
U233	1.00000E+00
U235	1.00000E+00
U238	1.00000E+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 MASSBALANCE be present in the inventory.key file and that a file name has been provided in the FILE MASS keyword 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 4.12 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 4.12** Example Mass Balance Result File from INVENTORY

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 4.13 describes the format of the .RES files.

**Table 4.13** Format of Result Output Files (.RES) in INVENTORY

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.

Record	Parameter / type	Description
	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 <sup>3</sup> )
	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 4.14 provides an example of a result file. The table illustrates the first record of the file and four data sets.

**Table 4.14** Example Result File from INVENTORY

```

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 INVENTORY.ALL File

A file named “inventory.all” is generated for use by the INGRAB data extractor to summarize results of the INVENTORY analysis. The file is not currently used by any other SAC codes. The first information in the file is a header that 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<sup>3</sup>) and disposal amount (Ci for radionuclides and Kg for chemicals). Table 4.15 describes the specific format of the records. The data records are written in comma-separated format for easy import into spreadsheet or plotting programs.

**Table 4.15** Data Record Descriptions for the INVENTORY.ALL 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 4.16 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 4.16** Example Records for a, INVENTORY.ALL 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 4.17 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 4.17** 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,	1.000E+00,
2,	1.000E+00,	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,	1.000E+00,
4,	1.000E+00,	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,	1.000E+00,
6,	1.000E+00,	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 the subdirectory where the INVENTORY code was invoked.

### 4.6 Error Recovery Guidance

There are over 330 error checks programmed into the INVENTORY code. Most of the error checks are designed to detect inconsistencies in input values. This section describes some errors 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 summary output file named inventory.out. Recovery information for some of the most common input errors are as follows:

- **Duplicate Waste Stream Names.** Waste stream names are tested to ensure there are no duplicate names. When the waste stream selection file is supplied (keyword SELECT WASTE in the inventory.key keyword file), names are read from that file. The entries in that file should be modified so the names are unique. 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 typically should not be encountered because the waste stream names master list is generated internally at the time of input.
- **Duplicate Aggregation Site Names.** Aggregation site names are tested to ensure there are no duplicate names. The aggregation site names are read from the ESD keyword file. That file should be checked for duplication of names.
- **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 to ensure there are no duplicate names. If duplicates are found, then the following error message is written. The inventory.key file should be reviewed to be sure there are no duplicates of waste types specified on the SELECT TYPE keyword record.
- **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, an error message is issued. Both files should be checked to determine discrepancies in definitions of the disposal action for the waste stream, waste type, and year of disposal indicated in the error message.

- **Release Site Name Not in Master List.** The primary source of release site names is the SITE keyword in the ESD keyword file. This list is compared against site release names read in the WASTEMAP keyword from the inventory.key file. If a name in the inventory.key file is not in the ESD file, a warning message is issued. 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.
- **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 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, an error message is written. When this error 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 whether 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.
- **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, an error message is written. 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.

## **5.0 VADER – Vadose Zone Release Module**

### **5.1 Purpose**

The purpose of VADER is to calculate quantities of a given analyte, either a specific radioisotope or a non-radioactive chemical released from waste containment to the vadose source zone at regular (annual) time steps based on physico-chemical release models generally expressed as differential equations. The time series of annual releases forms the boundary conditions or source term for subsequent programs such as STOMP to propagate movement of contaminants through the vadose zone to groundwater. VADER also has a capability to import and export quantities of the analyte to simulate remediation transfers.

A single VADER run generates one set of deterministic waste releases for a single analyte at a single site using a single realization of release model coefficients. Monte Carlo simulations are performed by running VADER numerous times with different realizations of model coefficients and inventory. Uncertainties in site inventories, release model coefficients, and timing and quantities of remediation transfers, can be addressed through Monte Carlo runs.

### **5.2 Algorithms and Assumptions**

VADER currently incorporates several solid release models, including models specified by Appendix D of Kincaid et al. (1998) and two liquid pass-through release models. The solids models are for waste forms designated as soil-débris, cake, glass, cement, reactor core, and reactor-compartment waste forms. In addition the glass model has two variants, and there are special models for specific analytes in designated waste, such as the C14 model for carbon-14 in core waste. Models are distinguished by the release mechanism they represent, for example: diffusion, dissolution, and so forth. 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 for input to STOMP. The river release model likewise passes liquid releases from inventory directly into the river (body of water) also with no retardation or decay for input to GWDROP. VADER does not have a specific atmospheric release model. In the SAC context, releases of gaseous substances to the atmosphere are handled in the Inventory Module.

During any given run, VADER uses the input annual time series of inventory deposits, plus net remediation deposits, 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. VADER also maintains an output file of quantities released and remaining at the site (`vader.table` file) for mass balance.

Most of the release models in VADER do not explicitly account for a delay in release from an intact waste container after the waste is placed in the environment. However, the RXCOMP model does explicitly model containment degradation as a function of steel corrosion rate. A delayed start date for onset of release is provided for all models to mimic the effect of a container.

VADER initiates a run by first reading a text file of keyword commands that directs processing. VADER then reads inventory and remediation data files, performs release calculations, and writes several output

files. These output files support mass-balance calculations, provide source term input to STOMP, support remediation transfers, and provide ESP information for managing overall analysis processing.

### **5.2.1 Limitations**

This version of VADER accounts for simple decay of radioactive contaminants but not chain decay between analytes. 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 in terms of releases into the environment.

VADER calculates quantities released to or injected into the release zone (vadose zone or river) without attempting to account for transport away from the release zone. Releases are considered to be at a point source. Once released, transport is the function of the STOMP, CFEST, and MASS2 modules.

### **5.2.2 Waste Forms in VADER**

Waste forms and release models may have the same name or identifier, but they refer to distinct entities. Accordingly, in this user guide, the release models are identified with upper case letters and the waste forms are identified with lower-case letters.

The waste forms handled by VADER are described in BHI (1999), Appendix D of Kincaid et al. (1998), and Riley and Lo Presti (2004). 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. Each site may have one or more waste forms. The INVENTORY file contains annual quantities deposited (or generated) at the site, decayed to the year of deposit, for each waste form. In the inventory file (see Section 5.5.1.2) quantities of a given contaminant analyte are assigned to one or more of several waste forms such as soildebris, saltcake, cement, glass, reactor core, liquid, or river. Each waste form at a given site has assigned to it a unique identifier. In some cases, the name refers to routing. For example, “river” wastes are liquid wastes directed to the Columbia River. Each waste form is assigned a unique default release model such as SOIL, CAKE, CMNT, GLASS, GLASBM, CORE, RXCOMP, LIQUID, or RIVER. Specific characteristics of the waste form such as density, solubility, or moisture content are expressed as model coefficient values for the given waste release model. Although the parameters are properties of the waste itself, or the waste site, they are expressed through the model specification.

The default release model associated with a given waste form depends on what the waste form is called in the INVENTORY file. Release model names can be embedded within a waste form name. For example, the “soiltg” waste form name would refer to the SOIL model by default.

Waste forms can also be explicitly assigned or mapped to a release model through the MAP keyword (see below). For example, some sites have inventories of 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 (an instantaneous pass-through process). But this waste may also be released by a different model by explicitly associating it with that model with the MAP keyword. If the carbon tetrachloride in the “liquid” waste is mapped to, or associated with the SOIL model in the INVENTORY file, it will be released using the SOIL release model. Since each VADER run works on only one analyte at a time, runs can be set up to run the different analytes in a waste form with different models. For example, CCl<sub>4</sub>

could be released from the “liquid” waste with the SOIL desorption/dissolution model, and technitium-99 could be released in another run with the LIQUIDS pass-through model.

### 5.2.3 Release Models

The release model algorithms in VADER are summarized in Table 5.1. The mathematical derivations for the release models are fully documented in Appendix D of Kincaid et al. (1998) and CCN 0512242 (BHI 1999) and Riley and Lo Presti (2004). The information provided here is intended only to highlight model assumptions and implementation features. In the equations described below in further detail,  $dM/dt$  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  $M_i$ . For all models except instantaneous release models (i.e., liquid release models), a fourth-order Runge-Kutta solution for  $dM/dt$  is applied for each time step. This allows for accurate accounting for time-dependent coefficients, such as the recharge rate  $Q_w$ . The release models can be summarized into categories such as the following:

- Whether release is calculated from initial stock  $M$  (non-depleting inventory), current stock  $M$  (depleting inventory), or is not calculated from  $M$  at all.
- Whether release is a function of elapsed time from deposit (as in the CMNT model) or not a function of elapsed time at all.
- Whether release specifies a specific physical process, or the physical process is unspecified.
- Whether release will be constant or will vary over time.
- Whether the input parameters are time-dependent or not. Currently only recharge rate is implemented as a time-dependent parameter.

**Table 5.1** Summary of release models implemented in VADER

Physical Principle	Example Equation	Time-Profile Behavior for Quantity Released	Based on Initial or Current Inventory	Equations in Kincaid et al. (1998) Appendix D	VADER Model Name
Release congruent with matrix dissolution	$dM_c / dt = M_c C_r A / M_{steel}$	Release rate is a constant fraction of initial quantity	Initial $M^{(a)}$		RXCOMP
Release congruent with matrix dissolution	$dM / dt = M_o Q_w A C_{wnsc} / M_{matrix}$	Release rate is a constant fraction of initial quantity. “Matrix” is e. g., saltcake or sludge.	Initial $M^{(a)}$	D.45 - D.51	CAKE
Diffusion of analyte to surface of structure	$dM / dt = M_o A / V \sqrt{D_c / \pi t}$	Release rate depends on elapsed time from deposit, therefore need to	Initial $M^{(a)}$	D.61 - D.63	CMNT

Physical Principle	Example Equation	Time-Profile Behavior for Quantity Released	Based on Initial or Current Inventory	Equations in Kincaid et al. (1998) Appendix D	VADER Model Name
		keep track of each year's deposit.			
Dissolution of analyte from soil	$dM / dt = C_{sol} A Q_w$	Release rate is a constant quantity based on capacity to dissolve	Not based on M	D.28 - D.44	SOIL ( $C_s$ phase) CSOL
Desorption of analyte from soil	$dM / dt = MAQ_w / (\theta RAh)$	Variable release rate is based on capacity of soil to retard movement of analyte	Current $M^{(b)}$	D.28 - D.44	SOIL ( $K_d$ phase)
Unspecified fraction	$dM / dt = M_o F_r$	Release rate is a constant fraction of initial quantity	Initial $M^{(a)}$	D.64 - D.67  D.52 - D.60	CORE ( $F_r$ small) LIQUID ( $F_r=1$ ) RIVER ( $F_r=1$ ) GLASS ( $F_g$ small)
Release time-dependent	$dM / dt = f(Q_w(t), t) * M_o$	Empirical glass model for fractional release rate. Release increases to asymptotic value over time, and varies as $Q_w$ varies over time as well.	Initial $M^{(a)}$		GLASBM
(a) Non-depleting inventory release model. Each annual deposit is tracked separately. (b) Depleting inventory release model. All deposits are pooled into one inventory quantity. (c) The equations are provided in Appendix D of Kincaid et al. (1998).					

Table 5.1 generally summarizes the release models, and Table 5.2 identifies the waste forms processed by these release models in VADER. 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 (e. g., the SOIL model). In nondepleting inventory models, the releases are based on initial inventories (e. g., the CMNT, CAKE, and GLASBM models).

**Table 5.2** Waste Forms Processed by VADER

<b>Waste Form</b>	<b>Characteristics</b>	<b>Source</b>	<b>Release Mechanism</b>	<b>Default Models</b>	<b>Routing</b>
Soil-Débris	High permeability	Soils, cribs, trenches, etc.	Desorption/ Solubility	SOIL CSOL	Top of vadose zone
Glass	Very low permeability/ Immobile	Vitrification	Dissolution of glass	GLASS GLASBM	Top of vadose zone
Saltcake/ Sludge	High permeability	Tank waste	Dissolution congruent to saltcake waste form	CAKE	Top of vadose zone
Concrete/ cement	Low/medium permeability	Misc.	Diffusion	CMNT	Top of vadose zone
Reactor Blocks	Other	Irradiated solids	Corrosion	CORE	Top of vadose zone
Reactor Compartments	Other	Irradiated solids	Corrosion of containment with congruent release	RXCOMP	Top of vadose zone
Special	Other	Other	Specified by table, mechanism not specified	BYPASS	Top of vadose zone to liquid or to solid pathway
Liquid	Instantaneous	Various	Pass-through	LIQUID	Top of vadose zone
River	Instantaneous	Reactor coolant	Pass-through	RIVER	River

The implementation accommodates annual changes in the source term from which to calculate annual releases. Up until the time of site closure, 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.

To reduce mathematical error in the solution of differential equations, the time step used should be no longer than one half life of a radioactive analyte (Stark 1970). Applying this rule avoids numerical

instabilities in the time series results arising from short-lived transients. For SAC radioisotopes, all the expected radionuclides have half-lives 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 probably adequate for analytes having a half-life of four 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  $\beta$  = the bulk density of the source zone soil or vadose zone soil ( $\text{g}/\text{cm}^3$ )  
 $K_d$  = linear equilibrium sorption coefficient for the contaminant ( $\text{cm}^3/\text{g}$ )  
 $\theta_w$  = 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

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

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

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:

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

where  $\lambda$  = the decay constant for radioactive analytes (1/yr)  
 $t$  = the length of a time step (typically one year)  
 $e^{-\lambda t}$  = 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 saltcake 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 saltcake is assumed to remain constant over time. Currently, the physical mix of saltcake, 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 saltcake. As the saltcake dissolves, the embedded contaminants are leached congruently with the dissolving saltcake. Annual release quantities are calculated based on net accumulated annual saltcake inventory; thus, the saltcake model is a nondepleting

release model. The model requires the groundwater recharge,  $Q_w$ , as a function of time to modulate the quantity of saltcake 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 saltcake mass directly. Instead, it provides waste volume, which is taken to be the volume of saltcake added to the site inventory annually. VADER converts the net accumulated volume to mass units using the saltcake 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 saltcake waste at each site. If the model specifications do not set a density, a default value of  $1.5 \text{ g/cm}^3$  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 ( $\text{cm}^2$ )  
 $C_{sol}$  = the solubility limited concentration for an analyte in water ( $\text{Ci/cm}^3$  or  $\text{g/cm}^3$ )  
 $M_{msco}$  = the original saltcake mass {  $C_{sol}$  and  $M_{msco}$  have the same mass units } ( $\text{Ci/cm}^3$  or  $\text{g/cm}^3$ ).

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

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

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

where  $\lambda$  = the decay constant for radioactive analytes (1/yr)  
 $t$  = the length of a time step (typically one year)  
 $e^{-\lambda t}$  = 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  $M_0$  = the original amount of contaminant in the cement (Ci or kg)  
 $A$  = the cross-sectional area of the waste exposed to the release mechanism (cm<sup>2</sup>)  
 $V$  = the volume of the waste (cm<sup>3</sup>)  
 $D_c$  = the effective diffusion coefficient of the contaminant within a cement waste form (cm<sup>2</sup>/yr)  
 $\pi$  = the mathematical constant  
 $\Delta t$  = 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:

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

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

where  $\lambda$  = the decay constant for radioactive analytes (1/yr)  
 $t$  = the length of a time step (typically one year)  
 $e^{-\lambda t}$  = 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  $F_r$  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:

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

where  $\lambda$  = the decay constant for radioactive analytes (1/yr)

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

$\Delta t$  = 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 Reactor Compartment Model Release Algorithm

The reactor compartment model is applied to the naval reactors stored in trenches at Hanford (e. g., the 218-E-12B site). Each unit consists of a reactor vessel contained by a steel hull. Although there are five sizes or types of reactor compartments (for cruisers, submarines, etc.), they are all treated the same way in that they are given the same dimensions and mass based on weighted calculations (Riley 2003; DOE 1996). The reactor compartment model has two stages. The first stage is hull corrosion in which the hull steel corrodes to the point of breaching at some date *STARTREL* which is calculated from input parameters as follows:

$$STARTREL = t / C_s + \text{Deposit year}$$

where  $t$  is hull steel thickness (e. g., 1.905 cm or 3/4 inch)

$C_s$  is steel corrosion rate in cm/yr

Deposit year is the year the naval reactor is placed into inventory (e. g. 1986 to 2026 A. D.)

*STARTREL* is the MODELS *STARTREL* parameter, which can be optionally calculated as well as supplied as an input parameter.

Once the hull is breached, the second stage is release of contaminants from the reactor vessel. This release is considered to be congruent with degradation of the reactor vessel and is expressed by the following equations:

$$\frac{dM}{dt} = M_c F_r$$

$$F_r = C_r A / M_o$$

where  $M_c$  = the original inventory quantity (mass or activity) of the contaminant in kg or Ci  
 $F_r$  = a calculated fractional release rate to apply to original inventory M  
 $C_r$  = the corrosion rate for the reactor vessel in kg/(dm<sup>2</sup> \* yr)  
 $A$  = the reactor vessel steel surface area in cm<sup>2</sup>  
 $M_o$  = the mass of steel in the reactor vessel in kg

Parameters  $t$ ,  $C_s$ , and  $C_r$  is read from the input vader keyword file. Parameters  $A$  and  $M_o$  are calculated internally in the RXCOMP code based on reactor compartment waste volumes as read from the INVENTORY file. It is expected that  $C_s$  and  $C_r$  would be treated as stochastic, since the reactor compartment volumes, masses, and dimensions are considered to be invariant for the purposes of SAC simulations.

Although each reactor compartment could be treated as an individual units with its own (calculated) start release date, in this revision of VADER all reactor compartments are considered to have the same elapsed time from deposit to release start date, based on either entry of STARTREL or on calculation of elapsed time from deposit to start release  $t/C_s$ .

The contaminant release rate is dependent on the original contaminant inventory  $M_c$  so the cumulative inventory without accounting for depletion is used to calculate the release each year. That is, the release rate is constant until the inventory is depleted. The inventory at any given time step  $i$  is given by the equations:

$$Release_i = \frac{dM_i}{dt} e^{-\lambda \Delta t}$$

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

where  $\lambda$  = the decay constant for radioactive analytes (1/yr)  
 $e^{-\lambda t}$  = set to 1 for nonradioactive analytes  
 $\Delta t$  = 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.

The reactor compartments dimensions hardwired into the RXCOMP model are given as:

$M_o = 108,000$  kg (mass of steel containing contaminants to release). A multiple of this is delivered when the volume read from the inventory file is multiplied by the above factor.

$V = 1050$  m<sup>3</sup> (overall reactor compartment volume)

$A = 31,564$  dm<sup>2</sup> (overall reactor compartment surface area)

The number of reactor compartments deposited per year is given indirectly in the INVENTORY file under rxcomp wastes in terms of waste volume in m<sup>3</sup> deposited per year. Since RXCOMP model requires  $M_o$  expressed in kg, the annual volumes read are multiplied by the factor 102.85 kg/m<sup>3</sup> to arrive at annual mass of steel  $M_o$ . The model also requires total surface area  $A$  accessible to the corrosion mechanism. With the surface area for an individual reactor known to be 31,564 dm<sup>2</sup>, to obtain effective surface area for the annual deposit the formula is (Volume read / V \* A).

### 5.2.3.6 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:

$$\text{Release}_i = \frac{dM_i}{dt} = M_i F$$

$$M_{i+1} = M_i - \frac{dM_i}{dt}$$

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:

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

where  $\lambda$  = the decay constant for radioactive analytes (1/yr)

$t$  = the length of a time step (typically one year)

$e^{-\lambda t}$  = 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.

### 5.2.3.7 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:

$$\text{Release}_i = \frac{dM_i}{dt} = M_i F$$
$$M_{i+1} = M_i - \frac{dM_i}{dt}$$

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:

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

where  $\lambda$  = the decay constant for radioactive analytes (1/yr)  
 $t$  = the length of a time step (typically one year)  
 $e^{-\lambda t}$  = 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 waste is SOIL waste to be placed in the working site inventory as of a given year, unless the waste form is specifically declared in the “remediate” file. Likewise, the data units are “Ci” or “kg” depending on the analyte unless specifically declared.

- **Transfers of wastes from previously computed sites 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 sites 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 5.4.16). 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**

This section describes how to run the VADER code.

#### **5.3.1 Location in the Processing Sequence**

The general processing sequence for SAC environmental runs is outlined in Figure 1.1. The inventory data for input to VADER is generated by the INVENTORY program. The vader.key file is generated by 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 the STOMP and GWDROP programs. Thus, the VADER and STOMP codes run in tandem.

#### **5.3.2 How the VADER Code Is Invoked**

Under a Windows operating system VADER is executed by entering one of the following commands in a DOS window:

```
VADER.exe .  
VADER.exe path
```



Under the Linux operating system VADER is executed through the following Bourne Shell or C Shell command:

```
VADER.exe .
VADER.exe path
```

For both operating systems the first command sets the working directory to the directory where VADER was invoked. The second command sets the working directory to the directory identified by the input path. The working directory is expected to contain the vader.key file, the inventory file, and the STOMP template file (input-esd), and will contain all the output files written by VADER.

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 *path* does not exist (VADER was invoked without a command-line argument), the path is set in effect to the working directory.

If application of the above rules does not identify a valid working directory VADER writes the vader.fail file and terminates execution. If vader.key is not present in the working directory VADER writes a vader.fail file and terminates execution. The following is an example command line entry following SAC current practice:

```
VADER /home/ANALYSIS4/CA1_median/vadose/200-E-12/H3/1/
```

## 5.4 VADER Keyword Descriptions

This section lists VADER keywords and command structures. Nominally the 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. Table 5.3 provides a summary of keywords used in VADER.

**Table 5.3** Summary of Keywords Used by VADER

Keyword	Description
ANALYTE	The ANALYTE keyword specifies the single analyte to be used in the simulation.
BYPASS	Load a release curve and waste volume to pass-through to the STOMP template. This method for injecting releases directly into the vadose zone operates in parallel to the release mechanisms specified via the MODELS keyword.
CONTROL	Control whether VADER reports informational messages or terminates abnormally. Pertains to whether all wastes must be released, or only some wastes are released. Default: requires all wastes found in INVENTORY to be assigned a release model.
CUTOFF	The optional CUTOFF keyword supplies lower quantity cutoff values for reporting annual quantity and annual volume releases to the vadose zone via the STOMP template.

<b>Keyword</b>	<b>Description</b>
DEBUG	The optional DEBUG keyword is used to activate additional output options that are useful for code debugging purposes. Debug output is written to VADER.debug file.
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 and supplies a terminating message to be written to the Log file.
FILE(S)	The optional FILE(S) keyword can be used to enter file names that are different from default file names. The keyword may be entered as either "FILE" or "FILES".
INFILT	This optional keyword allows entry of a water recharge value effective over a period of time as a lookup table. A series of INFILT lines creates a step function.
MAP	The optional MAP keyword provides a means for explicitly assigning a release model to a specific waste form present in the INVENTORY.
MODELS	The MODELS keyword specifies which release model simulations to run and supplies model release coefficients. Release models may have names which contain the embedded generic release models such as SOIL, CMNT, CORE, or RXCOMP.
NOCOMPUT	Check keywords and load inventory and remediations, then stop
PATH	The optional PATH keyword overrides the command-line argument keyword file path name. It states where to find input files and to write output files.
PERIOD	The PERIOD keyword identifies the start and stop years for the release simulation.
REALIZAT	The REALIZATION keyword identifies the single realization to be simulated.
REMIAT	The optional REMEDIAT keyword indicates the input files that define incoming remediation quantities, and also provides information for calculating outgoing remediation quantities.
SITE	The SITE keyword is used to specify the working aggregate release site to process.
STEPS	Post-hoc adaptive time-stepping for output records (STOMP input file)
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.

#### **5.4.1 ANALYTE Keyword for VADER**

The ANALYTE keyword specifies the analyte to be used in the simulation. There is no default value. 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 this keyword's syntax:

```
ANALYTE [ ID="Quote1" ] [ TYPE="Quote2" ] { HALFLIFE=N1 } { SPECIFIC=N2 }
        { UNIT="Quote3" }
```

Modifiers associated with the ANALYTE keyword are described in Table 5.4. These modifiers, together with their associated quote strings or numerical data, can be entered in any order.

**Table 5.4** Modifiers for the ANALYTE Keyword in VADER

Modifier	Description
HALFLIFE	The numerical entry associated with the HALFLIFE modifier is the half-life of a radioactive isotope in years. Although it is expected to apply to radioactive substances, it can also be entered for a nonradioactive analyte in order to emulate its degradation chemically in an exponentially decreasing manner. If not entered, defaults to HUGE (largest possible machine number) to specify infinite halflife.  Also if set to 0, defaults to HUGE to specify infinite halflife.
ID	The quote string associated with the ID modifier is an analyte identification string up to six characters in length.
SPECIFIC	The numerical entry associated with the SPECIFIC modifier is the Specific activity of a radioactive isotope in Ci/kg.
TYPE	The quote string associated with the TYPE modifier is a two-character string to specify one of four analyte types: <ul style="list-style-type: none"> <li>• NR - Inorganic radioactive isotope or compound containing a radioactive isotope</li> <li>• NS - Nonradioactive (stable) isotope or inorganic compound</li> <li>• OR - Organic radioactive isotope or compound containing a radioactive isotope</li> <li>• OS - Organic compound containing only nonradioactive (stable) isotopes</li> </ul>
UNIT	The quote string associated with the UNIT modifier is analyte quantity units up to two characters in length. This defines the quantity units for release calculations. If not entered, the default action is to set units according to the TYPE modifier value: <ul style="list-style-type: none"> <li>• NR and OR – set to “Ci”</li> <li>• NS and OS – set to “kg”</li> </ul> <p>Note: VADER uses volumetric units from the INVENTORY file.</p>

As an example, the following ANALYTE keywords would select Chromium-VI and neptunium-237 for analysis (in different runs). The TYPE modifier describes the analyte, and the HALFLIFE modifier supplies a radioisotope halflife. For Chromium-VI, which is not radioactive, the HALFLIFE modifier is not used. Default units are defined by the TYPE modifier and are “kg” and “Ci” respectively.

```
ANALYTE ID="CrVI" TYPE="NS"
ANALYTE ID="Np237" HALFLIFE 2.14e06 TYPE="NR"
```

### **5.4.2 BYPASS Keyword for VADER**

The optional BYPASS keyword supplies a mechanism to "inject" release profiles which are calculated off-line into the vadose zone. The following is this keyword's syntax:

```
BYPASS [FILE="Quote1"]
```

The quote string associated with the FILE modifier contains a file name of up to 180 characters in length. The BYPASS keyword causes the identified table file (see Section 5.5.2.1) containing release quantities and volumes to be read into VADER. These releases are passed through (without decay) to be added to either the liquid releases or solids releases as specified. There is no provision when using the BYPASS keyword to explicitly enter a containment failure date analogous to the STARTREL modifier of the MODELS keyword. It is expected that the release curve provided in the table accounts for containment degradation, if any. Release quantities and optional deposits are assumed decayed to the current year.

Two examples uses of this keyword are the following:

```
BYPASS FILE="ILAW"  
BYPASS FILE="/home/ANALYSIS/EIS/vadose/Site999/Sr90/01/WasteXX.table"
```

The first example causes a file named "ILAW" in the working directory to be read. The second example provides the full pathname to the file "WasteXX.table".

### **5.4.3 CONTROL Keyword for VADER**

The optional CONTROL keyword controls VADER when it encounters certain data configurations in the inventory data set and specified release models. The following is this keyword's syntax:

```
CONTROL [WASTES=["ALL" | "SELECTED"]]
```

When the string "ALL" is associated with the WASTES modifier, VADER verifies that all waste forms found in the inventory data have been assigned a release model specified in the vader.key file. If all required models are present, VADER processes all the waste forms. If at least one waste form does not have a release model, VADER terminates and writes the vader.fail file to signal ESP that a terminal error was encountered. Omission of the CONTROL keyword has the same effect as entering the following keyword:

```
CONTROL WASTES="ALL"
```

When the string "SELECTED" is associated with the WASTES modifier, VADER processes the release models present in the keyword file, and ignores the waste forms for which no release model is specified. Informational messages are written to the vader.log file to indicate which waste forms are represented in the VADER results.

#### **5.4.4 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 out 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 entire simulation time period to vader.table. The CUTOFF option is applied to the annual releases before the STEPS option is applied. The following is this keyword's syntax:

```
CUTOFF {ANALYTE=Na} {VOLUME=Nv}
```

The numerical values Na associated with the ANALYTE modifier is the smallest quantity of analyte released (for both the AGGREGATE solids release and LIQUIDS release) to write to the STOMP INPUT file. Cutoff value units for the ANALYTE modifier are activity or mass units (Ci/yr or kg/yr) depending on the analyte type. The numerical value Nv associated with the VOLUME modifier is the smallest volume of liquids release water to write to the STOMP input template. Cutoff value units for the VOLUME modifier are m<sup>3</sup>/yr or L/yr for volumes, as specified in the volumes file. Both Na and Nv default to 0 unless entered using this keyword. The default setting has the effect of writing all records over the simulation time period, no matter how small the release.

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<sup>-12</sup> Ci/yr, the INPUT file will contain quantities down to approximately 3.16 x 10<sup>-20</sup> Ci/sec. The same time scale considerations apply to volumes.

#### **5.4.5 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 14 optional modifiers, and multiple DEBUG keywords can be entered. Outputs activated by this keyword are written to the vader.debug file. The following is this keyword's syntax:

```
DEBUG {C14} {CAKE} {CMNT} {CORE} {CSOL} {GLASS} {GLASBM} {INVENTOR}  
      {LIQUID} {LOADING} {REMEDiate} {RIVER} {RXCOMP} {SOIL}
```

Table 5.5 shows the portions of VADER to be debugged.

**Table 5.5** Modifiers for the DEBUG Keyword in VADER

<b>Modifier</b>	<b>Portion of Code to Debug</b>
C14	C14 release model routines.
CAKE	Saltcake release model routines.
CMNT	Cement release model routines.
CORE	Reactor core release model routines.
CSOL	Solubility control release model routines
GLASS	Original glass release model routines.
GLASBM	Bacon/McGrail glass release model for ILAW.
INVENTOR	Inventory processing.
LIQUID	Liquid release model routines.
LOADING	Loading the inventory and remediation data into arrays.
REMEDIAT	Remediation processing routines (input and outgoing).
RIVER	River release model routines.
RXCOMP	Reactor compartment release model routines.
SOIL	Soil release model routines.

An example use of the DEBUG keyword to activate the CAKE, CMNT, and SOIL output options is the following:

```
DEBUG CAKE  
DEBUG CMNT SOIL
```

#### **5.4.6 DELTAT Keyword for VADER**

The optional DELTAT keyword supplies the time step for numerical solution of the release model differential equations. The default setting when the DELTAT keyword is omitted is 1 for one-year time-steps. The following is this keyword's syntax:

```
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 fourth order Runge-Kutta algorithm for all release models for solid waste.

### 5.4.7 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. If the END keyword is missing all lines to End-of-File will be treated as keyword lines. The following is this keyword's syntax:

```
END {"message" }
```

If a quote string with at least one character is entered, it is written to the log file upon run completion.

### 5.4.8 FILE Keyword for VADER

The optional FILE keyword can be used to enter file names that are different from the default file names shown in Table 5.6. 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 used. Four files may have their names changed; the other file names are hard-coded. If a filename (NAME) appears without a slash, it is treated as a filename to be found in the working directory, which is specified in the vader.key pathname (command line) or via the PATH keyword. The following is this keyword's syntax:

```
FILE [DONE | LOG_FILE | INVENTORY | REMEDIATE | OUTPUT_TABLE |  
STOMP_TEMPLATE | RIVER ] [NAME="quotel" ] {BINARY} {NOSTOMP}
```

**Table 5.6** Default File Names for VADER

Default File Name	Where referred to	OPEN	File Type
vader.key	Command line path name	READONLY	ASCII
input-esp	Internal	READONLY	ASCII
input	Internal	CREATE	ASCII
inv.dat	vader.key	READONLY	ASCII
remediate (input)	vader.key	READONLY	ASCII
vader.rem (output)	vader.key	CREATE	ASCII
vader.log	vader.key	CREATE	ASCII
vader.table	vader.key	CREATE	ASCII or BINARY
vader.river	vader.key	CREATE	ASCII
vader.running	Internal	SCRATCH	ASCII
vader.done	Internal	CREATE	ASCII
vader.fail	Internal	CREATE	ASCII
vader.debug	Internal	CREATE	ASCII

The modifiers for the FILE keyword are described in Table 5.7.

**Table 5.7** Modifiers for the FILE Keyword in VADER

<b>Modifier</b>	<b>Description</b>
NAME	The quote string associated with the NAME modifier contains the name of the file to be opened. File names are entered as quote strings and may contain path names up to 132 characters long. This modifier is required except when the FILE STOMP_TEMPLATE NOCOMPUTE option is used.
BINARY	The BINARY modifier is used to specify that the given file is in binary format. If not specified, the file type is assumed to be ASCII.
DONE	The DONE modifier identifies the output vader.done file.
INVENTORY	The INVENTORY modifier identifies an input inventory file generated by the inventory module (Default inv.dat)
OUTPUT_TABLE	The OUTPUT_TABLE modifier identifies the output file to contain tabulated release results (default vader.table)
REMEDiate	The REMEDIATE modifier identifies the input remediation file.
RIVER	The RIVER modifier identifies the output file to contain releases to the river (default vader.river).
STOMP_TEMPLATE	The STOMP_TEMPLATE modifier identifies the STOMP template file to load releases into (default input).
NOSTOMP	The NOSTOMP modifier applies only when the STOMP_TEMPLATE modifier is used. If present, the NOSTOMP modifier causes VADER to skip STOMP template file processing – it does not read the “input-esp” file nor write the “input” file.

When FILE keyword lines are absent, the default file names are set as shown in Table 5.7.

The following additional information applies to file operations:

- File names for incoming remedial actions are specified by the REMEDIAT keyword (see Section 5.4.16). Input remediation file names are vader.rem and remediate.
- The vader.running, vader.done and vader.fail file names are hardwired and cannot be specified by a FILE keyword.
- The vader.debug file is not accessible via the keyword definitions. 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.running file disappears.
- VADER can be 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 NOSTOMP turns off all STOMP template processing. The NAME modifier is not required in this case.



The following set of example keywords specifies a full set of files for running VADER for Realization 66 at Site 666 for Strontium-90. File names can include path information:

```
FILE INVENTORY  NAME="INV/Real66.inv"  OPEN=READONLY
FILE LOG_FILE   NAME="Site_666_Sr90_66.log"  OPEN= CREATE
FILE OUTPUT_TABLE  NAME="Site_666_Sr90_66.out"  OPEN=CREATE TYPE=BINARY
```

### 5.4.9 INFILTRA Keyword for VADER

The optional INFILTRA keyword allows entry of time-dependent infiltration or recharge ( $Q_w$ ) values at a given site as a lookup table. In this formulation, the infiltration or recharge table is represented as a step function, with each keyword line representing a period of constant  $Q_w$ . Multiple INFILTRATION keywords can be entered. Infiltration values entered by the INFILTRATION keywords apply to all release models for which the  $Q_w$  parameter is not entered specifically for a SOIL or CAKE release model (see Section 5.4.11). The following is this keyword's syntax:

```
INFILTRA [START=Year1] [END=Year2] [Qw] {UNIT="quote1" }
```

The value Year1 associated with the START modifier is the starting year for the infiltration rate  $Q_w$ . 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 a constant infiltration rate at the site. 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 built into VADER. Currently, conversion from m/yr, mm/yr and in/yr to cm/yr are implemented. If the START and END modifiers are not present, the START and END years from the PERIOD keyword (see Section 5.4.14) apply. If the entered infiltration values do not cover the entire span of years from the PERIOD keyword, the entered values are extended to earlier or later years as needed to cover all years of simulation, with an information message written to the LOG file.

If a value for  $Q_w$  is entered for a specific release model in the MODELS keyword, it overrides INFILTRA keyword entries for calculations pertinent to that release model. When no INFILTRA lines are present, the  $Q_w$  value specific to each release model keyword entry (see Section 5.4.11) is used over the entire simulation period for that release model. There is no attempt to reconcile different release model  $Q_w$  entries, as it is conceivable that it is desirable to use different recharge rates for different release models. When no INFILTRA keywords are present and no MODELS keyword  $Q_w$  parameter entries are present for models that use recharge rate, 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. To use the recharge table, do not specify the  $Q_w$  parameter on the MODELS keyword.

```
MODELS CAKE A=1234 Cs=0.2345 R=3.33
INFILTRA START=1944 END=1988 Qw=23.45 UNIT="cm/yr"
INFILTRA 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 3050 A.D., the ending simulation period year. To have  $Q_w$  be set to a different value, it must be explicitly set. Also note the handling of different units in the INFILTRA keywords.

```
PERIOD START=1944 STOP=3050 CLOSURE=2050
MODELS CAKE A=1234 Cs=0.2345 Qw=20 R=2.25
INFILTRA START=1944 END=1988 Qw=234.5 UNIT="mm/yr"
INFILTRA START=1989 END=2050 Qw=123.4 UNIT="mm/yr"
```

In the following example, the  $Q_w=20$  cm/yr from the CAKE model entry covers the entire simulation period as given in the PERIOD keyword. But that value of  $Q_w$  does not apply to the CSOL model, so that CSOL has no recharge rate specified. If no INFILTRA keywords are defined VADER treats this as an error and terminates the run.

```
MODELS CAKE A=1234 Cs=0.2345 R=1.11 Qw=20
MODELS CSOL Cs=.0023 A=123456
```

The CAKE model requires a groundwater recharge. In the following example, where no INFILTRATION keywords are entered, VADER prints an error message and terminates the run.

```
MODELS CAKE A=123456 R=1.23 Cs=0.2345
```

#### **5.4.10 MAP Keyword for VADER**

The optional MAP keyword provides a means for explicitly assigning a release model to a specific waste form present in the INVENTORY. This method complements the implicit association of INVENTORY waste to a release model through the waste name. Multiple MAP keywords can be specified. The following is this keyword's syntax:

```
MAP [EXPLICIT] WASTE="Quote1" MODEL="Quote2"
```

The quote string associated with the WASTE modifier defines the waste name and quote string associated with the MODEL modifier defines the release model name. The optional EXPLICIT modifier, when present, tells VADER to work with the exact waste name and do an exact compare of "Quote1" with the waste names found in the inventory. When EXPLICIT is not present, VADER does an embedded comparison for the root waste name, which has the effect of mapping all waste name variants with the same root to the release model.

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 to "process" the CMNTxx wastes. But if the following MAP keyword were present

```
MAP Waste "CMNTxx" Model "SOIL"
```

were present, the CMNTxx waste would be released using the SOIL release model.

Additional examples (can have all of these in one vader.key file) are the following:

```
MAP Waste "MUCK" Model "SOIL"  
MAP Waste "SOIL" Model "SOIL"  
MAP EXPLICIT Waste "SOILXX" Model "CMNT"  
MAP EXPLICIT Waste "MOLE" Model "CMNT"
```

In the first example, all waste forms with "MUCK" embedded in the waste form name, such as MUCK, MUCKXX, 1MUCKY, etc., are assigned to the SOIL release model. In the second example, all waste forms with SOIL embedded in the waste form name, such as SOILTC, SOIL666, or XSOIL are assigned to the SOIL release model. This example expresses the default VADER action. In the third line, SOILXX wastes are to be released by the CMNT release model, while other SOILxx wastes are released by the default SOIL release model. CMNT wastes continue to be released by the same CMNT model. In the fourth line, only the waste form MOLE is assigned to the CMNT release model. Waste forms MOLE1, AMOLE, or BMOLE2, having no default model name, would not be assigned a release model at all, and VADER would not calculate releases from these wastes.

The VADER default method for assigning release models to waste forms is to rely on the name of the waste form name containing embedded in it one of the release model names. In this manner all variants of a waste form can be assigned or mapped to a single release model in a concise manner. Rules for mapping are the following:

- Generic model name has to be embedded in specific release model name
- MAP keyword (w/o EXPLICIT) maps generic waste form to a specific release model
- MAP EXPLICIT keyword maps a specific waste form to a specific release model
- Wastename may have generic waste name the same as a generic model name
- Wastename can be mapped to specific model name (which itself refers to generic model)

#### **5.4.11 MODELS Keyword for VADER**

The MODELS keyword specifies which analytical release models to run and also specifies the release coefficient values. The syntax for the MODELS keyword is different for each of the models. The following is this keyword's syntax by model type (the model name must be the first entry after MODELS):

```
MODELS [SOIL] [Kd=Nkd] [Cs=Ncs] {Qw=Nqw} {Tw=Ntw} {B=Nb} {A=Na} {h=Nh}  
  [StartRel=NNNN StopRel=NNNN] {NOCOLLAPSE} {COMMENT="quote"}  
  
MODELS [CMNT] [Dc=Ndc] [Ac=Nac] [Vc=Nva] [StartRel=NNNN StopRel=NNNN]  
  {NOCOLLAPSE} {COMMENT="quote"}  
  
MODELS [CSOL] [Cs=Ncs] {Qw=Nqw} [A=Na] [StartRel=NNNN StopRel=NNNN]  
  {NOCOLLAPSE} {COMMENT="quote"}  
  
MODELS [CAKE] [Cs=Nds] {Qw=Nqw} [R=Nr] {A=Na} [StartRel=NNNN StopRel=NNNN]  
  {NOCOLLAPSE} {COMMENT="quote"}  
  
MODELS [C14] [T1=Nt1] [T2=Nt2] [T3=Nt3] [D1=Nd1] [D2=Nd2] [StartRel=NNNN  
  StopRel=NNNN] {NOCOLLAPSE} {COMMENT="quote"}
```

```
MODELS [RXCOMP] [Cs=NCs] [t=Nt] [Cr=Ncr] [A = Na] [M0 = Nm] [StartRel=NNNN
  StopRel=NNNN] {NOCOLLAPSE} {COMMENT="quote"}
```

```
MODELS [CORE] [Fr=Nfr] [StartRel=NNNN StopRel=NNNN] {NOCOLLAPSE}
  {COMMENT="quote"}
```

```
MODELS [LIQUID] [F=Nf] [StartRel=NNNN StopRel=NNNN] {NOCOLLAPSE}
  {COMMENT="quote"}
```

```
MODELS [RIVER] [F=Nf] [StartRel=NNNN StopRel=NNNN] {NOCOLLAPSE}
  {COMMENT="quote"}
```

```
MODELS [GLASS] [Fg=Ng] [StartRel=NNNN StopRel=NNNN] {NOCOLLAPSE}
  {COMMENT="quote"}
```

```
MODELS [GLASBM] [Qw=Nqw] [StartRel=NNNN StopRel=NNNN] {NOCOLLAPSE}
  {COMMENT="quote"}
```

For a single VADER run, release models are made unique through unique model names. Each release model name must embed the generic release model, and the keyword line must contain all the parameters associated with the generic release model (in some cases there are defaults). Example: The analyst could declare SOILXX as a specific release model which applies unique coefficients to the generic SOIL release model. In this way, specific release models can be applied to specific portions of the waste inventory which are labeled as “soilxx”. The MODELS keyword and the MAP keyword are used together to assign specific waste forms to a specific release model. In practice, the ESP 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.

Each release model requires its own MODELS line with its own unique set of coefficient values. The coefficients and expected physical units for each model are itemized in Table 5.

**Table 5.8** Modifiers for the MODELS Keyword in VADER

Release Model	Coefficients <sup>(a)</sup>	Modifier Description	Units
SOIL For soil wastes (With variants)	Kd=nnn Cs=nnn  Qw=nnn <sup>(b)</sup>  Tw=nnn B=nnn A=nnn h=nnn	<b>Solubility/desorption controlled release</b> 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.	cm <sup>3</sup> /g Ci/cm <sup>3</sup> kg/cm <sup>3</sup> cm/yr <sup>(b)</sup> unitless g/cm <sup>3</sup> cm <sup>2</sup> cm
CMNT For cement wastes with variants	Dc=nnn Ac=nnn Vc=nnn	<b>Diffusion controlled release</b> 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 <sup>2</sup> /yr cm <sup>2</sup> cm <sup>3</sup>

Release Model	Coefficients <sup>(a)</sup>	Modifier Description	Units
CSOL For soil wastes	Cs=nnn Cs=nnn Qw=nnn <sup>(b)</sup>  A=nnn	<b>Solubility controlled release</b> Aqueous solubility of the analyte (radionuclide). Aqueous solubility of the analyte (nonradionuclide). Darcy flux density of water flowing through source zone (recharge or infiltration rate). Effective cross-sectional area of contaminant source zone. Note: Cs (solubility) units must reflect the units of inventory.	Ci/cm <sup>3</sup> kg/cm <sup>3</sup> cm/yr <sup>(b)</sup>  cm <sup>2</sup>
CAKE For tank wastes	Cs=nnn Qw=nnn <sup>(b)</sup> A=nnn R=nnn	<b>Release congruent with dissolution of matrix</b> Aqueous solubility of major structural component of saltcake (assumed to be NaOH) Darcy flux density of water flowing through source zone (recharge) Effective cross-sectional area of cake waste form Density of structural matrix, for converting waste volume from the inventory file into inventory mass units. Default 1.5 g/cm <sup>3</sup> Note: the mass of structural matrix (saltcake) 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 <sup>3</sup> , and density R is used to convert from cubic meter to grams, so enter Cs in g/cm <sup>3</sup> .	g/cm <sup>3</sup> cm/yr <sup>(b)</sup> cm <sup>2</sup> g/cm <sup>3</sup>
CORE For reactor graphite core	Fr=nnn	<b>Release based on fraction of Initial Inventory</b> Initial fractional release rate for contaminant from graphite surplus reactor core waste form. Fr ranges from 0 to 1.	1/yr
C14 For Carbon-14 releases from reactor core (CORE waste)	T1=nnn T2=nnn T3=nnn D1=nnn D2=nnn	<b>Empirical model based on temperature change</b> Initial temperature of core (e.,g., 200 °C) Stable cocooning temperature (e. g., 22 °C) Stable burial temperature (e. g., 16 °C) Time in years between T1 and T2 Time in years between T2 and T3 Temperature-dependent carbon-14 version of graphite surplus reactor core release model. Built-in model assumes linear temperature decline between T1 and T2 over the D1 period. Note: If the INVENTORY file specifies “core” wastes as having quantities of C14 analyte, be sure to specify the MAP keyword to map the “core” wastes to the C14 release model, as in MAP Waste “CORE” Model “C14”.	°C °C °C years years

Release Model	Coefficients <sup>(a)</sup>	Modifier Description	Units
RXCOMP <sup>(d)</sup>  For reactor compartments (rxcomp wastes)	Cs=nnn t=nn Cr=nnn A=nnn M <sub>0</sub> =nnn	<p><b>Release dependent on pressure vessel steel corrosion rate</b></p> Hull steel corrosion rate (if not entered must enter STARTREL) Hull steel thickness (if not entered must enter STARTREL) Reactor vessel corrosion rate (assumes constant vessel thickness) Reactor vessel surface area for one reactor unit Reactor vessel steel mass for one reactor unit	cm/yr cm kg/dm <sup>2</sup> /yr dm <sup>2</sup> kg
GLASS For glass wastes	Fg=nnn	Original glass model uses empirically derived release fraction.	unitless

<b>Release Model</b>	<b>Coefficients<sup>(a)</sup></b>	<b>Modifier Description</b>	<b>Units</b>
GLASBM For glass wastes	Qw=nnn (or use recharge table)	Empirical model based on Bacon-McGrail glass model (STORM) for Immobilized Low Activity (ILAW) wastes. Developed and considered valid over the range 0.1 through 50 mm/year infiltration rate, but VADER allows extrapolated predictions over 0.1 through 140 mm.yr. Developed for LAWABP1 glass. May or may not apply to other glass formulations or waste forms.	cm/yr
LIQUID For liquid wastes	F=nnn	<b>Pass-through model for release to vadose zone</b> 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	<b>Pass-through model for release to “the river”</b> 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] <sup>(c,d)</sup>	Calendar year A. D. 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 0).	yr
STOPREL	[StopYear] <sup>(c,d)</sup>	Calendar 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 0).	yr
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
<p>(a) Coefficient definitions are taken from Table D.1 of Appendix D in Kincaid et al. (1998).</p> <p>(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 the RECHARGE keywords. The SOIL, CSOL, GLASBM and CAKE release models require recharge rate Qw in cm/yr. A global table of recharge values expressing several values for Qw over different time periods applying to all models without specific Qw entries can be entered via INFILTRATION keywords.</p> <p>(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.</p> <p>(d) Any STARTREL entry for the RXCOMP model is superceded by the the RXCOMP model calculation of STARTREL as calculated with parameters t and Cs and the deposit year.</p>			

VADER provides for specification of time periods over which the release mechanism is activated with the STARTREL and STOPREL parameters. They default to the simulation time periods. If STARTREL is not set to a specific calendar year, release is considered to begin the year the waste is deposited in inventory. The RXCOMP model has a provision for calculating a STARTREL date from input parameters. Release of contaminant stops when the waste inventory is depleted.

Although inventory and remediation actions may take place at dates outside the range specified by STARTREL and STOPREL, 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 explicitly through the STARTREL and STOPREL dates in each MODELS keyword line. If these modifiers are not present, the dates default to the entries in the PERIOD keyword. In addition, certain models such as the RXCOMP model calculate STARTREL. Calculations for release from containment take place only between these dates.

The following are some example invalid MODELS keyword entries (using default start and stop dates):

```
MODELS SOIL Kd=.0001 Cs=.0003 Qw=25 B=2.2 Tw=.05 COMMENT="Bottomlands Model"  
MODELS CORE Fr=.0045 COMMENT="Graphite Reactor Block"
```

The  $Q_w$  entry applies to the SOIL model entry only. Because the A and H parameters are not provided, this entry would be declared erroneous by VADER. The CORE entry would be declared complete.

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 specifies that the release mechanism was halted in 2000 A. D. after 44 years. The next CAKE model specifies 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 to be entered, the analyst uses the INFILTRATION keyword (see Section 5.4.9) and then skips entry of  $Q_w$  in the MODELS keyword for models that require  $Q_w$ . This table is global for all models that do not have an explicit  $Q_w$  modifier entered. If both  $Q_w$  for a specific model parameter and INFILTRATION keywords are present, the MODELS parameter supercedes the infiltration table for that model only. This means that infiltration rates that vary over the time period have to be entered via the INFILTRATION keywords, and that specific models that enter  $Q_w$  explicitly can have only a constant infiltration rate over the entire release period.

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. The default release models for these wastes would be the SOIL release model.

VADER has a provision to 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 different 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. The following three keywords show an example with



different solubilities and different soil moisture content for each SOIL model to invoke the SOIL release model with different parameters for each inventory:

```
MODELS SOILA Kd=.0001 Cs=.0003 Qw=25 B=2.2 Tw=.050 COMMENT="Kalahari Sands"  
MODELS SOILB Kd=.002 Cs=0.003 Qw=25 B=2.2 Tw=0.15 COMMENT="USN Garden Soil"  
MODELS SOIL Kd=.002 Cs=0.050 Qw=25 B=2.2 Tw=0.15 COMMENT="Mississippi Mud"
```

With these variant soil release model specifications, VADER will apply the appropriate model to each inventory SOILA, SOILB, and SOIL 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.

#### **5.4.12 NOCOMPUTE Keyword for VADER**

The optional NOCOMPUTE keyword is provided to allow VADER to test a set of keywords without attempting to execute a complete run. When the NOCOMPUTE keyword is present, VADER reads the keyword file and records their interpretations and error messages in the vader.debug file. VADER then terminates without writing any output files. In particular, it does not write a vader.done nor a vader.fail file. The following is this keyword's syntax:

```
NOCOMPUTE
```

#### **5.4.13 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 from which to read input files and to which to write output files. Only one PATH keyword can be entered. The default setting is to place the output files in the VADER working directory. The following is this keyword's syntax:

```
PATH ["pathname"]
```

The following is an example PATH keyword:

```
PATH "C:\SAC\PayRun\VADER\Site001\Sr90\Real001"
```

#### **5.4.14 PERIOD Keyword for VADER**

The PERIOD keyword identifies the start and stop years (calendar years A.D.) for the release simulation. Only one PERIOD keyword line can be entered. The following is this keyword's syntax:

```
PERIOD [START=Year1] [STOP=Year2 ] [CLOSURE=Year3]
```

The start of the simulation period is provided in the numerical value (Year1) associated with the START modifier. The end of the simulation period is provided in the numerical value (Year2) associated with the modifier STOP. Year2 must be greater than Year1. Start and stop years should be entered as whole numbers. The year of site closure is provided in the numerical value associated with the CLOSURE

modifier. Inventory deposits and remediation transfers are checked to determine there are no actions beyond the site closure year; a warning is issued if an action is found but processing continues.

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
```

If the PERIOD keyword is absent, START, STOP, and CLOSURE are set to 1944, 3050, and 2050 respectively.

#### **5.4.15 REALIZATION Keyword for VADER**

The REALIZATION keyword identifies the realization to be simulated. One or two values may be entered. The following is this keyword's syntax:

```
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. N2, when present, must be greater than or equal to N1. An example keyword for the 99<sup>th</sup> 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 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.

#### **5.4.16 REMEDIAT Keyword for VADER**

The optional REMEDIAT keyword has two functions. First, it indicates the input files that define incoming (imported) remediation quantities in specific waste forms from previously computed sites. Second, it provides information for calculating outgoing (exported) remediation quantities in specific waste forms, specifying sites to be computed later, and writing an output remediation file (vader.rem) in the working directory. VADER always remediates waste from 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 following is this keyword's syntax for imported remediations:

```
REMEDIAT [FROM="quote1"] [TO="quote2"]
```

The following is this keyword's syntax for exported remediations:

```
REMEDIAT [YEAR=N1] [FROM="quote1"] [TO="quote2"] {SOIL=N1} {CAKE=N2}
        {CMNT=N3} {CORE=N4} {GLASS=N5} {RXCOMP=N6}
```

The import form of the REMEDIAT keyword only needs enough information to locate the previously computed REMEDIAT and vader.rem files in the import site directories (all other keyword modifiers like year and fraction are ignored for imports). Because material might be imported from several sites during a given year, one or more import sites can be declared; each has its own REMEDIAT keyword line. The input files contain one or more records that provide transfer date, analyte, quantity in Ci or kg depending on the analyte, and waste form for a given transfer to the working site in a given year. The radioactive quantities in imported remediation transfers are assumed to be already decayed to the stated year. The modifiers for the REMEDIAT keyword that apply to imported wastes are described in Table 5.9.

**Table 5.9** REMEDIAT Keyword Modifiers for Imports to the Working Site in VADER

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 "Trollheim". The following REMEDIAT keyword identifies that waste will be imported from the separate previously computed site named *Dump*.

```
REMEDIAT FROM="Dump" TO="Trollheim"
```

VADER constructs path names to point to two files, one written by VADER and one written by STOMP in previously computed site directories. VADER attempts to read both files. (See Sections 1.2 and 1.4 for a brief discussion of the directory structure). 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 informative message and proceeds; it is not a fatal error to find no incoming "remediate" nor "vader.rem" file file.

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 times in calendar years; STOMP defines times in elapsed seconds from midnight preceeding January 1 of the base year (for SAC runs, 00:00:00 January 1, 1944).

The export form of the REMEDIAT keyword line needs to specify the export site and export year and the information necessary to calculate the quantities of wastes to transfer each year. Each REMEDIAT keyword identifies a separate transfer from the working site inventory to the export site during a given year. Each keyword 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 per year per export site is allowed. 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 previous remedial actions by the export fraction, before applying this year's release. 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 for Exports from the Working Site in VADER

<b>Modifier</b>	<b>Description</b>
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.
CAKE	The numerical value associated with the CAKE modifier identifies the fraction of remaining saltcake waste to export. Valid values are in the range 0 to 1.
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.
CORE	The numerical value associated with the CORE modifier identifies the fraction of the remaining reactor core waste to export. Valid values are in the range 0 to 1.
GLASS	The numerical value associated with the GLASS modifier identifies the fraction of the remaining reactor core waste to export. Valid values are in the range 0 to 1.
RXCOMP	The numerical value associated with the RXCOMP modifier identifies the fraction of the remaining reactor compartment 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 picked up by subsequent 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. A SAC run with remediation transfers relies on the system supplying REMEDIATE keywords in all the vader.key files to specify all the sites involved in remediation transfers.

The following REMEDIATE keyword lines 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 Memphis 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 nor releases 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 at Memphis.

```
REMEDiate YEAR=2022 FROM="Memfis" TO="Thebes" SOIL=0.45
REMEDiate YEAR=2022 FROM="Memfis" TO="Kairo" SOIL=0.55
```

The VADER run at working site Thebes will have been told to look for a vader.rem file in the Memphis working site directory (for this realization and analyte) by supplying a REMEDIATION (incoming) keyword. 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.

To transfer all the inventory of a given waste form over multiple years to the inventory at Thebes, the REMEDIATE keywords would look like this:

```
REMEDiate YEAR=2022 FROM="Memfis" TO="Thebes" SOIL=0.45
REMEDiate YEAR=2023 FROM="Memfis" TO="Thebes" SOIL=0.55
REMEDiate YEAR=2024 FROM="Memfis" TO="Thebes" SOIL=1.00
```

This tells VADER to export 45% of the SOIL inventory in 2022, 55% of the remaining balance in 2023, and the entire balance remaining in 2024.

Remediation files read by VADER typically were built by previous VADER or STOMP runs, and are intended to be read by subsequent VADER runs. VADER never reads VADER-built nor STOMP-built remediation files residing in the working directory.

VADER will import waste material from preceding sites only if the REMEDIATE keyword is present in vader.key. Moreover, even if quantities of contaminants are imported, if the MODELS keyword is not present, there will be no releases at the working site.

#### **5.4.17 SITE Keyword for VADER**

The SITE keyword is used to specify the working aggregate release site to process. Only one working release site is defined for each run; if more than one SITE 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 this keyword's syntax:

```
SITE [ ID="quote1" ] { TITLE="quote2" }
```

The quote string associated with the ID modifier contains the Site ID. The ID is limited to 15 characters and must be unique. It is used to associate other data with a specific site. The quote string associated with the optional TITLE modifier contains a descriptive title for the aggregation site that is used labeling purposes in output files. The title is limited to 72 characters. The following is an example SITE keyword:

```
SITE ID="600-148" TITLE="Env. Restoration Disposal Facility (ERDF)"
```

#### **5.4.18 STEPS Keyword for VADER**

The optional STEPS keyword provides for adaptive time-stepping for output files that supercedes the default annual time steps. Its action on the output "vader.table" and STOMP "input" files is to aggregate inventories remaining and quantities released from adjacent years into one record, starting at a stated year. In this way the vader.table and input files are considerably shortened especially in cases with long time periods. The action of this keyword is designed to preserve resolution in areas where variations in quantities released will be most pronounced. The following is this keyword's syntax:

```
STEPS {START=NNNN} {NSTEP=NS}
```

The numerical value associated with the START modifier identifies the calendar year the time-stepping is turned on. Time steps will be annual before this date. The default value is one calendar year past the closure year identified by the in PERIOD keyword. The numerical value associated with the NSETP modifier is the number of adjacent years to pool or aggregate into one record in the vader.table file and the STOMP input file. If NSTEP is specified, every NS records are pooled into one record. Release quantities are decayed to the midpoint of each interval. The default setting is NSTEP=10.

The following keyword entry outputs annual records until the year 3000. After that year, every 100 years will be pooled, so that years 3001 through 3100 are aggregated into one record and so forth:

```
STEPS START=3001 NSTEPS=100
```

The STEPS option is applied to annual releases after the CUTOFF option is applied.

#### **5.4.19 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. 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 quote string. The following is an example TITLE keyword:

```
TITLE "Scenario XYZ: Release Sr90 at a Waste Site 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 5.6 provides an overview of all files used by VADER.

**Table 5.6** Overview of Files Used by VADER

<b>File</b>	<b>I/O</b>	<b>Contents</b>	<b>Notes</b>
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.
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-esp 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.
vader.air	O	Time series of direct releases to the atmosphere	To be processed by
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 ESP that VADER is running.
vader.done	O	ASCII file, existence polled by ESP and STOMP. Contains special run information for SAC-ESD to direct further processing.	Signals 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 ESP that the VADER run failed.

File	I/O	Contents	Notes
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

The VADER keywords are described in Section 5.4. Table 5.7 provides an example file. This file is immediately opened and parsed for command information when VADER is invoked.

**Table 5.7** 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 processes 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.

An example inventory results file is provided in Table 5.8. In the SAC context this file is generated by the INVENTORY code (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 5.8** Example Inventory Results File

<pre> 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<sup>3</sup> 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<sup>3</sup> liquid, 100 m<sup>3</sup> 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<sup>3</sup> 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, </pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

"Cs137", "Ci", 40., 55.5,  
"CC14", "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 5.9.

**Table 5.9** WASTE Form to Release Model Mapping in VADER

Waste Form (INVENTORY)	Canonical Waste Form (VADER)	Release Model Name (vader.key)
SOIL <sub>xx</sub> <sup>(a)</sup>	SOIL	SOIL
GLASS	GLASS	GLASS
CEMENT	CMNT	CMNT
CMNT <sub>xx</sub> <sup>(a)</sup>	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.		

Two mapping operations are illustrated in Table 5.9. Variant release model names (such as CEMENT, CMNT, and CMNT<sub>xx</sub>) 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 5.4.11). 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 both 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 5.10 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 5.4.16):

```
REMEDiate YEAR=1999 FROM="116-B-1" TO="600-148"
```

**Table 5.10** 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 5.11 shows an example VADER-built remediation file (vader.rem) at working site 116-B-1 to export strontium-90 to site 600-148.

**Table 5.11** 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 5.4.16) allows exports from all waste forms in the inventory by specifying the fraction of the current cumulative inventory. The file in Table 5.11 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 5.11 are explained as follows. At 1999, let the Sr-90 net cumulative inventory in soil at the working site 116-B-1 be 52.11 Ci. In 1999, 45% (23.45 Ci) of this is sent to site 600-148, leaving the inventory at 28.66 Ci. 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 final transfer accounts for 100% of the core inventory.

#### 5.5.1.4 BYPASS file

The BYPASS option is used for scenarios in which the release profile – quantity released per year – is calculated “off-line” and supplied to VADER through a separate file. The BYPASS keyword in vader.key supplies the file name. This section describes the expected format of a BYPASS file. VADER can accept any number of BYPASS files in one run. Each bypass file is given a unique file name and is referenced by a unique BYPASS keyword line in vader.key. The analyst who prepares this file to assure that volumes, inventories, and release quantities are internally consistent, as VADER acts as a pass\_through agent with no internal processing.

The BYPASS file put specifications and the actual quantities released in one place – the data file. Routing can be to the vadose zone either through the aggregate solids pool or through the liquids pathway. An example bypass file is provided in Table 5.12.

**Table 5.12** Example BYPASS file for VADER

SITE	"ERDF"		
ANALYTE	"Sr90"		
REALIZATION	66 [note unquoted]		
QTYUNITS	"C"		
VOLUNITS	"m^3"		
ROUTING	"LIQUIDS"		
WASTE	"Termites"		
HANDLING	"INSTANT"		
DATA			
Year	Volume	Inventory	Release [note unquoted]
2000	100	123	13
2019	0	0	3
2020	50	175	0
2040	200	45	7
2048	1	1	1
2100	1	0	1
END			

The BYPASS file is read as found in the working directory. Entries are treated as pertaining with given years, so that years with no entries imply assuming that the quantities and volumes are zero. The waste form is called “Termites”. The released analytes are routed to the vadose zone via the liquids pathway, which means that VADER will store these quantities in the liquids section of the STOMP template file “input” along with the volume. The site, analyte, realization, and units are be matched against the VADER run units from the vader.key file to verify that they are in agreement.

The BYPASS file (Table 5.13) has two sections: a header section and a release data table section. The header section supplies identification and specifications needed by VADER to route the release profile correctly.

**Table 5.13** BYPASS file header records specifications for VADER

Field	Keyword	Description
Site	SITE	Release site ( "ERDF", for example)
Analyte	ANALYTE	SAC chemical symbol such as "Cs137"
Realization	REALIZAT	Integer realization number
Release Units	QTYUNITS	"Ci" or "kg" (also applies to inventory remaining)
Volume Units	VOLUNITS	"m^3" or "l" (cubic metres or litres)
Routing	ROUTETO	"SOLID" or "LIQUID"
Waste Name	WASTE	Waste form name
Handling	HANDLING	"INSTANT" or "STEP" or "INTERP"
Note: All alphanumeric items are in quotes.		

Most of these entries are self-explanatory. The HANDLING keyword describes how VADER will handle numerically handle the release profiles supplied in the data table.

If the HANDLING option is INSTANT, data records represent instantaneous deposits and releases, with no interpolations or step functions. If the HANDLING option is STEP, data records represent a step-function release, with constant release starting at given year, and continuing year-by-year until next record entry. If the HANDLING option is INTERP, data records represent a curvilinear function with year-quantity pairs. VADER will use linear interpolation for intermediate years. When STEP or INTERP are specified, volume, release, and inventory are interpolated using the same method.

The data section of the BYPASS file is described in Table 5.14. It starts with the DATA keyword and concludes with the END keyword. A column heading line must immediately follow the DATA keyword. Data records for each year is represented by a record with two, three, or four columns as identified in the columns headings line. Each year's quantity is assumed decayed to the stated year. The following rules apply:

- Columns may appear in any order
- Year and Release are mandatory
- Volume is mandatory for liquid releases
- Inventory is mandatory for incoming deposits.

**Table 5.14** BYPASS file data records description for VADER

Field or Column	Column Name	Description and storage array in VADER
Year	YEAR	Year of deposit
Release Quantity	RELEASE	Quantity (Ci or kg) released to the vadoze zone
Inventory Quantity	INVENTORY	Quantity (Ci or kg) deposited in inventory
Volume Quantity	VOLUME	Volume (m <sup>3</sup> or L) of waste released to vadose zone
Note: Units are as specified in the header records.		

### 5.5.1.5 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 provided as rates (amount per second). Table 5.15 provides an example for excerpted records for this file after modification by VADER.

**Table 5.15** Excerpted Records from an input-esp File Modified by VADER

<pre>#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,</pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

```
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 file is read by default in VADER, but it can be turned off by the following keyword entry in the vader.key file: “FILES STOMP\_TEMPLATE NOCOMPUTE” (see Section 5.4.8). 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 format shown in Table 5.16 and described in Table 5.17. This example file contains excerpted records from a file showing strontium-90 inventory deposits, remediation deposits, releases, and quantities remaining remaining for several waste forms at a hypothetical site, Memphis. This file starts with four header records and is followed by waste form information. The inventory and release models correspond to Realization 01. Waste volumes are in m<sup>3</sup> and analyte quantities are in Curies. Each waste form's deposit and release history is represented by 107 data records over the period 1944 through 2050 A. D.

**Table 5.16** Excerpted Records from a vader.table File

Title: "Remediating Pyramids for the Environmental Dynasty"										
Site: "Memphis"										
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

**Table 5.17** Field Descriptions for the vader.table file

Header Records	
Column	Description
Title	Description of run from the TITLE keyword in the keyword file
Site	Release site from the SITE keyword
Analyte	Analyte name from the ANALYTE keyword
Realization	Realization number from the REALIZATION keyword
Waste	Waste form name from the INVENTORY file
Coeffs	Release model coefficients from the MODELS keyword as mapped to the Waste form
"107"	This is the number of data records following representing the release history for this run
Release Data Records (all quantity units are assumed to be Ci or kg, the units for the input quantities)	
Column	Description
Year	Year during which quantities are deposited, transferred in our out, and released. All actions are assumed to occur January 1 of the stated year.
Qw (cm)	Infiltration quantity in centimeters for given year.
QTY_IN	Quantity deposited into the inventory during a given year
RMD_IN	Quantity imported or transferred in from a previous site during the given year
RMD_OUT	Quantity exported or transferred to a subsequent site during the given year
Release	Quantity released to the environment (vadose zone) during given year
CumRel	Accumulated quantity released as of the given year
CumQty	Net accumulated quantity remaining in inventory as of the given year after accounting for deposits, remediations, and releases.

### 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 5.4.11). 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 cubic meters, and quantity units are Curies for

radioactive analytes and kilograms for nonradioactive analytes - these are units as input from the Inventory Module. Table 5.18 provides excerpted records from a vader.river file.

**Table 5.18** Excerpted Records from a vader.river File

FILE_NAME:	VADER.river		
Run_Date:	08.06.2004	16:42:45	
RUN_TITLE:	VADER template keyword file for site: 1908-NE		
VADER 1.07 Version Date 14 Jul 2			
SITE:	1908-NE		
ANALYTE:	Sr90	m <sup>3</sup>	Ci
Realization:	1		
WasteForm:	RIVER		
NREC:	57	Volume	Release
1944	0.000000E+00	0.000000E+00	0.000000E+00
1945	0.000000E+00	0.000000E+00	0.000000E+00
1946	0.000000E+00	0.000000E+00	0.000000E+00
1947	0.000000E+00	0.000000E+00	0.000000E+00
...			
1996	0.230990E+07	0.116398E+00	
1997	0.230990E+07	0.126100E+00	
1998	0.230990E+07	0.281323E+00	
1999	0.230990E+07	0.678972E-01	
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 . Fluxes are written in mass (kg) or activity (Ci) units per second, and volumes are written in m<sup>3</sup>/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 5.15 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 5.19 provides an example VADER.log file.

**Table 5.19** 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      AAAAAA D      D E      R      R
V      V      A      A      D      DD E      R      R
V      V      A      A      DDDDD EEEEEEE R      R

                VADER                Version 1.07
                Last Modified on 14 Jul 2004

-----
                Vadose Zone Environmental Release Module
-----

                Developed By Battelle Memorial Institute
                Pacific Northwest National Laboratories
                Richland, Washington

Current Run ID = 20040806164244  User Name = SAC GW/VZ
System Date = 08-06-2004  System Time = 16:42:44.939

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: _____
lMAIN-VADER VADER Keyword File is vader.key
lECHO_CHECK OF VADER INPUT KEYWORD FILE SPECIFICATIONS
-----
TITLE  VADER template keyword file for site: 1908-NE
SITE   SITE=1908-NE          Outfall
ANALYTE ANALYTE= Sr90
Units=      Ci
Type=NR
HalfLife= 0.287800E+02  Yr
CONTROL Waste option is ALL
CUTOFF  CutOff Analyte below  0.100000E-12
        CutOff Volume below  0.100000E-12
DELTAT  1.000 yr
DEBUG

FILE          Show filenames for files to attempt to open
Unit Type Open File-name
  1  A  R  /home/ANALYSIS4/CA1_median/vadose/1908-NE/Sr90/1/vader.key
  2  A  R  /home/ANALYSIS4/CA1_median/vadose/1908-NE/Sr90/1/inv.dat
  3  A  R  /home/ANALYSIS4/CA1_median/vadose/1908-NE/Sr90/1/remediate
  7  A  C  /home/ANALYSIS4/CA1_median/vadose/1908-NE/Sr90/1/input-esp
 10  A  C  /home/ANALYSIS4/CA1_median/vadose/1908-NE/Sr90/1/vader.table
  9  A  C  /home/ANALYSIS4/CA1_median/vadose/1908-NE/Sr90/1/vader.log
 11  A  C  /home/ANALYSIS4/CA1_median/vadose/1908-NE/Sr90/1/vader.done

No BYPASS Files Entered
MAP      Number Explicit Waste-to-Model mappings:  0
MODELS   Number Models = 1

```

*User Instructions for the Systems Assessment Capability, Rev. 0, Computer Codes  
Volume 1: Inventory, Release, and Transport Modules*

---

```
Specific Algorithm
RIVER RIVER Collaps= T NonDepl= F 1944 12050 F = 0.10000E+01 Unitless
NOCOMPUT ComputeReleases = T Proceed to compute releases
PATHNAME PATHNAME=/home/ANALYSIS4/CA1_median/vadose/1908-NE/Sr90/1/
PERIOD Start Year= 1944
      Closure Year= 2070
      Stop Year= 12050
REALIZAT Realization Number is 1 Max Realization Number is 1
INFILTRA Number INFILT lines is 2. Units are cm/yr
Start End Qw(cm)
 1944 1968 0.1000
 1969 12050 0.1000
IF INFILTRATION table present, it overrides any entry of Qw in MODELS keywords
REMEDIAT Number Remediation Records is 0
STEPS Post-hoc adaptive time-steps turned on at year 2070
Equi-interval Time-steps Nstep= 10
EXIT VADER_KEY_SHOW
ELAPSE Finishing preliminaries
```

```
Date: 08-06-2004
Time: 16:42:44.947
```

```
ENTERING READ_INV
```

```
-----
DISPLAY_INV
```

```
INVENTORY REPORT
```

```
(w/o reference to Remediations)
```

```
Inventory File Name: /home/ANALYSIS4/CA1_median/vadose/1908-NE/Sr90/1/inv.dat
```

```
Analyte: Sr90
Site: 1908-NE
Realization: 1
Number Waste Forms Present 1 (Non-zero Quantities)
```

```
Waste Form RIVER
Rel. Model RIVER
 1972 m^3 0.48999E+09
 1972 Ci 0.92147
...
 1999 m^3 0.23099E+07
 1999 Ci 0.67897E-01
```

```
Waste Form RIVER
Rel. Model RIVER
```

```
Tot V 0.74731E+10
Tot Q 48.211
```

```
Total Waste Inventories and Volumes by Release Model
```

```
Model Quantity Volume
-----
RIVER 48.2106 Ci 0.747314E+10 m^3
DISPLAY_INV End Inventory Listing
-----
```

```
ENTERING CHECK_WASTE_MODELS
```

```
----Inventory---- ----Release Model----
```

```
Waste Associated Algorithm vader.key
Form Rel. Model (ModelNam) (MODELS)
RIVER RIVER RIVER RIVER

Number Waste Forms in Inventory and Remedys = 1
Number Waste Forms mapped to a Release Model= 1
Number Release Models Specified in MODELS = 1
Number Waste Forms with sum(waste)<CutOffQ = 0
CHECK_WASTE_MODELS PROCEED: RELEASE MODEL SPECIFIED FOR ALL WASTES>CUTOFF
ALL IMPORT WASTE FORMS HAVE RELEASE MODELS DECLARED
-----

ELAPSE Begin the Calculations Loop

Date: 08-06-2004
Time: 16:42:45.085
-----

Entering RIVER RIVER
Processing Release Model RIVER at index i= 1 Comment:
Coeff Value Units
Start Release during 1944
Stop Release during 12050
F 0.10000E+01 Unitless
HL 0.28780E+02 yr

INITIALIZER Initializing RIVER Last Inventory or Remediation-In Year 1999
EXIT SUBROUTINE RIVER Ierr= 0
No Outgoing Remediations Specified in Keyword File
No REMEDIATIONS file written to working directory
STOMP Template File INPUT CREATED
Matches Site, Analyte, and Realization Number
Copied 131 lines from "input-esp" file
STOMP input template pathnames:/home/ANALYSIS4/CA1_median/vadose/1908-
NE/Sr90/1/input-esp
STOMP final template pathnames:/home/ANALYSIS4/CA1_median/vadose/1908-NE/Sr90/1/input
Write Vader.done file to /home/ANALYSIS4/CA1_median/vadose/1908-NE/Sr90/1/vader.done
11 0
NOTICE: All Waste Forms in INVENTORY are associated with Release Models
TOTAL CPU TIME: 0.189999998

ELAPSE Finish VADER Run

Date: 08-06-2004
Time: 16:42:45.178
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.

### 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 ESP that VADER has finished and the STOMP source term is ready for processing. The vader.done file also signals 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. More comprehensive information is provided in the vader.table file. Clock time finished and CPU time to execute are also provided. Table 5.20 provides an example vader.done file.

**Table 5.20** Example vader.done File

```
VADER.done /home/ANALYSIS4/CA1_median/vadose/1908-NE/Sr90/1/vader.done  VADER
1.07 Version Date 14 Jul 2004
1908-NE
Sr90
1
Waste Quantity Released by Each Release Model
Number Models= 1          CumRel(StopYr) MaxAnnualRel
RIVER          1          1          27  0.0000          8.4396

Verify that all INVENTORY Wastes are referenced in vader.key
INVENTORY      RELEASE.MODEL
              RIVER          RIVER
NOTICE: All Waste Forms in INVENTORY are associated with Release Models

ELAPSE  Finish VADER Run

      Date: 08-06-2004
      Time: 16:42:45.177
TOTAL CPU TIME:  0.189999998
Error Code Ierr =  0
MAIN-VADER Successful VADER run
```

## 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 currently the main 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 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 has been enhanced since SAC Rev. 0 to provide 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.

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 ‘release’ files) 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. The use of this mode allows the representation of vadose zone moderating effects on liquid discharges to ground in the groundwater flow model.

### **6.2 Algorithms and Assumptions**

This section provides a brief description of algorithms and major assumptions for VZDROP.

#### **6.2.1 Nearest CFEST Node Determination for Vadose Zone Sites**

Each vadose zone site has from three to eight pairs of coordinates that define a polygon that represents the vadose zone areal extent in state plane coordinates. 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	08/06/2004 - 13:34:30
#SAC Case ID	: 2004 Composite Analysis 10,000-year (Median Inputs) Assessment
#SAC Template	: vadose/241-C-103/Tc99/template_stomp1.key
#SAC Site ID	: 241-C-103
#SAC Site Points	: 4
#SAC Site Eastings (m)	: 5.7519431E+05, 5.7521456E+05, 5.7521456E+05, 5.7519431E+05,
#SAC Site Northings (m)	: 1.3653737E+05, 1.3653737E+05, 1.3655762E+05, 1.3655762E+05,
#SAC Site Area (m <sup>2</sup> )	: 4.1008594E+02
#SAC Analyte	: Tc99
#SAC Realization	: 1

VZDROP extracts the location information from these file headers and computes the area (in square meters) represented by the polygon with the given points as vertices (assuming that the polygon is closed by a line from the last provided coordinate pair to the first provided coordinate pair).

Next, for each vadose zone release area polygon provided, VZDROP calculates the area of overlap between that vadose zone site area and the “catchment” of each CFEST finite element node (“catchment” here refers to the portion of each finite element in a CFEST finite element grid that is closest to the node in question). The same POLYINT routines developed for GWDROP are used in VZDROP to calculate these fractional areas. The resulting areas of overlap are used to develop fractions of each vadose zone area to be assigned to each CFEST node. If any portion, or all, of a vadose zone area does not occur within the CFEST model grid, then all of the missing fraction is automatically assigned to the nearest CFEST grid node that is in the domain.

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 are not allowed to be specified as nodal sources in CFEST; such nodes are automatically detected by VZDROP and removed from the list of possible nodes to receive sources. 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 state plane coordinates (0,0) before assignment of sources to nodes. This step ensures that these nodes are well off the Hanford Site and will not be selected as a source node for 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.

## **6.2.2 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.

VZDROP calculates arrays of fractional areas of each vadose zone polygonal area assigned to each CFEST node that is not a held-head or sink node. The assignment always uses the surface (topmost) node in the three-dimensional CFEST domain (i.e., assignment of sources to deeper layers in the CFEST domain is not supported by VZDROP). This spatial distribution scheme provides for geographically based distribution of fluxes for large vadose zone release sites to multiple groundwater nodes.

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 base time for all SAC codes (midnight preceding January 1, 1944). Similarly, every analyte release computed by the STOMP code can occur over several time steps, each of which can be denoted 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 may receive releases, VZDROP examines each STOMP release site to determine if it releases to the CFEST node being evaluated. For each STOMP release site that does receive a release, 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:

$$\Delta t = \text{MIN}(t_{C1}, t_{S1}) - \text{MAX}(t_{C0}, t_{S0})$$

The duration is used to express the fraction of the STOMP time step  $n$  that is within the current CFEST time step:

$$f_{\text{duration},n} = \frac{\Delta t}{t_{S1} - t_{S0}}$$

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:

$$Q_C = \sum_{n=1,m} f_{\text{duration},n} q_{S,n}$$

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 ( $f_{\text{duration},n}$ ) 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.

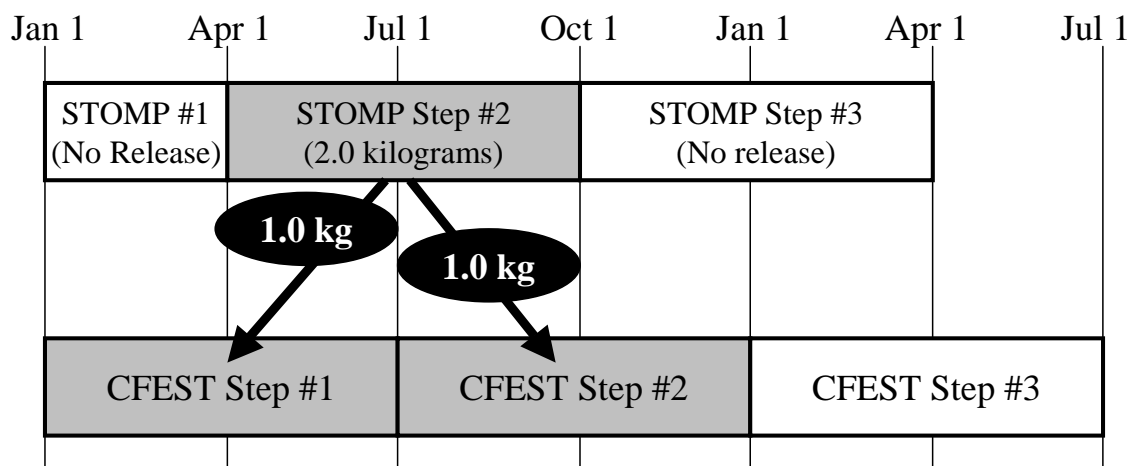
Before storing the mass source, it is converted to a concentration source if the liquid source already present in the CFEST L3I file for the current time step and node is nonzero. To do this, the liquid source rate already specified in the CFEST L3I file is multiplied by the CFEST time step duration 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 ( $\text{m}^3$ ) to obtain a concentration (in  $\text{kg}/\text{m}^3$ ). 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.

### **6.2.3 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.

## 6.2.4 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 nodes with catchment areas overlapping the vadose zone site release area based on the fractional areas involved.
- Upscaling of vadose zone area by the PRESTOMP code used to size the vadose zone wetted area for vadose flow and transport model is not taken into consideration in assignment of fluxes to the groundwater model; only the declared polygon boundary nodes found in the STOMP #SAC header information are used by VZDROP.

## 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 is a versatile tool that can be invoked independent of ESP for use in special case studies.

### **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 1.1. 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 compiler for the environment. In the Microsoft Windows environment, VZDROP is executed by any of the following DOS commands:

```
vzdrop-1  
vzdrop-1 vzdrop_case1.key  
vzdrop-1.exe vzdrop.key HIRES
```

In the first example, VZDROP is invoked and expects to use the default VZDROP keyword file named `vzdrop.key` which is located in the current directory. In the second example, VZDROP is invoked, and the VZDROP keyword file is explicitly named `vzdrop_case1.key`. The third example shows the use of the HIRES optional modifier; this modifier must be included when using the CFEST “003” version code that supports finite element grids with more than 10,000 surface nodes, and must be absent when using files for CFEST “001” version code that limits the number of surface nodes in the finite element grid to less than 10,000 nodes. When VZDROP is executed under the control of the ESP the HIRES modifier will automatically be included or excluded based on the form of the MODULE keyword in the ESD keyword file.

In the Linux environment, VZDROP is executed through any of the following Bourne Shell or C Shell commands:

```
vzdrop-1.exe  
vzdrop-1.exe vzdrop_case1.key  
vzdrop-1.exe vzdrop.key HIRES
```

These three commands have the same effect as the three examples described above for the Windows environment.

### **6.3.3 Code Control and Keyword Descriptions**

A single keyword 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 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 VZDROP Keyword 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 File Example 1

<pre>~VZDROP Key Input /home/ANALYSIS4/CA1_median/cfest/U238/1/input.lp1 /home/ANALYSIS4/CA1_median/cfest/U238/1/input.l3i Radioactive,9.7245E+3,Ci/g,H3,10, 1 /home/ANALYSIS4/CA1_median/vadose/241-A-101/U238/1/release</pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

When executed using this control file, VZDROP would read analyte release information found in the file /home/ANALYSIS4/CA1\_median/vadose/SXTank/U238/1/release, convert the release from activity expressed in Ci to mass expressed in kilograms using the specific activity of 9,724.5 Ci per gram (the

value for tritium in this case), and revise the CFEST file /home/ANALYSIS4/CA1\_median/cfest/U238/1/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 File Example 2

<pre>~VZDROP Key Input han_input_CA.lp1 han_input_CA.l3i Radioactive, 9.7245E+3, Ci/g, H3, 10, 10 /home/ANALYSIS4/CA1_median/vadose/216-A-10/H3/10/release /home/ANALYSIS4/CA1_median/vadose/216-A-5/H3/10/release /home/ANALYSIS4/CA1_median/vadose/216-A-21/H3/10/release /home/ANALYSIS4/CA1_median/vadose/216-U-8/H3/10/release /home/ANALYSIS4/CA1_median/vadose/216-S-7/H3/10/release /home/ANALYSIS4/CA1_median/vadose/216-S-21/H3/10/release /home/ANALYSIS4/CA1_median/vadose/216-S-9/H3/10/release /home/ANALYSIS4/CA1_median/vadose/216-S1#2/H3/10/release /home/ANALYSIS4/CA1_median/vadose/216-S-4/H3/10/release /home/ANALYSIS4/CA1_median/vadose/216-S-3/H3/10/release</pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

When the VZDROP code is used within the SAC framework, the ESPcode prepares the keyword control file internally. In this case, the file is always named vdrop.key and always contains as many STOMP release files as there are vadose zone sites identified by the SITE keywords in the SAC ESD file (see Section 2.1.27). In the SAC Initial Assessment, for example, 719 vadose zone sites were simulated and so the VZDROP keyword file created by the ESP would always specify 719 STOMP release files.

## 6.4 Data Files

The VZDROP keyword 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 file constitute inputs to the VZDROP code. 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. Two possible formats exist for the CFEST LP1 and L3I files, depending on whether they are prepared for the older “001” version that supported up to



10,000 surface nodes in the CFEST finite element grid, or the newer “003” version that supports up to 100,000 surface nodes. The “HIRES” modifier used when VZDROP is invoked informs VZDROP which format to assume in reading from the LP1 and L3I files and writing a revised L3I file.

#### 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 241-C-103, Realization 1, of the SAC Rev. 1 Composite Analysis Case 1 (median, 10,000-year) Assessment.

**Table 6.4** Example STOMP Release File

```
~STOMP SAC Mass/Activity Releases to Aquifer
#SAC STOMP input created by ESP 08/06/2004 - 13:34:30
#SAC Case ID : 2004 Composite Analysis 10,000-year (Median Inputs) Assessment
#SAC Template : vadose/241-C-103/Tc99/template_stompl.key
#SAC Site ID : 241-C-103
#SAC Site Points : 4
#SAC Site Eastings (m) : 5.7519431E+05, 5.7521456E+05, 5.7521456E+05, 5.7519431E+05,
#SAC Site Northings (m) : 1.3653737E+05, 1.3653737E+05, 1.3655762E+05, 1.3655762E+05,
#SAC Site Area (m^2) : 4.1008594E+02
#SAC Analyte : Tc99
#SAC Realization : 1
#SAC VADER Source Range,1,1,1,1,510,510,
3.705270595472E+09,s, 1.174242000000E+02,m, 6.997494133006E-14,
3.711797093208E+09,s, 1.174242000000E+02,m, 9.379988309630E-14,
3.718323590944E+09,s, 1.174242000000E+02,m, 1.185762610831E-13,
...
3.186893499429E+11,s, 1.174242000000E+02,m, 2.648197341729E-06,
3.189146688000E+11,s, 1.174242000000E+02,m, 2.583078811822E-06,
3.189462048000E+11,s, 1.174242000000E+02,m, 1.113840690611E-06,
```

#### 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, and the fraction of release for that vadose zone site assigned to this CFEST node.

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, and all of the polygonal area assigned to 216-U-8 falls within the catchment of CFEST node 1448. When VZDROP is run, 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 (1.000) ;
  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 (1.000) ;
  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, with 100 percent of releases from that site assigned to this node.

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 for 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 4 through 220 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; however, 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	V	ZZZZZZZ	DDDDD	R	R	OOOOO P
VZDROP      Version 3.4 Build: 29 Jan 2004						
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 : 08/18/2004 at 12:46:53						

```

Message originating in routine VZTOSZ
Message: Number of Vadose Zone Release Sites = 1023
        Number of Original Source/Sink Nodes = 769
        Number of Revised Source/Sink Nodes = 771

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      365.250  2.52332E-04  5.64386E-08
0002       365.250      730.500  2.77602E-04  6.20908E-08
0003       730.500     1095.750  5.14563E-04  1.15091E-07
<lines deleted>
0221  3672954.000  3691216.500  8.75897E-06  1.95910E-09
0222  3691216.500  3709479.000  4.04465E-05  9.04660E-09
0223  3709479.000  3709479.000  0.00000E+00  0.00000E+00

Message originating in routine VZTOSZ
Message: CCCMAS = 2.24000E-16

Message originating in routine VZTOSZ
Message: Total Release by STOMP = 8.89465E-02 Ci, or 1.98945E-05 kg
        Total Inject to CFEST = 8.89455E-02 Ci, or 1.98943E-05 kg
        Disprecency = 9.68575E-07 Ci (# 0.001%)

Message originating in routine VZDROP
Message: VZDROP Execution Successful
        Execution Finish: 08/18/2004 at 12:49:43
        Execution Duration: 169.0 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.

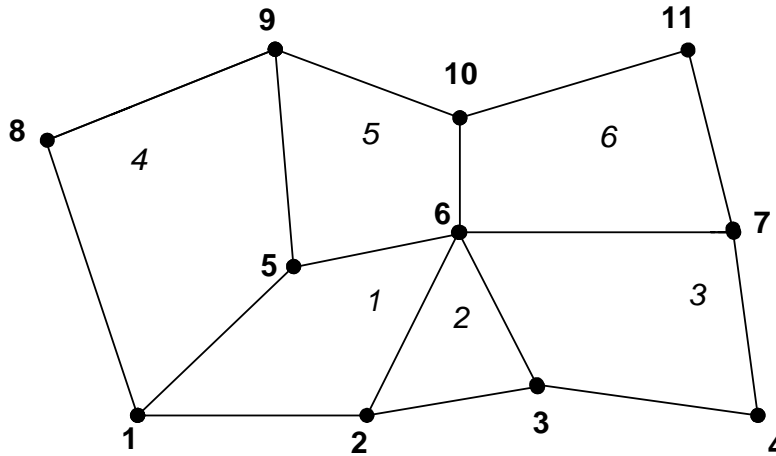
### **7.2 Algorithms and Assumptions**

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 not dependent upon each 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 two 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 overlap, 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 two 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. An example is shown in Figure 7.1. Example node numbers are 1 through 11. Example element numbers are 1 through 6.

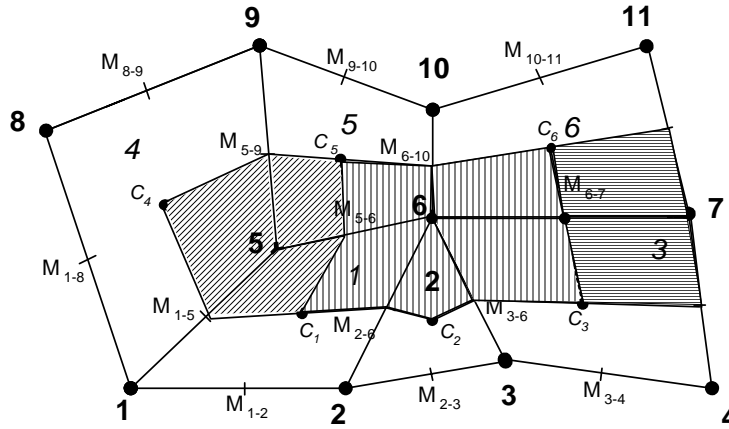


Element	Node	Node	Node	Node
1	1	2	6	5
2	2	3	6	
3	3	4	7	6
4	1	5	9	8
5	5	6	10	9
6	6	7	11	10

**Figure 7.1** Example Nodes and Elements Layout

Via the LPI 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 three or four 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 four 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  $M_{i,j}$  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  $C_1, M_{5-6}, C_5, M_{5-9}, C_4,$  and  $M_{1-5}$  in that order. Node 6 polygon is the broken line connecting  $C_1, M_{2-6}, C_2, M_{3-6}, C_3, M_{6-7}, C_6, M_{6-10}, C_5,$  and  $M_{5-6}$ . 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 ten sides. Node 7 is assumed to be on an edge of the finite element grid and thus is constructed from points  $C_3, M_{4-7}, C_7, M_{7-11}, C_6,$  and  $M_{6-7}$ .



**Figure 7.2** Example with Nodal Polygons Shaded

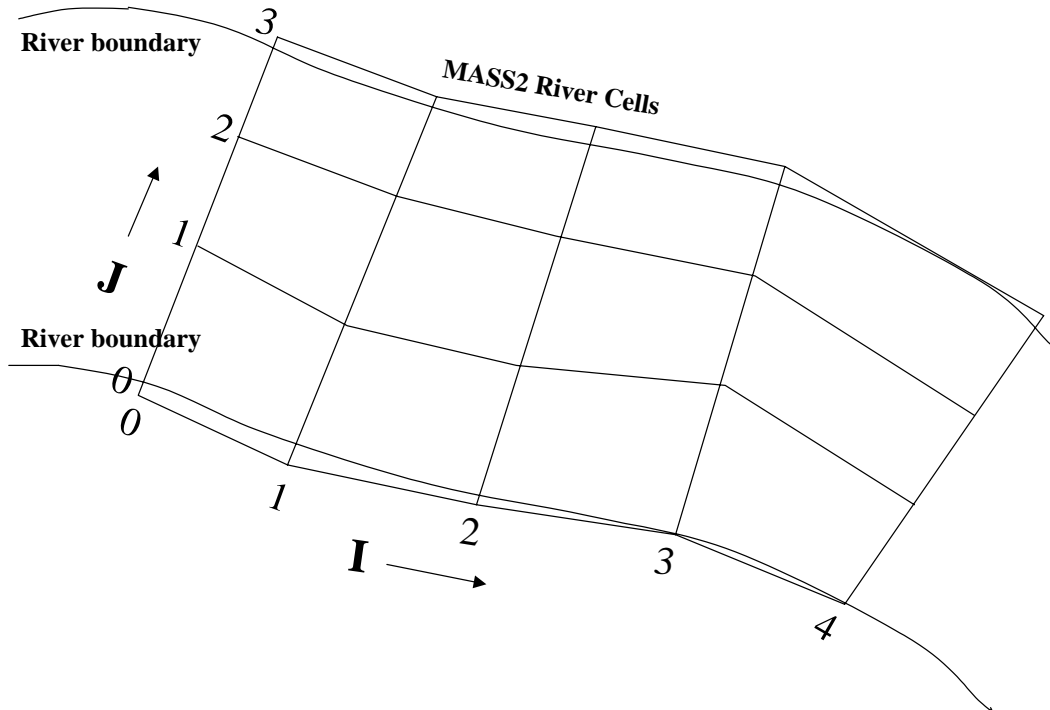
Not all CFEST nodes output to river cells. Nodes that output to the river are defined in the river interface nodes list file, a file that contains a simple list of node numbers. Nodes that are not listed in this file will not have entries in COV or TMS files.

### 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, or 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. Figure 7.3 shows 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,NI<sub>max</sub>,NJ<sub>max</sub>), 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 three 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 146,097 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 2.1.17 and 2.1.29). 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 three-dimensional 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 two-dimensional CFEST model, the concentration data are taken from the surface node. The concentrations are adjusted for units changes, then multiplied by  $(H_{\text{HEAD}} - Z)$  and divided by  $T_{\text{THICK}}$ , where:

$H_{\text{HEAD}}$  is hydraulic head (meters)  
 $Z$  is elevation (meters)  
 $T_{\text{THICK}}$  is the transport thickness parameter (meters).

This last calculation is a rough method for translating concentrations in a two-dimensional 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 1.1.

#### **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

<b>Line</b>	<b>Description</b>
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).
6	Relative path to Columbia River nodes file.

Line	Description
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 <i>MM-DD-YYYY HH:MM:SS</i> .
11	<p>Polygon intercept options, enter POLYINT or POLYCEN. Additional modifiers for this line are TTHICK, SKIP, RIVER, and CLIP.</p> <ul style="list-style-type: none"> <li>• TTHICK – transport layer thickness for a 2-D CFEST model run.</li> <li>• SKIP - Number of CFEST time planes to skip.</li> <li>• 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.</li> <li>• CLIP NODES – If specified, clips every node output to MASS2 in every time step to positive.</li> <li>• CLIP TOTAL – If specified, clips non-positive nodes to zero when total output to MASS2 is negative.</li> <li>• CLIP NONE – No clipping is performed. This is the default.</li> </ul> <p>Example: POLYINT : TTHICK 20.000 : SKIP 1 : RIVER R : CLIP NODES</p>
12	Realization number to process. Enter an integer.
13	Q.TAB file I format for reading node number list. Example ( <b>A,T13,0500I15</b> ).
14	Q.TAB file G format for reading flow flux values. Example ( <b>A,T13,0500G15.7</b> ).
15	M.TAB file I format for reading node number list. Example ( <b>A,T13,0500I15</b> ).
16	M.TAB file G format for reading flow flux values. Example ( <b>A,T13,0500G15.7</b> ).

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.
B	Relative path to the CFEST binary files directory – files required are B01, B03, B07, and ZZZ.

Key	Description
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: <b>VADER\116-B-7\H3\05\Vader.river, 565257.3, 145324.5</b>
-	-: 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 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 Line 11 shown in 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 impact 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, nominally 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.ct1
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 than 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 single 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 the example GWDROP.LIS specify the path and names of three MASS2 river cell geometry files. Prefix R: is present on these lines to identify river files. The file names are G:\Initial\GWDrop\hanfnad83m-pt.###, where ### is 000, 001, 002. Each of these files can portray a different geometry, and generally are situated end-to-end along the river flow direction. More than three files can be identified using more lines with the prefix R:.

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 water 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 water 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 are 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 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 to access the following keywords: FILE, REALIZATION, LOCATION, and TIMES. 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 ECDA and ECDA Index Files

Concentration data for every analyte are contained in a separate file. GWDROP writes to an ECDA file for every analyte being simulated, but only one analyte is processed during a single run of GWDROP. Sections 2.2.1 and 2.2.2 describe the format for the ECDA index and ECDA data files. The ECDA and index file names are supplied in the ESD keyword file.

#### 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. 1, runs are hanfnad83m-pt.000, hanfnad83m-pt.001, hanfnad83m-pt.002, through hanfnad83m-pt.056.

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 5.5.2.1 describes the contents of a VADER river release 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<sup>3</sup> 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 excerpts from a GWDROP log file (some lines in the file wrap to two lines in the table).

**Table 7.5** Excerpts from a GWDROP Log File

```

17-AUG-2004 09:06:47.221 :: GWDROP version 1.32 build 16 Dec 2003 started...
17-AUG-2004 09:06:47.271 :: GWDROP reading CFEST B01 input file...
17-AUG-2004 09:06:47.298 :: GWDROP opening/reading file cfest/C14/1/cfest.ct1
17-AUG-2004 09:06:47.300 :: GWDROP opening/reading file cfest/C14/1/lgethheld.ct1
17-AUG-2004 09:06:47.310 :: GWDROP opening/reading file cfest/C14/1/bincl/cfest.b01
17-AUG-2004 09:06:49.524 :: GWDROP reading CFEST border nodes file...
17-AUG-2004 09:06:49.537 :: GWDROP allocating dynamic memory...
17-AUG-2004 09:06:49.570 :: GWDROP computing CFEST node/element areas...
17-AUG-2004 09:06:49.983 :: Total of all CFEST element area = 806345806.2249995
17-AUG-2004 09:06:49.983 :: Total of all NODE POLYGON areas = 806345806.2250016
17-AUG-2004 09:06:49.985 :: GWDROP reading CFEST river nodes file...
17-AUG-2004 09:06:49.985 :: GWDROP opening/reading file mass2/col_river_node.dat
17-AUG-2004 09:06:50.051 :: GWDROP reading CFEST flow & analyte files...
17-AUG-2004 09:06:55.038 :: GWDROP opening/reading file cfest/C14/1/hheldq001.tab
17-AUG-2004 09:07:00.852 :: GWDROP opening/reading file cfest/C14/1/hheldm001.tab
17-AUG-2004 09:07:02.325 :: GWDROP reading MASS2 river cell files...
17-AUG-2004 09:07:02.344 :: GWDROP opening/reading file mass2/hanfnad83m-pt.000
17-AUG-2004 09:07:02.358 :: GWDROP opening/reading file mass2/hanfnad83m-pt.001
17-AUG-2004 09:07:02.359 :: GWDROP opening/reading file mass2/hanfnad83m-pt.002
<lines deleted>
17-AUG-2004 09:07:02.413 :: GWDROP opening/reading file mass2/hanfnad83m-pt.055
17-AUG-2004 09:07:02.413 :: GWDROP opening/reading file mass2/hanfnad83m-pt.056
17-AUG-2004 09:07:02.415 :: GWDROP opening/reading file mass2/hanfnad83m-pt.000
17-AUG-2004 09:07:02.415 :: GWDROP opening/reading file mass2/hanfnad83m-pt.001
17-AUG-2004 09:07:02.416 :: GWDROP opening/reading file mass2/hanfnad83m-pt.002
<lines deleted>
17-AUG-2004 09:07:02.485 :: GWDROP opening/reading file mass2/hanfnad83m-pt.055
17-AUG-2004 09:07:02.486 :: GWDROP opening/reading file mass2/hanfnad83m-pt.056
17-AUG-2004 09:07:34.869 :: GWDROP reading CFEST river nodes file...
17-AUG-2004 09:07:34.869 :: GWDROP opening/reading file mass2/col_river_node.dat
17-AUG-2004 09:07:34.874 :: Non river node 10775 mapped to river node 10774
17-AUG-2004 09:07:34.874 :: Non river node 12484 mapped to river node 12483
<lines deleted>
17-AUG-2004 09:07:34.878 :: Non river node 14553 mapped to river node 14554
17-AUG-2004 09:07:34.888 :: Open/read /home/ANALYSIS4/CAL_median/vadose/116-B-7/C14/1/vader.river
17-AUG-2004 09:07:34.993 :: Open/read /home/ANALYSIS4/CAL_median/vadose/116-D-5/C14/1/vader.river
<lines deleted>
17-AUG-2004 09:07:35.808 :: Open/read /home/ANALYSIS4/CAL_median/vadose/132-C-2/C14/1/vader.river
17-AUG-2004 09:07:35.810 :: Open/read /home/ANALYSIS4/CAL_median/vadose/1908-NE/C14/1/vader.river
17-AUG-2004 09:07:38.138 :: GWDROP writing CFEST output files for MASS2...
17-AUG-2004 09:08:05.846 :: Computing mass balances
17-AUG-2004 09:08:09.223 :: Maximum Q balance ratio = 1.14E-13
17-AUG-2004 09:08:09.223 :: Maximum M balance ratio = 2.30E-09
17-AUG-2004 09:08:09.223 :: clip negative CFEST FLOW inp = -0.862266E+12
17-AUG-2004 09:08:09.223 :: clip negative CFEST MASS inp = -0.324742E-05
17-AUG-2004 09:08:09.223 :: GWDROP writing ECDA output files...
17-AUG-2004 09:09:00.062 :: Starting LCONCSUB.
17-AUG-2004 09:09:41.823 :: Finished LCONCSUB.
17-AUG-2004 09:09:44.498 :: Starting ECDA file writing.
17-AUG-2004 09:11:33.081 :: Finished ECDA file writing.
17-AUG-2004 09:11:33.081 :: Number of negative CONC from CFEST = 30561
17-AUG-2004 09:11:33.081 :: GWDROP finished/successful.

```

### 7.4.2.2 Time Series Files (TMS)

TMS files contain the time series for water 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. As an example, Table 7.6 provides the first 20 lines of a flow TMS

file for serial number 10727. The numbers on the comment line are the CFEST node numbers contributing flux to the river cell mapped to serial 10727.

**Table 7.6** Excerpted Records from a Water Flow TMS file for Serial Number 10727

# mass2/C14/1/cfest/TMS-10727-FLOW.DAT		
12-31-1944	07:12:00	0.0000000000000000E+00
12-31-1945	12:00:00	0.4034747000000000E+07
12-31-1946	16:48:00	0.7168771000000000E+07
01-01-1948	00:00:00	0.8245145000000000E+07
12-31-1948	04:48:00	0.8866377699999999E+07
12-31-1949	12:00:00	0.8866377699999999E+07
12-31-1954	19:12:00	0.2697827770000000E+08
01-01-1960	00:00:00	0.7380393770000000E+08
12-31-1964	07:12:00	0.1461395377000000E+09
12-31-1969	12:00:00	0.2213559077000000E+09
12-31-1974	16:48:00	0.2817323477000000E+09
01-01-1980	00:00:00	0.4306745477000000E+09
12-31-1984	07:11:59	0.4885795177000000E+09
12-31-1989	12:00:00	0.4885795177000000E+09
12-31-1994	16:48:00	0.4885795177000000E+09
01-01-2000	00:00:00	0.4885795177000000E+09
12-31-2004	04:48:00	0.4885795177000000E+09
12-31-2009	12:00:00	0.4885795177000000E+09
12-31-2014	19:11:59	0.4885795177000000E+09

As a second example, Table 7.7 provides the first five years of data for a C14 TMS flux file for serial number 10727.

**Table 7.7** Excerpted Records from a Flux TMS file for Serial Number 10727

# mass2/C14/1/cfest/TMS-10727-C14.DAT		
12-31-1944	07:12:00	0.0000000000000000E+00
12-31-1945	12:00:00	0.0000000000000000E+00
12-31-1946	16:48:00	0.0000000000000000E+00
01-01-1948	00:00:00	0.0000000000000000E+00
12-31-1948	04:48:00	0.0000000000000000E+00
12-31-1949	12:00:00	0.0000000000000000E+00
12-31-1954	19:12:00	0.0000000000000000E+00
01-01-1960	00:00:00	0.581312116116600E-61
12-31-1964	07:12:00	0.581312116116600E-61
12-31-1969	12:00:00	0.581312116116600E-61
12-31-1974	16:48:00	0.293897554326420E-16
01-01-1980	00:00:00	0.903408957735042E-15
12-31-1984	07:11:59	0.107614960920370E-14
12-31-1989	12:00:00	0.107614960920370E-14
12-31-1994	16:48:00	0.107614960920370E-14
01-01-2000	00:00:00	0.107614960920370E-14
12-31-2004	04:48:00	0.107614960920370E-14
12-31-2009	12:00:00	0.107614960920370E-14
12-31-2014	19:11:59	0.107614960920370E-14

### 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 water 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

An ECDA 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 realization during any one run. GWDROP writes groundwater concentration data for the single realization into the middle of a record that may contain data for multiple realizations. This method significantly reduces the amount of I/O done by GWDROP but it also reduces the amount of error checking that can be performed. 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 AIRDROP – Atmospheric Concentration Extraction Module**

### **8.1 Code Purpose**

AIRDROP (AIR Data Restructure for Other Programs) is a data transfer utility code that performs the following functions:

- Extracts the results from the Atmospheric Transport Module (RATCHET2 code output), the Inventory Module, and the Vadose Zone Flow and Transport Module (STOMP code output)
- Integrates the above mentioned results
- Produces air concentrations for all analytes, times and locations required by the impacts modules that have an atmospheric medium.
- Produces surface concentrations from air deposition for all analytes, times and locations required by the impacts modules that have an atmospheric medium.
- Places the calculated concentration data in the Environmental Concentration Data Accumulator.

### **8.2 Algorithms and Assumptions**

The purpose of the AIRDROP code is to scale the results of RATCHET2 simulations, which simulate a unit release, by the release values produced by the INVENTORY and STOMP codes. No release or transport calculations are performed by the AIRDROP code. RATCHET2 simulates two types of releases; stack and surface. For each release type, a RATCHET2 simulation of a unit release is performed for each RATCHET2 node that represents a potential release location, generating an output file containing results for each node in the RATCHET2 grid. The AIRDROP code then calculates the final concentration based on INVENTORY and STOMP releases and translate these results to impact locations.

### **8.3 Code Environment**

The AIRDROP code is run under the control of the ESP. The AIRDROP code is run for one realization and one analyte at a time.

#### **8.3.1 Location in Processing Sequence**

The AIRDROP code runs after the STOMP model has completed execution for an analyte for a given realization as shown in Figure 1.1. AIRDROP can only be invoked for a given analyte and realization after successful simulation of RATCHET2, INVENTORY, and STOMP for that analyte and realization.

#### **8.3.2 How the AIRDROP Code Is Invoked**

AIRDROP may be run in either Microsoft Windows or Linux environments, provided the code has been compiled by an appropriate Fortran 95 compiler for that environment. In the Microsoft Windows environment AIRDROP is executed by the following DOS command:

```
AIRDROP
```

In the Linux environment AIRDROP is executed through the following Bourne Shell or C Shell commands:

```
airdrop-1.exe
```

where AIRDROP.EXE or airdrop-1.exe is the name of the executable program

### 8.3.3 Code Control and Keyword Descriptions

Data controlling operation of the code are provided in a single keyword control file that must be named “airdrop.ctl”. This file contains necessary identifiers (such as analyte and realization number) and the names and locations of the ESD file and the INVENTORY release file, and locations of RATCHET output files and the STOMP release files.

#### 8.3.3.1 Keyword Control File Format

The keyword control file for AIRDROP is prepared by the ESP when AIRDROP is used in the SAC framework. However, a user may also prepare the keyword control file directly. In either case, the input file has the same format. Table 8.1 lists the format of the airdrop.ctl file. All keywords are required and must be complete and in the order listed or the run will error terminate.

**Table 8.1** Description and Order of Keywords Used by the AIRDROP Code

Line	Required Keyword	Description
1	~AIRDROP control file	File identifier – must be the literal string shown here.
2	USER	User responsible for developing the airdrop.ctl keyword input file.
3	OS_SLASH	Directory delimiter (\ for Windows, / for Linux).
4	ANALYTE	This keyword is expected to be followed by 3 strings: SAC analyte ID (up to 6 characters), SAC analyte name (up to 72 characters) , Analyte class (up to 8 characters; must be one of the literal strings “Particle”, “Iodine”, or “Noble”).
5	REALIZATION	This keyword is expected to be followed by 2 integers: Current realization, and total number of realizations in the analysis.
6	ESD_KEYFILE	Name of ESD keyword file (without path).
7	RATCHET GRID	This keyword defines the RATCHET2 grid location and size and is expected to be followed by 5 values: Easting of RATCHET grid lower left corner (real), Northing of RATCHET grid lower left corner (real), RATCHET grid spacing in easting direction (integer), RATCHET grid spacing in northing direction (integer), Number of nodes in easting direction (integer), Number of nodes in northing direction (integer).

Line	Required Keyword	Description
8	RELEASE	Name of file containing list of release sites, including path.
9	RATCHET STACK	Keyword to signify that RATCHET stack release paths will follow.
10	nx	Number of paths that follow (i.e. number of RATCHET stack simulations - integer).
11...(nx+11)	N1.....nx	Path to RATCHET output files based on stack releases – one for each RATCHET simulation (character).
(nx+12)	RATCHET SURF	Keyword to signify that RATCHET surface release paths follow.
(nx+13)	ny	Number of paths that follow (i.e. number of RATCHET surface simulations - integer).
(nx+14)... (nx+14) + ny	N1.....ny	path to RATCHET output files based on surface releases – one for each RATCHET simulation.
(nx+14) + ny + 1	INVENTORY FILE	Keyword to signify path to vadose zone release files (STOMP output files) will occur on next line.
(nx+14) + ny + 2		Path to vadose zone release file.
(nx+14) + ny + 3	VADOSE PATH	Keyword to signify path to vadose zone release files (STOMP output files) will occur on next line.
(nx+14) + ny + 4		Path to vadose zone release files.

Easting and northing coordinates are defined in terms of the Lambert projection of the Washington State Plane North American Datum of 1983, expressed in meters.

### 8.3.3.2 Example Keyword Control Files

Table 8.2 shows an example airdrop control file. Full path names for files are specified for a Linux environment; hence, this control file would be suitable for execution from any location within the Linux environment.

**Table 8.2** AIRDROP Keyword Control File Example

<pre> ~AIRDROP control file : File identifier USER "SKW" : User identifier OS_SLASH "/" : Directory delimiter (\ for windows, / for unix) ANALYTE "C14","Carbon-14","Noble" : Analyte-id, analyte-name, analyte class REALIZATION 3 25 : NREAL, ESD_NREAL : Current Realization, Number of realizations ESD_KEYFILE "ESD_TEST.KEY" : Name of ESD key file to read RATCHET GRID 583401.7605 111718.4758 4000 4000 41 53 : RELEASE "/home/ANALYSISx/ratchet/release.lst" : name of file containing release sites RATCHET STACK          : Keyword to signify stack release paths to follow 5          : Number of paths to follow "/home/ANALYSISx/ratchet/03/stack/23-21" </pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

```

"/home/ANALYSISx/ratchet/03/stack/19-22"
"/home/ANALYSISx/ratchet/03/stack/23-27"
"/home/ANALYSISx/ratchet/03/stack/32-14"
"/home/ANALYSISx/ratchet/03/stack/7-33"
RATCHET SURFACE      : Keyword to signify surface release paths to follow
4      : Number of paths to follow
"/home/ANALYSISx/ratchet/03/surf/23-21"
"/home/ANALYSISx/ratchet/03/surf/19-22"
"/home/ANALYSISx/ratchet/03/surf/23-27"
"/home/ANALYSISx/ratchet/03/surf/32-14"
INVENTORY FILE : Keyword to signify inventory file name to follow
"/home/ANALYSISx/inventory/inv03.res"
VADOSE PATH     : Keyword to signify vadose path to follow
"/home/ANALYSISx/"

```

When executed using this control file, AIRDROP would read analyte release information found in the vadose zone release files and the inventory release files.

### 8.4 Data Files

The AIRDROP code reads from a keyword file, a release location file, RATHCET2 output files, INVENTORY and STOMP release files, and ECDA files. Summary information is written to a log file and concentration data are written to the ECDA file. Table 8.3 provides a summary of the files accessed by the AIRDROP code.

**Table 8.3** Files Accessed by the AIRDROP Code

File	Use	Description
airdrop.ctl	Input	AIRDROP keyword control file.
ESD keyword file	Input	Source of general information for the overall analysis.
release.lst	Input	File containing list or release locations and release type.
iodine.bin	Input	RATCHET2 output file containing time-integrated air concentration and deposition for iodine.
particle.bin	Input	RATCHET2 output file containing time-integrated air concentration and deposition for particles.
noble.bin	Input	RATCHET2 output file containing time-integrated air concentration for noble gases.
airborne	Input	Vadose Zone release (STOMP) output file – one for each site, realization and analyte.
invn.res	Input	Inventory (INVENTORY) release results file – one for each realization.
ECDA map file	Input	Map file for ECDA files.
ECDA file	Output	ECDA file, opened and modified by the AIRDROP code.
airdrop.log	Output	Log file containing summary information from the AIRDROP run.

## 8.4.1 Input Files

### 8.4.1.1 Data Provided in the AIRDROP Keyword Control File

The AIRDROP keyword control file provides necessary identifiers (such as analyte and realization number) and the names and locations for selected input and output files. Section 8.3.3 defines the keywords for this file. The keywords must all be present in the control file and in the order described in Table 8.1.

### 8.4.1.2 Data Read from the ESD file by the AIRDROP Code

The AIRDROP code reads information from the ESD keyword file to determine the names and coordinates of impacts locations at which to write out concentrations and the name of the ECDA file to write to. The ESD keywords used by AIRDROP are FILE, LOCATION, PERIOD, TIMES and END. Section 2.1 describes these keywords. The AIRDROP code identifies and stores the impact locations specified with the AIRC or AIRD modifier so that concentrations at those locations can be written to the ECDA file.

### 8.4.1.3 Description of the Release Location File

The release location file provides the AIRDROP code with names and locations of release sites. This file, which must be named release.lst, is generated by the ESP code. Table 8.4 shows an example release location file. The release.lst file has the following format:

- **First Line.** The number of sites listed in the file.
- **Succeeding Lines.** For the number of lines listed in line 1, each line contains the following information: Site-ID (character), Easting (real), Northing (real), RATCHET2 node in easting (integer), RATCHET2 node in northing (integer), release height (real - negative number denotes surface release).

**Table 8.4** Example Release Location File

23					
216-U-1%2-Fast	567243.2	135001.9	13	39	-1.0000
100-B-15	565460.0	145500.0	12	44	-1.0000
100-B-3	565290.0	144369.0	12	43	-1.0000
100-B-5	565437.0	144568.0	12	43	-1.0000
100-B-8	565342.2	144504.9	12	43	-1.0000
100-C-3	565392.5	143952.0	12	43	-1.0000
100-C-6	565387.8	143991.5	12	43	-1.0000
100-D-23	573751.2	151234.2	16	47	-1.0000
100-D-24	573745.6	151540.8	16	47	-1.0000
100-D-29	573832.4	152132.5	16	47	-1.0000
100-D-3	573818.6	151513.9	16	47	-1.0000
100-D-32	573857.4	151432.5	16	47	-1.0000
100-D-40	574039.1	151546.0	16	47	-1.0000
100-D-42	573848.8	151522.9	16	47	-1.0000
100-D-43	573852.0	151507.5	16	47	-1.0000

100-D-45	573821.5	151500.8	16	47	-1.0000
100-D-47	574000.1	151493.6	16	47	-1.0000
100-D-49	573900.0	151800.0	16	47	-1.0000
100-D-53	573753.0	151209.6	16	47	-1.0000
100-D-59	573250.8	151552.2	16	47	-1.0000
100-D-60	574000.0	152900.0	16	48	-1.0000
100-D-8	572625.0	151569.5	16	47	-1.0000
100-F-10	580490.2	147590.9	20	45	-1.0000

The AIRDROP code uses the RATCHET easting and northing node numbers to identify which RATCHET2 output file to scale for each release location. Releases at sites that correspond to the same RATCHET2 node will be summed, and that sum will be used as the scaling factor for that RATCHET2 node. The concentration field that results is then sampled by the AIRDROP code at the specified impact locations and those values are output to the ECDA file.

#### 8.4.1.4 Description of RATCHET2 Output Files

The RATCHET2 code has been modified for SAC to produce three output files, one for each analyte type. Each file contains header information and values at each RATCHET output node. These file names are assigned in the RATCHET2 input file (ratchet.rsf) which is generated by the ESP program. The files are described in the list below:

- iodine.bin – binary file containing a  $\chi/Q$  field for iodine releases. This file contains air concentrations and surface deposition.
- particle.bin – binary file containing a  $\chi/Q$  field for releases of analytes defined as particles. This file contains air concentrations and surface deposition.
- noble.bin – binary file containing a  $\chi/Q$  field for releases of analytes defined as noble gases. This file contains air concentrations only.

#### 8.4.1.5 Description of STOMP Airborne File

The SAC modification set to the STOMP code includes a subroutine that records releases of analyte mass to the atmosphere with time in a special release file that is always named “airborne”. This feature is invoked by use of the STOMP ~SAC Release Card in a STOMP input file. In response, STOMP generates a file with the following format:

- **First Line.** The literal string “~STOMP SAC Mass/Activity Releases to Atmosphere”.
- **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 of Analyte (in kilograms or Curies) (real).

Table 8.5 provides an example airborne release file. In this example, releases of tritium (H3) are recorded by STOMP for vadose zone site 216-A-8, realization 1, of a SAC assessment.

**Table 8.5** Example STOMP Airborne Release File

```

~STOMP SAC Mass/Activity Releases to Atmosphere
#SAC STOMP input created by ESP    08/06/2004 - 13:34:30
#SAC Case ID      : 2004 CA 10,000-year (Median Inputs)
#SAC Template    : vadose/100-D-47/I129/template_stomp2.key
#SAC Site ID     : 100-D-47
#SAC Site Points : 4
#SAC Site Eastings (m) : 5.7396869E+05, 5.7403156E+05, 5.7403156E+05,
5.7396869E+05,
#SAC Site Northings (m): 1.5146216E+05, 1.5146216E+05, 1.5152509E+05,
1.5152509E+05,
#SAC Site Area (m^2)  : 3.9568652E+03
#SAC Analyte       : I129
#SAC Realization   : 1
#SAC VADER Source Range,1,1,1,1,113,146,
0.000000000000E+00,s, 0.000000000000E+00,m, 0.000000000000E+00,
1.499999930000E+11,s, 0.000000000000E+00,m, 0.000000000000E+00,
    
```

#### 8.4.1.6 Description of the INVENTORY Release Results File

The INVENTORY code produces a single release result file for each realization. This file contains information for all release sites and analytes and is described in Section 4.5.2.4. The AIRDROP code scans this file for site and analytes that are designated as releases to the atmosphere and uses these values along with the vadose zone releases to scale the RATCHET2 results.

### 8.4.2 Output Files

Details of files modified or generated by the AIRDROP code are described in the following sections.

#### 8.4.2.1 AIRDROP Log File

The log file contains a heading with the run time, title, and user name from the airdrop.ctl 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 8.6 provides an example summary output file for CRDROP.

**Table 8.6** Example Log File for AIRDROP

```

          A      I I I I I R R R R R      D D D D D D      R R R R R      O O O      P P P P P
        A  A      I  R   R  D      D R   R   O   O  P   P
        A  A      I  R   R  D      D R   R   O   O  P   P
        A A A A A A      I  R R R R R      D      D R R R R R      O   O  P P P P P
        A  A      I  R   R  D      D R   R   O   O  P
        A  A      I  R   R  D      D R   R   O   O  P
    
```



*User Instructions for the Systems Assessment Capability, Rev. 0, Computer Codes*  
*Volume 1: Inventory, Release, and Transport Modules*

---

A    A I I I I I R    R D D D D D R    R    O O O    P

AIRDROP            Version 1.2  
Last Modified on 09 Jan 2004

-----  
AIR Data Restructure Other Programs  
SAC Rev. 1 Data Processing  
-----

Developed By Battelle Memorial Institute  
Pacific Northwest National Laboratory  
Richland, Washington

Current Run ID = 200407151424    User Name = Eslinger-DWE-WEN  
System Date = 07-15-20    System Time = 14:24:49

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 AIRDROP control file =====

Title: ~airdrop control file  
User: Eslinger-DWE-WEN  
Operating System: unix  
Delimiter: /  
Analyte name: Iodine-129  
Analyte id: I129  
Analyte class: IODINE  
Current realization is    1 of    1 total realizations  
ESD Keyfile name: ../../../../ESD\_CA1\_median-Rat.key  
RATCHET grid coordinates:  
  Grid center - Easting: 583402. Northing: 111718.  
  Delta x: 2000. Delta y: 2000.  
  Number in x: 41 Number in y: 53  
Name of file containing list of release sites: ../../release.lst  
Number of RATCHET stack release files : 0  
Number of RATCHET surface release files : 11  
Names of RATCHET surface release files :  
  ../surf/13-39  
  ../surf/12-44  
  ../surf/12-43  
  ../surf/16-47  
  ../surf/16-48  
  ../surf/20-45  
  ../surf/18-47  
  ../surf/18-48  
  ../surf/14-44  
  ../surf/14-45

*User Instructions for the Systems Assessment Capability, Rev. 0, Computer Codes*  
*Volume 1: Inventory, Release, and Transport Modules*

---

```
../surf/15-46
Path to and name of Inventory release file: ../../../../inventory/inv1.res
Path to Vadose zone airborne release files: ../../../../vadose

===== AIRDROP control file read successfully =====
=
===== Reading ESD keyword file =====

  Reading ESD Keywords - First Pass
  Reading ESD Keywords - Second Pass

===== ESD keyword file read successfully =====

===== Reading release site file =====

===== List of release sites read successfully =====

===== Mapping release site and impact locations to RATCHET nodes =====

Center of RATCHET grid (x,y): 21,27
Release Site: 216-U-1%2-Fast   RATCHET node (x,y): 13,39
Release Site: 100-B-15       RATCHET node (x,y): 12,44
Release Site: 100-B-3        RATCHET node (x,y): 12,43
Release Site: 100-B-5        RATCHET node (x,y): 12,43
Release Site: 100-B-8        RATCHET node (x,y): 12,43
Release Site: 100-C-3        RATCHET node (x,y): 12,43
Release Site: 100-C-6        RATCHET node (x,y): 12,43
Release Site: 100-D-23       RATCHET node (x,y): 16,47
Impact Location: UH0001 RATCHET node (x,y): 17,48
Impact Location: UH0002 RATCHET node (x,y): 17,48
Impact Location: UH0003 RATCHET node (x,y): 17,48
Impact Location: UH0004 RATCHET node (x,y): 18,48
Impact Location: UH0005 RATCHET node (x,y): 18,48
Impact Location: UH0006 RATCHET node (x,y): 17,48
Impact Location: UH0007 RATCHET node (x,y): 17,48
Impact Location: UH0008 RATCHET node (x,y): 17,48
Impact Location: UH0009 RATCHET node (x,y): 18,48
Impact Location: UH0010 RATCHET node (x,y): 18,48
Impact Location: UH0011 RATCHET node (x,y): 16,48
Impact Location: UH0012 RATCHET node (x,y): 16,47
Impact Location: UH0013 RATCHET node (x,y): 17,47

=== Release sites and impact locations assigned to RATCHET nodes successfully ===

===== Reading ECDA Map file =====
Header information from the ECDA map file
  ECDA_PTITLE = 2004 Composite Analysis 10,000-year (Median Inputs) Assessment
  ECDA_PRGNAM = ECDA
  ECDA_PRGVER = 2.00.A.4
  ECDA_PRGDAT = 16 Oct 2003
  ECDA_USRNAM = Eslinger-DWE-WEN
```

*User Instructions for the Systems Assessment Capability, Rev. 0, Computer Codes*  
*Volume I: Inventory, Release, and Transport Modules*

---

```
ECDA_CRUNID = 20040617140439
ECDA_BLOCK  = 23155
ECDA_RECLEN = 18
ECDA_NREAL  = 1
===== ECDA Map file read successfully =====
Opening ECDA Concentration file: /home/ANALYSIS4/CA1_median/ecda/I129_CA1_median.dat

===== Reading RATCHET output files =====
Number of RATCHET stack release output files  0
Creating RATCHET plot file ratsurf.dat
Creating RATCHET plot file ratsurfdep.dat
Number of RATCHET surface release output files 41
Reading from RATCHET output file ../surf/13-39/iodine.bin
RATCHET simulation year: 1994
Reading from RATCHET output file ../surf/12-44/iodine.bin
RATCHET simulation year: 1994
Reading from RATCHET output file ../surf/12-43/iodine.bin
RATCHET simulation year: 1994
Reading from RATCHET output file ../surf/16-47/iodine.bin
RATCHET simulation year: 1994
Reading from RATCHET output file ../surf/16-48/iodine.bin

===== Reading INVENTORY release file =====
Qsurf filled for inventory for the following values...
SITE, RELEASE HEIGHT, RATCHET EAST, RATCHET NORTH, ANALYTE, YEAR, INV RELEASE,
CUMMULATIVE VALUE
200-E-137      surf  16 39 i129   1945 1.78813003E-02 1.78813003E-02
200-E-137      surf  16 39 i129   1950 5.29979006E-04 5.29979006E-04
200-W-45       surf  13 40 i129   1945 1.78718995E-02 1.78718995E-02
200-W-45       surf  13 40 i129   1950 4.88427992E-04 4.88427992E-04
200-W-45       surf  13 40 i129   1955 1.46097003E-03 1.46097003E-03
291-S-1        surf  13 38 i129   1955 4.46819980E-03 4.46819980E-03
291-S-1        surf  13 38 i129   1960 2.10115989E-03 2.10115989E-03
291-S-1        surf  13 38 i129   1965 4.26915986E-03 4.26915986E-03
291-WTP        surf  17 39 i129   2005 6.85975998E-11 6.85975998E-11
291-WTP        surf  17 39 i129   2006 1.00867004E-08 1.00867004E-08
291-WTP        surf  17 39 i129   2007 1.76011000E-07 1.76011000E-07
291-WTP        surf  17 39 i129   2015 1.10288001E-05 1.10288001E-05
291-WTP        surf  17 39 i129   2020 6.88382988E-06 6.88382988E-06
291-WTP        surf  17 39 i129   2025 5.45506009E-06 5.45506009E-06
296-A-13       surf  17 39 i129   1960 1.34052997E-02 1.34052997E-02
296-A-13       surf  17 39 i129   1965 1.34036997E-02 1.34036997E-02
296-A-13       surf  17 39 i129   1970 3.06864991E-03 3.06864991E-03
296-A-13       surf  17 39 i129   1985 0.291004986 0.291004986
296-A-13       surf  17 39 i129   1990 0.106691003 0.106691003

===== Reading Vadose zone release (airborne) files =====
SITE, RATCHET EAST, RATCHET NORTH, ANALYTE, YEAR, VADOSE RELEASE, CUMMULATIVE RELEASE
100-B-5        12 43 I129   1945 1.014648429582542E-15 1.01464838E-15
100-B-5        12 43 I129   1950 1.014648429582542E-15 1.01464838E-15
100-B-5        12 43 I129   1955 7.619607699166057E-14 7.61960753E-14
100-B-5        12 43 I129   1960 2.878077743970677E-16 2.87807766E-16
100-B-5        12 43 I129   1965 1.729807567155059E-14 1.72980764E-14
100-B-5        12 43 I129   1970 1.729807567155059E-14 1.72980764E-14
```

```
100-B-5      12 43 I129   1975 1.729807567155059E-14 1.72980764E-14
100-B-5      12 43 I129   1980 1.734546766119462E-14 1.73454678E-14
100-B-5      12 43 I129   1985 1.729807567155059E-14 1.72980764E-14
100-B-5      12 43 I129   1990 1.729807567155059E-14 1.72980764E-14
100-B-5      12 43 I129   1991 1.729807567155059E-14 1.72980764E-14

===== Calculating concentration grids =====
Creating RATCHET plot file concurf.dat
Creating RATCHET plot file depsurf.dat
Creating RATCHET plot file totconc.dat
Creating RATCHET plot file totdep.dat

Writing to ECDA file for 272 impact times and 4351 impact locations

Impact location: UH0001 Impact time: 1945 Concentration:    0.195E-16
Impact location: UH0001 Impact time: 1945 Deposition:      0.453E-19
Impact location: UH0002 Impact time: 1945 Concentration:    0.195E-16
Impact location: UH0002 Impact time: 1945 Deposition:      0.453E-19
Impact location: UH0003 Impact time: 1945 Concentration:    0.195E-16
Impact location: UH0003 Impact time: 1945 Deposition:      0.453E-19
Impact location: UH0004 Impact time: 1945 Concentration:    0.180E-16
Impact location: UH0004 Impact time: 1945 Deposition:      0.439E-19

Message originating in routine AIRDROP
Message: AIRDROP Execution Successful
      Execution Finish: 07/15/2004 at 14:31:22
      Execution Duration:    393.0 sec
```

#### **8.4.2.2 Modifications Made to the ECDA File**

The ECDA file contains concentrations of analytes at specific locations in the environment. The AIRDROP program updates these concentrations for the specified analyte and realization at all impact locations. The concentrations are calculated by the AIRDROP code based on RATCHET2 results and vadose zone and inventory releases as described in Section 8.3.

## **9.0 CRDROP – Columbia River Concentration Extraction Module**

### **9.1 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 all aquatic impact locations.

### **9.2 Algorithms and Assumptions**

The code reads data from river model result files and places selected data into the ECDA data files (see Section 2.2.1). No release or transport calculations are performed by the CRDROP code. The location selection algorithm chooses the river solution at the closest location to the desired location for impacts calculations. Section 9.5.2 describes the selection algorithm.

### **9.3 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.

#### **9.3.1 Location in the Processing Sequence**

The CRDROP code runs after the MASS2 (river) model has completed execution for an analyte for a given realization. Figure 1.1 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. The CRDROP\_INDEX code is described in Section 9.5.

#### **9.3.2 How the CRDROP Code Is Invoked**

CRDROP runs only in a Linux environment under the control of the ESP. CRDROP is executed through the following Bourne Shell or C Shell command:

```
crdrop-1.exe "ESDfilename" N1 "Analyte ID"
```

where `crdrop-1.exe` is the name of the executable program, "ESDfilename" is the name of the ESD keyword file (the file name can include path information), N1 is the number of the current realization being processed, and "Analyte ID" is an analyte identification string.

An example invocation of CRDROP for realization 12 of an I129 analysis would be the following:

```
CRDROP "/home/test/ESD.key" 12 "I129"
```

### 9.3.3 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 in the ESD keyword file. The following sections describe the keywords used in the `crdrop.key` file. Table 9.1 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 9.1** Summary of Keywords Used by the CRDROP Code

<b>Keyword</b>	<b>Source</b>	<b>Information Used by CRDROP</b>
END	ESD file and CRDROP.KEY	Signifies end of a keyword file.
EXECUTE	CRDROP.KEY	Cause complete execution of CRDROP.
FILE	ESD file	ECDA file name for this analyte and the name of the ECDA index map file.
FILE	CRDROP.KEY	Grid index file and path to the MASS2 output file.
IGNORE	CRDROP.KEY	Optionally ignore data from MASS2 after a specified year.
OS	ESD file	Operating system information.
REALIZATION	ESD file	Number of realizations in analysis.
REPORT	CRDROP.KEY	Name of output report for CRDROP run summary and error messages.
TITLE	CRDROP.KEY	Title of <code>crdrop.key</code> keyword input file.
USER	CRDROP.KEY	User responsible for developing the <code>crdrop.key</code> keyword input file.

#### 9.3.3.1 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 following is this keyword's syntax:

```
END
```

#### 9.3.3.2 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 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 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. The following is this keyword's syntax:

```
EXECUTE
```

### **9.3.3.3 FILE Keyword for CRDROP**

The FILE 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. The following is this keyword's syntax:

```
FILE [MASS2="quote 1" | INDEX_CR="quote 1"]
```

The quote string associated with the MASS2 modifier contains indicates the directory path (no file name) to the river model output files. A quote string containing a single blank should be used if the code is started in the directory where the MASS2 file resides and local files names are desired. The actual name of the river model output file is generated during execution of CRDROP from the analyte and realization being analyzed and is not included in the input quote string. The quote string associated with the INDEX\_CR contains the path/file name for the location cross-index file. Both modifiers may be included on the same FILE keyword record. Three example keywords are the following:

```
FILE MASS2="F:\test\river\  
FILE INDEX_CR="CrossIndex.grd"  
FILE MASS2=" " INDEX_CR "CrossIndex.grd"
```

### **9.3.3.4 IGNORE Keyword for CRDROP**

The optional IGNORE keyword instructs CRDROP to ignore all data from MASS2 after a specified year. The following is this keyword's syntax:

```
IGNORE N1
```

The single numerical value entered on this keyword is a calendar year. All data in MASS2 output files after this calendar year will be ignored by CRDROP. All data in the MASS2 output file will be processed if this keyword is not entered.

### **9.3.3.5 REPORT Keyword for CRDROP**

The REPORT keyword is used to define the name of the output report (summary log) file. It must be the first keyword entered in the keyword file. The following is this keyword's syntax:

```
REPORT ["quote"]
```

The name of the report file is entered in a quote string. File names up to 200 characters long are supported, and path names can be included. An example REPORT keyword is the following:

```
REPORT "/SAC/SystemCodes/Test.rpt"
```

### **9.3.3.6 TITLE Keyword for CRDROP**

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 following is this keyword's syntax:

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

The title is entered in a quote string. 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 RIPSAC code."
```

### **9.3.3.7 USER Keyword for CRDROP**

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 following is this keyword's syntax:

```
USER "John Q. Public"
```

## **9.4 Data Files**

The CRDROP code reads from six input files and writes to two output files. A summary of these files is the following:

- CRDROP keyword file: This input file is the primary source of run control information.
- ESD keyword file: This input file is the general information for the overall analysis.
- Location cross-index file: This input file provides location cross-index values between the river model grid and the impact locations.
- River model result file: This input file contains the provides concentration values generated by the river model.
- CRDROP report file: This output file contains summary output information for the CRDROP analysis, including error messages, if any.
- ECDA index file: This input file contains indexing information for the ECDA files.
- ECDA data file: This output binary file contains the concentration data from the river model that will be used later by the impacts codes.

### **9.4.1 Input Files**

The following sections provide details of files providing input to the CRDROP code.



### 9.4.1.1 Data Provided in the CRDROP Keyword File

The CRDROP keyword file provides file names for selected input and output files (Table 9.2). Section 9.3.3 defines the keywords in this file. The first keyword record in the file must be the REPORT keyword, which provides the name of the file to be used for saving summary information on the analysis 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 or the code will error terminate.

**Table 9.2 Example Keyword File for CRDROP**

```
REPORT "crdrop.out"
TITLE "2004 Composite Analysis 10,000-year (Median Inputs) Assessment"
USER "Eslinger-DWE-WEN"
EXECUTE
! Ignore data after the year 10205
IGNORE 12050
!
! Cross index file bewteen MASS2 grid and impact locations
FILE INDEX_CR "CrossIndex.grd"
!
! MASS2 output file path
FILE MASS2 ""
!
END
```

### 9.4.1.2 Data Read from the ESD file by the CRDROP Code

The CRDROP code reads control information from the ESD keyword file to determine the operating system, names of the ECDA files to access, and the number of realizations allowed for the analysis. The ESD keywords used by CRDROP are FILE, OS, REALIZAT, and END. Section 2.1 describes these keywords.

### 9.4.1.3 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 9.5. The file has the information and format indicated in Table 9.3.

**Table 9.3 Content and Format of the Location Cross-Index File**

Record	Information	Description
1	Descriptive title for the file	Character variable: String of up to 72 characters containing a problem title.
2	Number of locations	Integer variable: Number of impact locations.

Record	Information	Description
Remaining records	ID of impact location	Character variable: String of up to 6 characters containing the ID for the location.
	River location index	Integer variable: Index of the river grid location that corresponds to the current impact location.
	Easting coordinate	Real variable: Easting coordinate (m) for the location. These coordinates are defined in terms of the Lambert projection of the Washington State Plane North American Datum of 1983, expressed in meters.
	Northing coordinate	Real variable: Northing coordinate (m) for the location. These coordinates are defined in terms of the Lambert projection of the Washington State Plane North American Datum of 1983, expressed in meters.

Table 9.4 shows an example cross-index file for an analysis using 8 locations.

**Table 9.4** Example Cross-Index File for CRDROP

"Test keyword file for INDEX_CRDROP"			
8," Impact locations"			
"UH0001",	0,	0.0,	0.0,
"UH0002",	0,	0.0,	0.0,
"UH0003",	0,	0.0,	0.0,
"QTV007",	1931,	559037.0,	146013.0,
"QTV008",	1932,	559027.0,	146042.0,
"QTV009",	1932,	559016.0,	146067.0,
"QTV010",	1933,	559000.0,	146099.0,
"QHRWI1",	1766,	595499.6,	109743.6,

The cross-index file prepared by the CRDROP\_INDEX code contains the easting and northing coordinate value for the river location in addition to the river location index values. However, the easting and northing information is not read or used by the CRDROP code.

#### 9.4.1.4 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. 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 four-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  
Sr90\_0020.RIV

The content of the river model output file is organized in groups by time periods (years). Within each time data set, concentration data are provided for three media for all MASS2 river locations. Table 9.5 describes the file format and content of MASS2 river model output files.

**Table 9.5** Structure of MASS2 River Model Output Files

<b>Record</b>	<b>Parameter</b>	<b>Type</b>	<b>Description</b>
1	Title	String	Descriptive title for the file.
	Analyte name	String	Name of analyte for which data are provided in this file.
	Number of time periods	Integer	Number of time period for which data are 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 river bottom sediment analyte concentration for current year and location.

Table 9.6 provides excerpts from a river output file for the analyte Tc99. The units of the concentration parameters are as given in Section 2.2.1. CRDROP makes no modifications to the numerical values in transferring to the ECDA files.

**Table 9.6** 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)

```

## 9.4.2 Output Files

Details of files modified or generated by the CRDROP code are described in the following sections.

### 9.4.2.1 CRDROP Report File

The report file 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 9.7 provides an example report file for CRDROP.

**Table 9.7** Example Report File for CRDROP

```

          CCCCC RRRRR DDDDD RRRR   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 P P P P
C       R R   D   D R R   O   O P
C       C R   R D   D R   R   O   O P
CCCCC R   R DDDDD R   R   OOO  P

          CRDROP      Version 2.0.B.1
          Last Modified on 5 Sep 2004

-----
Columbia River Data Restructure Other Programs
          SAC Rev. 1 Data Processing
-----

Developed By Battelle Memorial Institute
Pacific Northwest National Laboratory
          Richland, Washington

Current Run ID = 200409051327  User Name = Engel, Eslinger,

```

```
System Date = 09-05-2004   System Time = 13:27:07

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. 1 Confirm-median Assessment
User: Engel, Eslinger,
Current realization is 1 of 1 total realizations
Operating System is: UNIX, Delimiter is: /
Analyte ID: H3

Processing based on keyword data
File = crdrop.key
Allocate space using FIRST_ALLLOCATE
CALL SCAN_FILES_ANALYTES
Read cross-index values for MASS2 grid locations
Opening MASS2 output file
Opening ECDA file: /home/ANALYSIS3/Confirm-median/ecda/H3_Confirm-median.dat
Date: 09-05-2004 Time: 13:28:14 CRDROP Normal Termination
```

#### **9.4.2.2 Modifications Made to ECDA Files**

The ECDA files contain concentrations of analytes at specific locations in the environment. The CRDROP program updates these concentrations for one analyte and one realization at all river locations in a single run. 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. Media concentrations for all river locations is read one year at a time. If the time is one of the times when ECDA are to be saved, then the river model concentration data are 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 desired locations. 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.

## **9.5 CRDROP\_INDEX Utility Program**

### **9.5.1 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.

### **9.5.2 Algorithms and Assumptions**

The CRDROP\_INDEX code reads the MASS2 river grid locations and the location coordinate data for all ECDA locations. The processing involves finding the river grid location that is the closest 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.

### **9.5.3 Code Environment**

The CRDROP\_INDEX 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 ESP.

#### **9.5.3.1 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. In addition, it must be preceded by a run of the ECDA code.

### **9.5.3.2 How the CRDROP\_INDEX Code Is Invoked**

CRDROP\_INDEX can run under either the Windows or the Linux operating systems. Under the Windows operating system (2000, or XP), CRDROP\_INDEX executes in a DOS box. A run of CRDROP\_INDEX is initiated by entering the following command line:

```
CRDROP_INDEX "Keyfilename"
```

Under the Linux operating system CRDROP\_INDEX is executed through any of the following Bourne Shell or C Shell commands:

```
crdrop_index-1.exe "Keyfilename"
```

For these commands, CRDROP\_INDEX.EXE or crdrop\_index-1.exe is the name of the executable program and "Keyfilename" is the name of an existing control keyword file. Both the name of the executable program and the keyword file may contain path information. If CRDROP\_INDEX is invoked without entering the name of the keyword file, then the code will prompt the user for the file name. If CRDROP\_INDEX cannot find or open the keyword file, then the code will terminate execution after writing an error message to the standard output device.

### **9.5.4 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.

#### **9.5.4.1 DISTANCE Keyword for CRDROP\_INDEX**

The DISTANCE keyword defines the maximum distance allowed between an impact location and the nearest river grid point. The following is this keyword's syntax:

```
DISTANCE [N1]
```

The single numerical value is the maximum distance (meters) allowed between an 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 is as follows:

```
DISTANCE 1000.0
```

This keyword indicates a distance criterion of 1000 meters is to be used in the analysis.

#### **9.5.4.2 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 following is this keyword's syntax:

```
END
```

### **9.5.4.3 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
```

### **9.5.4.4 FILE Keyword for CRDROP\_INDEX**

The FILE keyword is used to specify file names for the input river model grid file and the location cross-index output file. The following is this keyword's syntax:

```
FILE {MASS2GRI="quote1"} {INDEX_CR="quote2"}
```

The quote string associated with the MASS2GRI modifier contains the name of the river model grid file. File names may be up to 180 characters in length. The quote string associated with the modifier

INDEX\_CR contains the file name for the output location cross-index file. Both modifiers may be included on the same FILE keyword or may be entered on separate keywords. Two example keywords are as follows:

```
FILE MASS2GRID "crdrop-grid.dat" compute  
FILE GRID "CrossIndex.grd" compute
```

### **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 "quote"
```

The single input string contains the report file name. The file name may be up to 180 characters in length.

### **9.5.4.5 TITLE Keyword for CRDROP\_INDEX**

The TITLE keyword provides a descriptive file title for the INDEX.KEY file. The problem title will be written to the report file. 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. 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."
```



### 9.5.4.6 USER Keyword for CRDROP\_INDEX

The USER keyword identifies 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 report file. The following is this keyword's syntax:

```
USER [ "user name" ]
```

where the "user name" is a string of length up to 16 characters. An example USER keyword is the following:

```
USER "John Doe"
```

### 9.5.5 Data Files

The CRDROP\_INDEX code reads two keyword files, a river model grid file, and the ECDA map index file. Summary information is written to the report file and an impact location cross-index file is generated. The files accessed by the CRDROP\_INDEX code are summarized in Table 9.8.

**Table 9.8** Files Accessed by the CRDROP\_INDEX Code

File	Use	Description
CRDROP_INDEX keyword file	Input	Primary source of run control information.
ESD keyword file	Input	Source of general information for the overall analysis, including location coordinates. The path name from this file identifies the base directory for the analysis.
ECDA Map Index File	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 report file	Output	Summary output information for the CRDROP_INDEX analysis, including error messages, if any.
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.

### 9.5.5.1 Input Files

The purpose of files providing input to the CRDROP\_INDEX code are described as follows:

- **ECDA Map Index File.** The ECDA map index file is read to provide the impact location names. The location names are used to extract the location coordinates from the ESD file. The format of this file is described in Section 2.2.2.
- **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 END, FILE, LOCATION, and OS keywords. The syntax for these keywords are defined in Section 2.1.
- **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. These keywords are described in Section 9.5.4 and an example keyword file is provided in Table 9.9.

**Table 9.9 Example Keyword File for CRDROP\_INDEX**

```
! Keywords for INDEX_CRDROP to generate location cross index values
!  
REPORT "crdrop_index.rpt"  
!  
TITLE "CA 2004 Revised Grid - Keyword file for CRDROP_INDEX"  
USER "Paul W. Eslinger"  
!  
EXECUTE  
!  
! Maximum allowed value for the minimum distance between locations  
DISTANCE 10000.  
!  
! 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
```

- **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 9.10

provides the first few lines of an example of a river model grid location file. The files typically contain several thousand records.

**Table 9.10** 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

### 9.5.5.2 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 9.11).

**Table 9.11** Example Report file from CRDROP\_INDEX

CCCC	RRRR	DDDD	RRRR	OOO	PPPP	III	N	N	DDDD	EEEE	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	NN	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	DDDD			R	R	OOO		P	=====	III	N	N	DDDD		EEEE	X	X

CRDROP_INDEX	Version 1.00.001
Last Modified on 5 Sep 2004	
-----	
Columbia River Data Restructure Other Programs	
Location Cross-Index Generation	
SAC Rev. 1 Data Processing	
-----	
Developed By Battelle Memorial Institute	
Pacific Northwest National Laboratory	
Richland, Washington	
-----	

Current Run ID = 20040905192301	User Name = Paul W. Eslinger
System Date = 09-05-2004	System Time = 19:23:01

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: CA 2004 Revised Grid - Keyword file for CRDROP_INDEX  
User: Paul W. Eslinger  
Operating System: UNIX, Delimiter is: /  
  
Impact parameters are set as follows:  
ECDA MAPFILE: /home/ANALYSIS3/Confirm-median/ecda/ECDA_Confirm-median.map  
Number of Location:4351  
  
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 impact locations. The file has the information and format indicated in Table 9.3. Table 9.4 provides an example cross-index file. The name of the output file is specified using the FILE keyword in the file index.key.

**Temporary Run File.** A file named “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 ESP program. The normal operating mode is to run the CRDROP\_INDEX code in a stand-alone mode 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 named “crdrop\_index.done” is written; otherwise the file named “crdrop\_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.

## **10.0 RIPSAC – Riparian Zone Module**

### **10.1 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 ECDA files (see Section 2.2.1), calculates the concentrations for seep water and soil, and writes the calculated concentrations back into the same files.

### **10.2 Algorithms and Assumptions**

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:

$$C_{\text{seep}} = (Df_{\text{seep}} C_{\text{gw}}) + (1 - Df_{\text{seep}}) C_{\text{river}}$$

where

- $C_{\text{seep}}$  = Concentration of an analyte in seep water (Ci/m<sup>3</sup> or Kg/m<sup>3</sup>)
- $C_{\text{gw}}$  = Concentration of an analyte in ground water (Ci/m<sup>3</sup> or Kg/m<sup>3</sup>)
- $C_{\text{river}}$  = Concentration of an analyte in river (surface) water (Ci/m<sup>3</sup> or Kg/m<sup>3</sup>)
- $Df_{\text{seep}}$  = 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:

$$C_{\text{soil}} = C_{\text{seep}} Kd_{\text{soil}}$$

where

- $C_{\text{soil}}$  = Concentration of an analyte in soil (Ci/kg soil or kg/kg soil)
- $C_{\text{seep}}$  = Concentration of an analyte in seep water (Ci/m<sup>3</sup> or kg/m<sup>3</sup>)
- $Kd_{\text{soil}}$  = Partition coefficient for the analyte between water and soil (m<sup>3</sup>/kg)

### **10.3 Code Execution Environment**

RIPSAC was developed as a program to be invoked by the master SAC control software, SAC ESP. However, it can be invoked in a stand-alone mode.

#### **10.3.1 Location in the Processing Sequence**

The general processing sequence for SAC environmental runs is outlined in Figure 1.1. As noted in the figure, RIPSAC is the next to 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 all analytes. RIPSAC may also be run on a subset of data for any combination of realizations and analytes.

### 10.3.2 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-1.exe "keyword file"  
ripsac-1.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.

### 10.4 Keyword Descriptions for RIPSAC

In general, the keywords for RIPSAC (Table 10.1) 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.

**Table 10.1** Summary of Keywords Used by RIPSAC

<b>Keyword</b>	<b>Description</b>
ANALYTE	The ANALYTE keyword defines the analytes to be used in a run of RIPSAC.
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.
EXECUTE	The EXECUTE keyword indicates the problem will be executed in addition to performing input consistency checks.
FILE	The FILE keyword is used to enter the name of the ESD keyword file.
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 a run of RIPSAC.

<b>Keyword</b>	<b>Description</b>
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.

#### **10.4.1 ANALYTE Keyword for RIPSAC**

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

```
ANALYTE ["quote1"] {"quote2"} ... {"quoteN"}
```

One or more analyte ID's are entered in quote strings. The analytes ID's entered must be a subset of the analytes defined in the ESD keyword file. Multiple ANALYTE keywords may be entered and multiple analytes can be processed in a single run of the code. Two example keywords are the following:

```
ANALYTE "H3"  
ANALYTE "C14" C136" Se79" Sr90"
```

#### **10.4.2 DEBUG Keyword for RIPSAC**

The DEBUG keyword is used to activate dumping of intermediate calculations to the report file. It should be used only on small problem definitions with only one or two realizations, otherwise the volume of output can easily exceed 2 gigabytes. The following is this keyword's syntax:

```
DEBUG
```

#### **10.4.3 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 following is this keyword's syntax:

```
END
```

#### **10.4.4 FILE Keyword for RIPSAC**

The FILE keyword is used to enter the name of the ESD keyword file for the assessment. The names of the files providing concentration data are contained in the ESD keyword file rather than in the RIPSAC keyword file. The following is this keyword's syntax:

```
FILE [ESD="quote1"]
```

The file name must be entered in a quote string,. Path names up to 200 characters long are supported. An example entry is the following:

```
FILE ESD "G:\RIPSAC\Tests\Test_Case_12.Key"
```

### **10.4.5 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 location and analyte. The following is this keyword's syntax:

```
KDSOIL [ANALYTE="quote1"] [KDSOIL="quote2"] [LOCATION="quote2"  
 {"quote4"} ... {"quoteN"} | TERSE]
```

Multiple KDSOIL keyword entries may be required if there is more than one analyte and more than one riparian zone location. The KDSOIL keywords must cover every combination of analyte and riparian zone locations where soil concentrations are to be computed.

The quote string associated with the ANALYTE modifier must contain the ID of an analyte identified in the ESD keyword file (see Section 2.1.1). The quote string associated with the KDSOIL modifier must contain the ID string of KDSOIL variable identified in the ESD keyword file (see Section 2.1.16).

Only one of the TERSE and LOCATION modifiers can be used. If the TERSE modifier is entered then the same KDSOIL definition will be used for all locations. If the LOCATION modifier is used it must be associated with one or more quote strings containing location ID's for riparian zone locations that were defined in the ESD keyword file.

Example keyword entries include the following:

```
KDSOIL TERSE ANALYTE="H3"      KDSOIL="KDH"  
KDSOIL TERSE ANALYTE="C14"     KDSOIL="KDC"  
KDSOIL ANALYTE="Sr90" KDSOIL="KDSr" LOCATION= "RFP001" "RFP002" "RFP003"
```

### **10.4.6 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 ESD keyword file. The following is this keyword's syntax:

```
LOCATION [PRIMARY="quote1"] [SECOND="quote2"] [DF="quote3"]
```

There must be a LOCATION keyword entry for every location where seep water or soil concentrations are to be computed. The quote string associated with the PRIMARY modifier must contain the ID string of a riparian location defined in the ESD keyword (see Section 2.1.17). This location must have the groundwater concentration solution activated. The quote string associated with the SECOND modifier must contain the ID string of a location where a surface water concentration solution has been identified in the ESD keyword file. The quote string associated with the DF modifier must contain the ID string of a dilution variable (see Section 2.1.6) identified in the ESD keyword file.



Example keyword entries include the following:

```
LOCATION PRIMARY="RHP001" SECOND="QHP001" DF="DF5m"
LOCATION PRIMARY="RHP002" SECOND="QHP002" DF="DF5m"
LOCATION PRIMARY="RHP003" SECOND="QHP003" DF="DF5m"
LOCATION PRIMARY="RHP004" SECOND="QHP004" DF="DF5m"
LOCATION PRIMARY="RHP005" SECOND="QHP005" DF="DF5m"
```

### 10.4.7 REALIZATION Keyword for RIPSAC

The REALIZATION keyword is used to define the realizations to be processed in the RIPSAC code. The following is this keyword's syntax:

```
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 10.2 describes the modifiers used on the REALIZATION keyword. Only one of the ALL, LIST or RANGE 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 10.2** Modifiers for 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
```

### **10.4.8 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 following is this keyword's syntax:

```
REPORT [ "quote" ]
```

The name of the report file is entered in a quote string. File names up to 200 characters long are supported, and path names can be included. An example REPORT keyword is the following:

```
REPORT "/SAC/SystemCodes/Test.rpt"
```

### **10.4.9 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 following is this keyword's syntax:

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

The title is entered in a quote string. 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 RIPSAC code."
```

### **10.4.10 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 following is this keyword's syntax:

```
USER "John Q. Public"
```

## **10.5 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.

### **10.5.1 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.

### 10.5.1.1 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 are required:

- ANALYTE – Defines analytes in the assessment.
- END – End of the ESD keyword data.
- FILE – File names for ECDA-related files.
- LOCATION – Locations defined in the assessment.
- REALIZAT – Number of realizations to be simulated.
- TIMES – Times at which ECDA data are saved for the the impacts codes.
- TITLE – Title line from the ESD keyword file.

### 10.5.1.2 RIPSAC Keyword File

Section 10.4 describes the individual keywords for RIPSAC. Table 10.3 provides an example RIPSAC keyword file for two analytes and four riparian locations using the TERSE option on the KDSOIL keyword..

**Table 10.3** Example RIPSAC Keyword File

```
REPORT "RipSac_CA1_median.Rpt"
FILE ESD "/home/ANALYSIS4/CA1_median/ESD_CA1_median.key"
TITLE "2004 CA - RIPSAC keyword file"
USER "Carmen Arimescu"
REALIZATION ALL
ANALYTE ID="H3"
ANALYTE ID="C14"
EXECUTE
KDSOIL TERSE ANALYTE="H3"      KDSOIL="KDH"
KDSOIL TERSE ANALYTE="C14"    KDSOIL="KDC"
! PRIMARY location is for groundwater on the shore - Hanford
! SECOND location is for the associated surface water - Hanford
! Riparian and Aquatic - Hanford Shore
LOCATION PRIMARY="RHP001" SECOND="QHP001" DF="DF5m"
LOCATION PRIMARY="RHP002" SECOND="QHP002" DF="DF5m"
LOCATION PRIMARY="RHP003" SECOND="QHP003" DF="DF5m"
LOCATION PRIMARY="RHP004" SECOND="QHP004" DF="DF5m"
END
```

### 10.5.1.3 DILUTE Stochastic Data

The file containing dilution factors for groundwater and river water interaction coefficients is used for every run of the code. 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 provides an example file. The file name is supplied in the ESD keyword file by using the FILE keyword and the DILUTE modifier. Typically, the user of the RIPSAC code will not modify this file.

#### 10.5.1.4 KDSOIL Stochastic Data

The file containing stochastic values for the soil-water partition coefficients is used for every run of the code. 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 provides an example file. The file name is supplied in the ESD keyword file by using the FILE keyword and the KDSOIL modifier. Typically, the user of the RIPSAC code will not modify this file.

#### 10.5.1.5 Concentration Data Files

Concentration data for different analytes are contained in separate ECDA files. RIPSAC reads from, and writes to, a concentration data file for every analyte being simulated. If there are ten analytes, then ten concentration data files will be required, along with one record map file. Sections 2.2.1 describes the format for each of these files. The ECDA data and map file names are defined in the ESD keyword file.

### 10.5.2 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.

#### 10.5.2.1 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 10.4 provides excerpted records from an example RIPSAC report file.

**Table 10.4** 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 2.00.A.1					
Last Modified on 17 Jun 2003					
Stochastic River-Shore Analysis					
Systems Assessment Capability (SAC), Revision 0					
-----					
Developed By Battelle					
Pacific Northwest National Laboratory					
Richland, Washington					
-----					

*User Instructions for the Systems Assessment Capability, Rev. 0, Computer Codes*  
*Volume 1: Inventory, Release, and Transport Modules*

---

Current Run ID = 200407121935    User Name = Carmen Arimescu  
System Date = 07-12-2004    System Time = 19:35:06

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: 2004 CA - RIPSAC keyword file  
User: Carmen Arimescu  
1 Realizations identified in the ESD keyword file  
1 Realizations utilized in this run

A map of realization activations (true/false) is:  
1- 1 T

File Name for RIPSAC Input Keyword Data  
File: RipSac\_CA1\_median.key

File Name for ESD Input Keyword Data  
File: /home/ANALYSIS4/CA1\_median/ESD\_CA1\_median.key

File Name for Dilution Data  
File: /home/ANALYSIS4/CA1\_median/ecda/DILUTE\_CA1\_median.Dat

File Name for KDSOIL Data  
File: /home/ANALYSIS4/CA1\_median/ecda/KDSOIL\_CA1\_median.dat

File Name for Concentration Index Map File  
File: /home/ANALYSIS4/CA1\_median/ecda/ECDA\_CA1\_median.map

File Name for Media Concentrations: Analyte ID="C14"  
File: /home/ANALYSIS4/CA1\_median/ecda/C14\_CA1\_median.dat

<data rows deleted>

Analyte Information for 15 analytes.

1 : C14 : Carbon-14

<data rows deleted>

15 : U238 : Uranium-238

A total of 12 analytes have been requested.

Number of times is 272

Index    Year

1 1945

2 1950

<data rows deleted>

```
271 12000
272 12050

Number of locations is 4351
Index   Location ID
      1 UH0001
      2 UH0002
<data rows deleted>
4350 QTV010
4351 QHRW11

Soil-Water KD and Dilution Factor Mapping
Location      Analyte      KDSOIL ID String      DILUTE ID String
-----
2864: RFP001   1: C14      KDC                    DF5m
2864: RFP001   2: Cs137    KDCs                   DF5m
<data rows deleted>
4057: R21I08   14: U235    KDU                    DF5m
4057: R21I08   15: U238    KDU                    DF5m

===== End of the Problem Definition =====

Message originating in routine PROCESS_IMPACTS
Message: Entering subroutine for analyte "C14"
      On 07/12/2004 at 19:35:16.973

<data rows deleted>

Message originating in routine PROCESS_IMPACTS
Message: Entering subroutine for analyte "U238"
      On 07/12/2004 at 19:55:06.954

Message originating in routine RipSac
Message: Normal Termination
      Run Completed on 07/12/2004 at 19:56:56.033
```

### 10.5.2.2 RIPSAC.Run File

A file named "RipSac.Run" is always written just as RIPSAC starts execution. Presence of the file signals the 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.

### 10.5.2.3 RIPSAC.Done File

A file named "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 10.4) will contain more detailed information about the type of error encountered.

#### **10.5.2.4 Concentration Data Files**

Concentration data for different analytes are contained in separate ECDA files. RIPSAC reads from, and writes to, a concentration data file for every analyte being simulated. If there are ten analytes, then ten concentration data files will be required, along with one record map file. Section 2.2.1 describes the format for each of these files. The ECDA data and map file names are defined in the ESD keyword file using the FILE keyword.

## 11.0 SOIL – Soil Accumulation Module

### 11.1 Purpose

The soil accumulation module (SOIL) calculates the concentrations of analytes in near-surface soil. The soil model algorithms apply to dry land locations as well as locations that use irrigation. SOIL reads air deposition rates and analyte concentrations in groundwater and surface water from ECDA files (see Section 2.2), calculates the concentrations in soil, and writes the calculated concentrations back into the same data files.

### 11.2 Algorithms and Assumptions

The soil concentration model includes the effects of air deposition, precipitation and evaporation, irrigation with contaminated water, leaching from soil due to precipitation and irrigation, and radioactive decay. These processes are modeled in the upper soil layer, which is nominally assumed to be 15 cm in depth. The general equation for the rate of analyte accrual in the upper soil layer is the following:

$$M'(t) = I(t) \times 0.01 \times C_{\text{water}}(t) + A_{\text{dep}}(t) - M(t) \times (\lambda_L(t) + \lambda_{\text{Rad}})$$

where

- $M'(t)$  = rate of change in the amount of analyte in the upper soil layer [Ci/(m<sup>2</sup>-day) or kg/(m<sup>2</sup>-day)]
- $M(t)$  = amount of analyte in the upper soil layer (Ci/m<sup>2</sup> or kg/m<sup>2</sup>)
- $I(t)$  = irrigation rate (cm/day), set to zero outside the growing season
- 0.01 = conversion factor (m/cm)
- $C_{\text{water}}(t)$  = concentration of analyte in the irrigation water (Ci/m<sup>3</sup> or kg/m<sup>3</sup>)
- $A_{\text{dep}}(t)$  = air deposition of the analyte [Ci/(m<sup>2</sup>-day) or kg/(m<sup>2</sup>-day)]
- $M(t)$  = amount of analyte in the upper soil layer (Ci/m<sup>2</sup> or kg/m<sup>2</sup>)
- $\lambda_L(t)$  = leach rate constant due to water moving through the upper soil layer (day<sup>-1</sup>)
- $\lambda_{\text{Rad}}$  = radioactive decay constant (day<sup>-1</sup>) (set to zero for nonradioactive analytes)

Baes and Sharp (1981) provide a method for estimating the annual leach rate of contaminants out of the upper layer of irrigated soils. After adapting their model to use daily units, the leach rate is represented by:

$$\lambda_L(t) = \frac{P + I(t) - E}{d_s \theta \left(1 + \frac{\rho}{\theta} K_d\right)}$$

where

- $\lambda_L(t)$  = leach rate (day<sup>-1</sup>)
- P = annual average precipitation rate (cm/day)
- $I(t)$  = annual average irrigation rate (cm/day), set to zero outside the growing season
- E = annual average evapotranspiration rate (cm/day)



- $d_s$  = depth of surface soil (cm), nominally 15 cm
- $\theta$  = surface soil volumetric water content ( $\text{mL}/\text{cm}^3$ ), nominally  $0.5 \text{ mL}/\text{cm}^3$
- $\rho$  = surface soil density ( $\text{g}/\text{cm}^3$ ), nominally  $1.5 \text{ g}/\text{cm}^3$
- $K_d$  = soil distribution coefficient ( $\text{mL}/\text{g}$ )

The soil concentration is then computed from:

$$C_{\text{soil}}(t) = M(t) / S_{\text{mass}}$$

where

- $C_{\text{soil}}(t)$  = soil concentration ( $\text{Ci}/\text{kg}$  or  $\text{kg-analyte}/\text{kg-soil}$ )
- $M(t)$  = amount of analyte in the upper soil layer ( $\text{Ci}/\text{m}^2$  or  $\text{kg}/\text{m}^2$ )
- $S_{\text{mass}}$  = mass of soil ( $\text{kg}/\text{m}^2$ ) in the upper soil layer, a nominal value is  $225 \text{ kg}/\text{m}^2$

The soil concentration varies as a function of time due to the seasonal effects of irrigation, and never reaches a long-term equilibrium. However, the solution desired for SAC is an annual soil concentration. This concentration is computed as the average of the daily concentrations during the irrigation season (nominally 120 to 150 days in length). This averaged value reaches an equilibrium value under some conditions.

### **11.3 Code Execution Environment**

SOIL was developed as a program to be invoked by the master SAC control software, ESP. However, it can also be invoked in a standalone mode.

#### **11.3.1 Location in the Processing Sequence**

The general processing sequence for SAC environmental runs is outlined in Figure 1.1. As noted in the figure, SOIL is the last step in the environmental transport sequence and operates using internal realization looping. Thus, only one run of SOIL is required to process all realizations for all analytes. SOIL also has the capability to run on a subset of data for any combination of realizations and analytes.

#### **11.3.2 How the Code Is Invoked**

SOIL 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 SOIL is executed by either of the following DOS commands:

```
SOIL-1 "Keyword File"  
SOIL-1
```

In the Linux environment SOIL is executed through either of the following Bourne Shell or C Shell commands:

```
soil-1.exe "keyword file"
soil-1.exe
```

In the first example for both operating systems, SOIL is invoked and assumes the file identified by "keyword file" (file names are case sensitive under Linux) contains the controlling keywords for SOIL. In the second example, the keyword file name is missing from the command line, thus SOIL queries the user for the name of the file. In either case, if the input keyword file is not found, SOIL writes an error message and terminates execution.

## 11.4 Keyword Descriptions for SOIL

In general, the control keywords specific to the SOIL code can be entered in any order. A summary of the keywords is provided in Table 11.1. 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.

**Table 11.1** Summary of Keywords Used by SOIL

Keyword	Description
ANALYTE	The ANALYTE keyword defines the analytes to be used in a run of SOIL.
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.
EXECUTE	The EXECUTE keyword controls optional execution after the control inputs are checked for consistency.
FILE	The FILE keyword is used to enter the name of the ESD keyword file.
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 assign data to locations for use in the soil concentration calculations.
REALIZATION	The REALIZATION keyword is used to define the realizations to be processed in the SOIL code.
REPORT	The REPORT keyword is used to define the name of the output report (log) file.
SOIL	The SOIL keyword is an optional keyword used to define the depth of the surface soil layer and the density of the soil.
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.

### 11.4.1 ANALYTE Keyword for SOIL

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

```
ANALYTE ["quote 1"] {"quote 2"} ... {"quote n"}
```

Analytes are requested by entering their ID as a quote string. The analytes requested must be defined in the ESD keyword file. A separate ANALYTE keyword may be entered for every analyte or multiple analytes can be defined on the same keyword. The same analyte ID string must not be entered more than once. Example entries defining for analytes are the following:

```
ANALYTE "Cs137"  
ANALYTE "Np237"
```

The same definition can be obtained by entering the following single keyword entry:

```
ANALYTE "Cs137" "Np237"
```

### 11.4.2 DEBUG Keyword for SOIL

The DEBUG keyword is used to activate dumping of intermediate calculations to the report file. It should be used only on small problem definitions with only one or two realizations, otherwise the volume of output can easily exceed 2 gigabytes. The following is this keyword's syntax:

```
DEBUG {LOOP} {CONCENTR} {COMPUTE}
```

The modifiers associated with the DEBUG keyword are defined in Table 11.2.

**Table 11.2** Modifiers for the DEBUG Keyword in SOIL

Modifier	Description
LOOP	The LOOP modifier initiates reporting to the screen that shows program looping and control progress.
CONCENTR	The CONCENTR modifier initiates reporting all media concentrations to the report file.
COMPUTE	The COMPUTE modifier initiates writing detailed values from the leaching and concentration calculations to the report file.

### 11.4.3 END Keyword for SOIL

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 following is this keyword's syntax:

```
END
```

#### **11.4.4 EXECUTE Keyword for SOIL**

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. The EXECUTE keyword must be present when SOIL is exercised under the control of the ESP or the analysis will not proceed correctly. The following is this keyword's syntax:

```
EXECUTE
```

#### **11.4.5 FILE Keyword for SOIL**

The FILE keyword is used to enter the name of the ESD keyword file that defines the assessment. The names of the files providing concentration data are contained in the ESD keyword file rather than in the SOIL keyword file. The following is this keyword's syntax:

```
FILE [ESD="quote1"]
```

The file name must be entered in a quote string. Path names up to 200 characters long are supported. An example entry is the following:

```
FILE ESD "G:\SOIL\Tests\Test_12.Key"
```

#### **11.4.6 KDSOIL Keyword for SOIL**

The KDSOIL keyword is used to assign analyte-specific soil-water partition coefficients to locations for use in the soil concentration calculations. The actual numerical values are extracted from a library of values defined in the ESD keyword file (see Section 2.1.16). The KDSOIL keywords in SOIL are assigned as a function of location and analyte. The following is this keyword's syntax:

```
KDSOIL [ANALYTE="quote1"] [KDSOIL="quote2"]  
[LOCATION="quote3" {"quote4"} ... {"quote5"} | TERSE]
```

Multiple KDSOIL keyword entries may be required if there is more than one analyte and more than one soil concentration location. Subsequent KDSOIL entries build upon the definition of previous entries. A KDSOIL keyword entry may be used for every combination of analyte and location where soil concentrations are to be computed. Alternately, multiple locations can be defined on a single keyword. The modifiers for the KDSOIL keyword can be entered in any order. The modifiers used with the KDSOIL keyword are defined in Table 11.3.

**Table 11.3** Modifiers for the KDSOIL Keyword in SOIL

Modifier	Description
ANALYTE	The quote string associated with the ANALYTE modifier must contain the ID of an analyte identified in the ESD keyword file.
KDSOIL	The quote string associated with the KDSOIL modifier must contain the ID string of a KDSOIL variable identified in the ESD keyword file.
LOCATION	One or more quote strings are associated with the LOCATION modifier. Each quote string must contain the ID of a location identified in the ESD keyword file. The LOCATION modifier is not used if the TERSE modifier is used.
TERSE	This modifier can be used instead of the LOCATION modifier. The effect is to assign the same KDSOIL definition for this analyte to every location where a soil concentration solution is to be computed.

Example keyword entries include the following:

```
KDSOIL ANALYTE="U234" LOCATION="HL0012" KDSOIL="KD#042"
KDSOIL ANALYTE="Cr" LOCATION="HL0012" KDSOIL="KD#023"
KDSOIL ANALYTE="Cr" LOCATION="HL0014" KDSOIL="KD#023"
KDSOIL ANALYTE="Cr" LOCATION="HL0015" KDSOIL="KD#023"
KDSOIL ANALYTE="Cr" LOCATION="HL0021" KDSOIL="KD#023"
```

The information using the last four keywords above could be entered using the single following keyword:

```
KDSOIL ANALYTE="Cr" LOCATION="HL0021" "HL0015" "HL0014" "HL0012"
KDSOIL="KD#023"
```

The following examples assign the same Kd to every location for each the analyte:

```
KDSOIL TERSE ANALYTE="H3" KDSOIL="KDH"
KDSOIL TERSE ANALYTE="C14" KDSOIL="KDC"
KDSOIL TERSE ANALYTE="C136" KDSOIL="KDC1"
KDSOIL TERSE ANALYTE="Se79" KDSOIL="KDSe"
KDSOIL TERSE ANALYTE="Sr90" KDSOIL="KDSr"
```

### 11.4.7 LOCATION Keyword for SOIL

The LOCATION keyword is used to identify groundwater and surface water concentrations and infiltration values in order to complete soil concentration calculations. The infiltration rates are extracted from a library of values that are defined in the ESD keyword file file (see Section 2.1.16). The following is this keyword's syntax:

```
LOCATION [ID="quote 1"] [INFILT="quote 2"]
```

Multiple LOCATION keyword entries may be required. There must be a LOCATION keyword entry for every location where soil concentrations are to be computed. The modifiers for this keyword can be entered in any order. The modifiers for the LOCATION keyword are described in Table 11.4.

**Table 11.4** Modifiers for the LOCATION Keyword in SOIL

Modifier	Description
ID	The quote string associated with the ID modifier must contain the ID string of a location identified in the ESD keyword file. The presence of the LOCATION keyword with this ID signifies that one or more soil concentration calculations will occur at this location.
INFILT	The quote string associated with the INFILT modifier must contain the ID string of an infiltration variable identified in the infiltration data library. This variable defines the infiltration

The following examples illustrate the use of the LOCATION keyword:

```
LOCATION ID="UH0001" INFILT="Wa-s"
LOCATION ID="UH0002" INFILT="Ba-s"
LOCATION ID="UH0003" INFILT="Ba-s"
LOCATION ID="UH0004" INFILT="Ba-s"
```

#### 11.4.8 REALIZATION Keyword for SOIL

The REALIZATION keyword is used to define the realizations to be processed in the SOIL code. The following is this keyword's syntax:

```
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 assessment. The SOIL code can process all realizations at one time or it can process a subset of the realizations. Table 11.5 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 11.5** Modifiers for the REALIZATION Keyword in SOIL

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 will invoke processing for all realizations defined in the ESD keyword file.

```
REALIZATION ALL
```

The following set of two REALIZATION keywords assume that at least 100 realizations are defined in the ESD keyword file. These keywords define processing in SOIL for realization numbers 2, 4, 6, 12, 13, 14, 15, 99, and 100.

```
REALIZATION LIST 2 4 6 100 99  
REALIZATION RANGE 12 15
```

#### **11.4.9 REPORT Keyword for SOIL**

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 following is this keyword's syntax:

```
REPORT [ "quote" ]
```

The name of the report file is entered in a quote string. File names up to 200 characters long are supported, and path names can be included. An example REPORT keyword record is the following:

```
REPORT "/SAC/SystemCodes/Cultural/Test1.rpt"
```

#### **11.4.10 SOIL Keyword for SOIL**

The SOIL keyword is an optional keyword used to define the depth of the surface soil layer and the density of the soil. If this keyword is not entered the soil depth will be 15 cm and the soil density will be 1.5 g/cm<sup>3</sup>. The following is this keyword's syntax:

```
SOIL [DEPTH=N1] [DENSITY=N2]
```

The numerical value associated with the DEPTH modifier gives the soil depth (centimeters) for the upper soil layer. If the DEPTH modifier is missing a value of 15 cm is used. The numerical value associated with the DENSITY modifier gives the soil density (g/cm<sup>3</sup>) for the upper soil layer. If the DENSITY modifier is missing a value of 1.5 g/cm<sup>3</sup> is used. The values for DEPTH and DENSITY are the same for all locations.

#### **11.4.11 TITLE Keyword for SOIL**

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 following is this keyword's syntax:

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

The title is entered in a quote string. Titles up to 200 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."
```

### **11.4.12 USER Keyword for SOIL**

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 following is this keyword's syntax:

```
USER ["quote"]
```

The user name is entered in a quote string. 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"
```

## **11.5 Data Files**

The SOIL 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.

### **11.5.1 Input Files**

The input files for SOIL are two keyword files, a KDSOIL data file, a INFILT data file, and a suite of ECDA files. The SOIL keyword file controls the case SOIL 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 SOIL only needs to identify the ESD keyword file. The input files are the following:

### **11.5.2 ESD Keyword File**

The SOIL code reads a number of keywords from the ESD keyword file. A summary of these keywords is provided in Table 11.6. Typically, the user of the SOIL code will not modify the ESD keyword file. These keywords are described in detail in section 2.1.

**Table 11.6** ESD Keywords Used by SOIL

<b>Keyword</b>	<b>Description</b>
ANALYTE	The ANALYTE keyword defines the analytes to be used in a run of SOIL.
END	The END keyword signifies the end of all keyword data in the ESD keyword file.
FILE	The FILE keyword is used to define the names of the ECDA-related files.
IRRIGATE	The IRRIGATE keyword defines the year to start irrigating and the irrigation rate.
LOCATION	The LOCATION keyword defines the impact locations.
PERIOD	The period keyword defines the time period for the entire analysis.



REALIZATION	The REALIZATION keyword defines the number of realizations in the entire analysis.
TIMES	The TIMES keyword defines the times (set of years) for which impact analyses can be calculated.
TITLE	The TITLE keyword defines a single-line problem title.

### 11.5.3 SOIL Keyword File

Section 11.4 describes the individual keywords for SOIL. Table 11.7 provides an example SOIL keyword file for nine analytes. This example file uses four locations for impacts calculations. A SOIL keyword file for a production run incorporating hundreds of locations for several analytes would be several thousand lines long.

**Table 11.7** Example SOIL Keyword File

```

REPORT "Soil_CA1_median.rpt"
FILE ESD "/home/ANALYSIS4/CA1_median/ESD_CA1_median.key"
USER "Paul W. Eslinger"
TITLE "2004 CA - SOIL keyword file - 10,000-year median inputs case"
REALIZAT ALL
EXECUTE
!
ANALYTE ID="H3"
ANALYTE ID="C14"
ANALYTE ID="C136"
ANALYTE ID="Se79"
ANALYTE ID="Sr90"
ANALYTE ID="Tc99"
!
KDSOIL TERSE ANALYTE="H3"      KDSOIL="KDH"
KDSOIL TERSE ANALYTE="C14"     KDSOIL="KDC"
KDSOIL TERSE ANALYTE="C136"   KDSOIL="KDC1"
KDSOIL TERSE ANALYTE="Se79"   KDSOIL="KDSe"
KDSOIL TERSE ANALYTE="Sr90"   KDSOIL="KDSr"
KDSOIL TERSE ANALYTE="Tc99"   KDSOIL="KDTc"
KDSOIL TERSE ANALYTE="I129"   KDSOIL="KDI"
!
LOCATION ID="UH0001" INFILT="Wa-s"
LOCATION ID="UH0002" INFILT="Ba-s"
LOCATION ID="UH0003" INFILT="Ba-s"
LOCATION ID="UH0004" INFILT="Ba-s"
!
END

```

### 11.5.4 KDSOIL Stochastic Data

The file containing stochastic values for the soil-water partition coefficients is used for every run of the code. 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 provides an example file. The file name is supplied in the ESD keyword file by using the FILE keyword and the KDSOIL modifier. Typically, the user of the SOIL code will not modify this file.

### 11.5.5 INFILT Stochastic Data

The file containing stochastic values for the water infiltration data must be supplied for every run of the code. This file is generated by the ESP program from INFILTRATION keywords contained in the ESD keyword file. Section 2.3.3 defines the file format and provides an example file. The file name is supplied in the ESD keyword file by using the FILE keyword and the INFILT modifier. Typically, the user of the SOIL code will not modify this file.

### 11.5.6 ECDA Files

Concentration data for different analytes are contained in separate ECDA files. SOIL reads from, and writes to, a concentration data file for every analyte being simulated. If there are ten analytes, then ten concentration data files will be required, along with one record map file. Sections 2.2.1 describes the format for each of these files. The ECDA data and map file names are defined in the ESD keyword file.

## 11.6 Output Files

SOIL writes a report file, a run-time signal file named “soil.run”, and a run completion signal file named “soil.done” as well as modifying ECDA 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.

### 11.6.1 SOIL Report File

The SOIL 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 11.8 provides excerpts from a SOIL report file. The original files was 6100 lines long.

**Table 11.8** Excerpts from a SOIL Report File

<pre>      SSS      OOOOO      IIIIIIII  L S   SS  O    O    I    L S   S   O    O    I    L       SS      O    O    I    L S   S   O    O    I    L SS  S   O    O    I    L       SSS      OOOOO      IIIIIIII  LLLLLLL                  Soil 1.00.D.1                 Last Modified on 29 Dec 2003                  Stochastic Soil Concentration Analysis</pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

*User Instructions for the Systems Assessment Capability, Rev. 0, Computer Codes*  
*Volume 1: Inventory, Release, and Transport Modules*

---

Systems Assessment Capability (SAC), Revision 1  
-----

Developed By Battelle  
Pacific Northwest National Laboratory  
Richland, Washington  
-----

Current Run ID = 20040824101440    User Name = Paul W. Eslinger  
System Date = 08-24-2004    System Time = 10:14:40.653

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: 2004 CA - SOIL keyword file - 10,000-year median inputs case

User: Paul W. Eslinger

1 Realizations identified in the ESD keyword file

1 Realizations utilized in this run

A map of realization activations (true/false) is:

1- 1 T

Year definitionsd

1944 : Start year for the simulation

12050 : Stop year for the simulation

2070 : Site closure year for the simulation

2004 : Start year for irrigation

File Name for SOIL Input Keyword Data

File: Soil\_CA1\_median.key

File Name for ESD Input Keyword Data

File: /home/ANALYSIS4/CA1\_median/ESD\_CA1\_median.key

File Name for Infiltration Data

File: /home/ANALYSIS4/CA1\_median/ecda/infilt.dat

File Name for KDSOIL Data

File: /home/ANALYSIS4/CA1\_median/ecda/KDSOIL\_CA1\_median.dat

File Name for Concentration Map File

File: /home/ANALYSIS4/CA1\_median/ecda/ECDA\_CA1\_median.map

File Name for Media Concentrations: Analyte ID="H3"

File: /home/ANALYSIS4/CA1\_median/ecda/H3\_CA1\_median.dat

Analyte Information for 15 analytes.

```

1 : C14      : ==> Not used
2 : Cs137   : ==> Not used
3 : Cl36    : ==> Not used
4 : I129    : ==> Not used
5 : Np237   : ==> Not used
6 : Pa231   : ==> Not used
7 : Ra226   : ==> Not used
8 : Se79    : ==> Not used
9 : Sr90    : ==> Not used
10 : Tc99   : ==> Not used
11 : H3     : Tritium
12 : Eu152  : ==> Not used
13 : U233   : ==> Not used
14 : U235   : ==> Not used
15 : U238   : ==> Not used
A total of 1 analytes have been requested.

Number of times is 274
  Index  Year
    1    1945
    2    1950
    3    1955
    4    1960
    5    1965
                                     <deleted lines>

Total number of locations is 4351
Activated locations are shown below
  Index  ID      SW Irrig  Name
-----
  1 : "UH0001" : "QHP025" : "UnsuitableForAgricul"
  2 : "UH0002" : "QHP025" : "UnsuitableForAgricul"
  3 : "UH0003" : "QHP025" : "UnsuitableForAgricul"
  4 : "UH0004" : "QHP025" : "AgExclusionBuffer"
  5 : "UH0005" : "QHP025" : "AgExclusionBuffer"
  6 : "UH0006" : "QHP025" : "AgExclusionBuffer"
  7 : "UH0007" : "QHP025" : "AgExclusionBuffer"
  8 : "UH0008" : "QHP025" : "Upland"
  9 : "UH0009" : "QHP025" : "AgExclusionBuffer"
                                     <deleted lines>

Soil-Water KD and Dilution Factor Mapping
  Location      Analyte      KDSOIL ID String      INFILT ID String
-----
  1: "UH0001"  11: "H3"      " "KDH"              " "Wa-s"            "
  2: "UH0002"  11: "H3"      " "KDH"              " "Ba-s"            "
  3: "UH0003"  11: "H3"      " "KDH"              " "Ba-s"            "
                                     <deleted lines>

===== End of the Problem Definition =====

Header information from the map file
  ECDA_PTITLE = 2004 Composite Analysis 10,000-year (Median Inputs) Assessment
  ECDA_PRGNAM = ECDA
  ECDA_PRGVER = 2.00.A.4

```

```
ECDA_PRGDAT = 16 Oct 2003
ECDA_USRNAM = Eslinger-DWE-WEN
ECDA_CRUNID = 20040802132252
ECDA_BLOCK  = 23155
ECDA_RECLen = 18
ECDA_NREAL  = 1
Date: 08-24-2004 Time: 10:14:55 Processing impacts for H3

Message originating in routine PROCESS_SOIL
Message: Entering subroutine for analyte "H3"
         On 08/24/2004 at 10:14:55.115

FN_DONE = "Soil.Done"

Message originating in routine Soil
Message: Normal Termination
         Run Completed on 08/24/2004 at 10:24:38.150
```

### **11.6.2 “soil.run” File**

A file named “soil.run” is written just as SOIL starts execution. Presence of the file signals the ESP that a SOIL run is in progress. This file does not contain any data. The file is closed and deleted when the SOIL run terminates.

### **11.6.3 “soil.done” File**

A file named “soil.done” is always written just as SOIL 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 11.8) will contain more detailed information about the type of error encountered.

### **11.6.4 Concentration Data Files**

Concentration data for different analytes are contained in separate ECDA files. SOIL reads from, and writes to, a concentration data file for every analyte being simulated. If there are 10 analytes, then 10 concentration data files will be required, along with one record map file. Sections 2.2.1 describes the format for each of these files. The ECDA data and map file names are defined in the ESD keyword file.

## **12.0 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 256 characters of information. Lines longer than 256 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 1. 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 uppercase before being stored. Quote strings are text strings that are enclosed in double quotation marks. These are stored exactly as they are read from the input file.

### **12.1 Keyword Records**

Keyword records start in column 1 with any letter from A to Z, in either upper- or lowercase. 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. The following is an example keyword record (where SAMPLEKEY starts in column 1):

```
SAMPLEKEY 2 0 500 1 100
```

The word SAMPLEKEY is the keyword. The numbers 2, 0, 500, 1, and 100 are numeric data.

### **12.2 Continuation Records**

Continuation records start with any valid separator character (except a double quotation mark). These are treated as additional data to the previous keyword record. Section 5.5 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
```

### **12.3 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. The following are some examples of comment lines:

```
$This is a comment line  
/This is a comment line  
!This is a comment line  
*This is a comment line
```

The following are some examples of in-line comments:

```
SAMPLEKEY 3 4.0 5.0 !Trailing information is ignored after the !  
SAMPLEKEY 3 4.0 5.0 $Trailing information is ignored after the $
```

### **12.4 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 ten 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 uppercase, 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 are significant. Keywords fewer than eight characters long are left justified and blank filled.

#### **12.4.1 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 quotation marks and are typically used for passing file names into a program. Only the first 200 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. The following is an example of quote string usage:

```
FILE "c:\apps\human\test.dat"
```

### **12.4.2 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 quotation mark, double quotation mark, and tab character.

Also, any character with a ASCII character storage code of less than 10 is treated as a separator character. This is used mainly to identify the ASCII tab character as a data separator. Double quotation marks are used differently than the other separators. They indicate text strings that are stored without conversion by the program. 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
```



## 13.0 References

- Baes, CF and RD Sharp. 1981. *Predicting Radionuclide Leaching from Root Zone Soil for Assessment Applications*. CONF-81606, Oak Ridge National Laboratory, Oak Ridge, Tennessee.
- 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.hanford.gov/cp/gpp/modeling/sacarchive/9-30rep.pdf>.
- 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, California.
- DOE. 1996. *Final Environmental Impact Statement of the Disposal of Decommissioned, Defueled Cruiser, Ohio class, and Los Angeles Class Naval Reactor Plants*. DOE/EIS-0259, U. S. Department of the Navy, Washington, D. C.
- DOE. 1998. *Screening Assessment and Requirements for a Comprehensive Assessment*. DOE/RL-96-16, Rev 1, Final, U.S. Department of Energy, Richland, WA. March 1998. Available on the internet at <http://sesp.pnl.gov/Reports/CRCIA/reports.htm>.
- DOE. 1999a. *Radioactive Waste Management Manual*. DOE M 435.1-1, U.S. Department of Energy, Washington, D.C. Available on the internet at <http://www.directives.doe.gov/pdfs/doe/doetext/neword/435/>
- DOE. 1999b. *Implementation Guide for Use with DOE M 435.1-1*. DOE G 435.1-1, U.S. Department of Energy, Washington, D.C.
- DOE Order 435.1. 2001. *Radioactive Waste Management*. U. S. Department of Energy Headquarters, Washington, D.C. Available on the internet at <http://www.directives.doe.gov/pdfs/doe/doetext/neword/435/>
- EPA. 1998. *Guidelines for Ecological Risk Assessment*. EPA/630/R-95/002F, Risk Assessment Forum, U.S. Environmental Protection Agency, Washington, D.C. (also published in the Federal Register 63(93):26846-26924).
- Eslinger PW, C Arimescu, DW Engel, BA Kanyid, and TB Miley. 2004a. *User Instructions for the Systems Assessment Capability, Rev. 1, Computer Codes. Volume 2: Impact Modules*. PNNL-13932-Volume 2, Pacific Northwest National Laboratory, Richland, Washington.
- Eslinger PW, DW Engel, CA Lopresti, TB Miley and WE Nichols. 2004b. *User Instructions for the Systems Assessment Capability, Rev. 1, Computer Codes. Volume 3: Utility Codes*. PNNL-13932-Volume 3, Pacific Northwest National Laboratory, Richland, Washington.

Firestone M, P Fenner-Crisp, T Barry, D Bennet, S Chang, M Callahan, A Burke, J Michaud, M Olsen, P Cirone, D Barnes, WP Wood, and SM Knott. 1997. *Guiding Principles for Monte Carlo Analysis*. EPA/630/R-97/001, Risk Assessment Forum, U.S. Environmental Protection Agency, Washington, D.C.

Iman RL, and WJ Conover. 1982. "A Distribution-Free Approach to Inducing Rank Correlations Among Input Variables." In *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.

Perkins WA and MC Richmond, 2004a. MASS2, Modular Aquatic Simulation System in Two Dimensions, Theory and Numerical Methods. PNNL-14820-1. Pacific Northwest National Laboratory, Richland, Washington.

Perkins WA and MC Richmond, 2004b. MASS2, Modular Aquatic Simulation System in Two Dimensions, User Guide and Reference. PNNL-14820-2. Pacific Northwest National Laboratory, Richland, Washington.

Ramsdell JV Jr., CA Simonen, and KW Burk. 1994. *Regional Atmospheric Transport Code for Hanford Emission Tracking (RATCHET)*. PNWD-2224 HEDR, Battelle, Pacific Northwest Laboratories, Richland, Washington.

Riley RG. 2003. *Information and Data Packet for Finalizing Release Model Capabilities and Data for SAC Rev.1 Naval Reactor Compartment Models*. January 8, 2003 (unpublished draft).

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.

Riley RG and CA Lopresti. 2004. *Release Data Package for the 2004 Composite Analysis*. PNNL-14760 Rev. 0, Pacific Northwest National Laboratory, Richland, Washington.

Stark, P.A. 1970. *Introduction to Numerical Methods*. The Macmillan Company, London.

Streng DL. 1997. "A General Algorithm for Radioactive Decay with Branching and Loss from a Medium." *Health Physics* 73(6):953-957.

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.

