SANDIA NATIONAL LABORATORIES
WASTE ISOLATION PILOT PLANT

Computational Code Execution and File Management for the 2026 Compliance Recertification Application Performance Assessment: Final State Repository (CRA-2026 FS)

REVISION 0

JENNIFER LONG (Affiliate)

Digitally signed by JENNIFER LONG (Affiliate)
Date: 2025.08.05 13:05:21 -06'00'

Authentication

Author:	Jennifer Long	Org.	08881
Technical Reviewer:	Michael Feng	Org.	08881
QA Reviewer:	Catrina Carrillo	Org.	08883
Management Reviewer:	Steve Wagner	Org.	08881

JULY 2025

WIPP:4.2.1:PA:QA-L:CRA26

Information Only

Executive Summary

The Waste Isolation Pilot Plant (WIPP) was initially certified as a repository consisting of 10 panels, which were excavated for the long-term storage and placement of transuranic waste (U.S. EPA 1998). However, due to operational constraints, the entire volume of the 10 panels is no longer available for waste disposal. In March 2024, the U.S. Department of Energy (DOE) submitted the Replacement Panels Planned Change Request (RPPCR) to the U.S. Environmental Protection Agency (EPA) for approval of waste emplacement in two new panels. These panels, referred to as replacement panels, are located to the west of the existing operations area. The RPPCR received EPA approval July 2025 (U.S. EPA 2025).

The addition of the two replacement panels is not expected to provide adequate space for the disposal of the allowable waste volume designated for WIPP under the Land Withdrawal Act (LWA); thus, the 12-panel configuration is regarded as an interim state of the repository. The DOE may pursue approval for waste disposal in seven additional panels to accommodate the full allowable volume of waste. This hypothetical 19-panel configuration is considered the anticipated final state of the repository.

The DOE demonstrates compliance with containment requirements as outlined in the Certification Criteria of Title 40 Code of Federal Regulations (CFR) Part 191.13 through performance assessment (PA). PA calculations estimate the probability and consequences of potential radionuclide releases from the repository to the accessible environment over a regulatory period of 10,000 years following facility closure. WIPP PA models are utilized to support the periodic recertification of WIPP.

The 2026 Compliance Recertification Application (CRA-2026) represents the fifth recertification application submitted to the EPA. A PA has been carried out by Sandia National Laboratories (SNL) to support the DOE's submission of the CRA-2026. The results presented in the CRA-2026 PA are compared to those from the RPPCR PA, the most recent regulatory submission.

This report documents the computational code execution and file management component for the CRA-2026 final state repository analyses (CRA26 FS).

Information Only

TABLE OF CONTENTS

EXECUTIVE SUMMARY II

1.0 INTRODUCTION 7

1.1 Key Personnel 7

2.0 WIPP PA LINUX CLUSTER 8

3.0 WIPP PA SOFTWARE CONFIGURATION MANAGEMENT AND RUN CONTROL SYSTEMS 9

3.1 Library System 10

3.2 Run Control Scripts 10

3.3 File Naming Conventions 11

3.4 WIPP PA Parameter Database 11

3.5 Access Control 12

4.0 WIPP PA CODES 12

5.0 CALCULATION FLOW 13

5.1 Inventory Decay Calculations (EPAUNI) 13

5.2 Sampling of Uncertain Parameters (LHS) 15

5.3 Salado Flow Calculations (BRAGFLO) 16

5.4 Actinide Mobilization Calculations (PANEL) 20

5.5 Salado Transport Calculations (NUTS) 22

5.6 Single-Intrusion Spallings Volume Calculations (DRSPALL) 25

5.7 Single-Intrusion Solids Volume Calculation (CUTTINGS_S) 25

5.8 Single-Intrusion Direct Brine Release Calculations (BRAGFLO-DBR) 27

5.9 Culebra Flow Calculations (MODFLOW6 and DTRKMF) 30

5.10 Culebra Transport Calculations (SECOTP2D) 33

5.11 CCDF Construction (CCDFGF) 36

6.0 REFERENCES 40

Information Only

LIST OF TABLES

Table 1 – CRA-2026 FS Run Control Team 8
Table 2 – CRA-2026 FS Analysts 8
Table 3 – WIPP PA Linux Cluster Nodes Used in CRA-2026 FS 9
Table 4 – WIPP PA Codes Used for the CRA-2026 FS PA 13
Table 5 – The EPAUNI run script files used were: 14
Table 6 – The EPAUNI input files used were: 14
Table 7 – The EPAUNI CVS repositories used were: 14
Table 8 – The EPAUNI log files used were: 14
Table 9 – The EPAUNI output files produced were: 14
Table 10 – The EPAUNI executable file used was: 15
Table 11 – The LHS run script files used were: 15
Table 12 – The LHS input file used was: 15
Table 13 – The LHS CVS repositories used were: 16
Table 14 – The LHS log files used were: 16
Table 15 – The LHS output files produced were: 16
Table 16 – The LHS executable files used were: 16
Table 17 – The BRAGFLO Scenarios 17
Table 18 – The BRAGFLO run script files used were: 17
Table 19 – The BRAGFLO input files used were: 17
Table 20 – The BRAGFLO CVS repositories used were: 18
Table 21 – The BRAGFLO log files used were: 18
Table 22 – The BRAGFLO output files produced were: 19
Table 23 – The BRAGFLO executable files used were: 19
Table 24 – The PANEL run script files used were: 20
Table 25 – The PANEL input files used were: 20
Table 26 – The PANEL CVS repositories used were: 21
Table 27 – The PANEL log files used were: 21
Table 28 – The PANEL output files produced were: 21
Table 29 – The PANEL executable files used were: 22
Table 30 – The NUTS run script files used were: 22

Information Only

Table 31 – The NUTS input files used were: 22
Table 32 – The NUTS CVS repositories used were: 23
Table 33 – The NUTS log files used were:.....23
Table 34 – The NUTS output files produced were:.....23
Table 35 – The NUTS screened-in vectors were:.....24
Table 36 – The NUTS executable files used were:.....24
Table 37 – The CUTTINGS_S run script files used were:.....25
Table 38 – The CUTTINGS_S input files used were:.....25
Table 39 – The CUTTINGS_S CVS repositories used were:.....25
Table 40 – The CUTTINGS_S log files used were:.....26
Table 41 – The CUTTINGS_S output files produced were: 26
Table 42 – The CUTTINGS_S executable files used were:.....26
Table 43 – The BRAGFLO_DBR run script files used were:.....27
Table 44 – The BRAGFLO_DBR input files used were:.....27
Table 45 – The BRAGFLO_DBR CVS repositories used were:.....28
Table 46 – The BRAGFLO_DBR log files used were:.....28
Table 47 – The BRAGFLO_DBR output files produced were: 29
Table 48 – The BRAGFLO_DBR executable files used were:.....29
Table 49– The MODFLOW6 and DTRKMF script files used were:.....30
Table 50 – The MODFLOW6 and DTRKMF input files used were:.....30
Table 51 – The MODFLOW6 and DTRKMF CVS repositories used were: 32
Table 52 – The MODFLOW6 and DTRKMF log files produced were: 32
Table 53 – The MODFLOW6 and DTRKMF output files produced were: 32
Table 54 – The MODFLOW6 and DTRKMF executable files used were:.....33
Table 55 – The SECOTP2D run script files used were:.....33
Table 56 – The SECOTP2D input files used were:.....34
Table 57 – The SECOTP2D CVS repositories used were:.....34
Table 58 – The SECOTP2D log files used were:.....35
Table 59 – The SECOTP2D output files produced were:.....35
Table 60 – The SECOTP2D executable files used were:.....36
Table 61 – The CCDFGF run script files used were:.....36

Information Only

Table 62 – The CCDFGF input files used were: 37
Table 63 – The CCDFGF CVS repositories used were: 38
Table 64 – The CCDFGF log files used were: 38
Table 65 – The CCDFGF output files produced were: 38
Table 66 – The CCDFGF executable files used were: 39

Information Only

1.0 INTRODUCTION

The Waste Isolation Pilot Plant (WIPP), located in southeastern New Mexico, has been developed by the U.S. Department of Energy (DOE) for the geologic disposal of transuranic (TRU) waste. The containment of TRU waste at WIPP is regulated by the U.S. Environmental Protection Agency (EPA) under Title 40 of the Code of Federal Regulations (CFR), Part 191.

The DOE demonstrates compliance with the containment requirements of 40 CFR Part 191.13 by means of performance assessment (PA) as described in the Certification Criteria of Title 40 CFR Part 194. Sandia National Laboratories (SNL) conducts PAs at the request of the DOE. The PA calculations estimate the probability and consequences of potential radionuclide releases from the repository to the accessible environment over a regulatory period of 10,000 years following facility closure. The PA models are continuously maintained and updated with new information, which typically leads to refinements and modifications in both the models and their parameters. Additionally, planned changes to the repository and its components necessitate updates to the WIPP PA models.

All previous WIPP certifications have been based on a repository consisting of 10 panels. In March 2024, the DOE submitted the Replacement Panels Planned Change Request (RPPCR) to seek EPA approval for waste emplacement in two new panels, referred to as replacement panels, located to the west of the existing operations area (Bollinger 2024). The RPPCR received EPA approval July 2025 (U.S. EPA 2025).

The addition of the two replacement panels is not expected to provide sufficient space for the disposal of the allowable waste volume designated for WIPP under the Land Withdrawal Act (LWA). The 12-panel configuration is thus regarded as an interim state of the repository. The DOE may seek approval for waste disposal in up to seven additional panels to accommodate the full allowable volume of waste. This 19-panel configuration is considered the anticipated final state of the repository.

The 2026 Compliance Recertification Application (CRA-2026) is the fifth recertification application for the WIPP submitted by the DOE for EPA approval. The PA executed by SNL in support of the CRA-2026 PA is described in the Analysis Plan 214 (AP-214) (Hansen et al. 2025). This report documents the computational code execution and file management component for the CRA-2026 final state repository analyses (CRA26 FS).

1.1 Key Personnel

The Run Control Coordinator modified and maintained the scripts used to run the WIPP PA codes, created and maintained the Concurrent Versions System (CVS) repository used to archive calculation results, and performe the calcualtions. The run control team members are listed in Table 1.

Information Only

Table 1 – CRA-2026 FS Run Control Team

Function	Personnel
CRA-2026 PA Task Lead	Seth King
Run Control Coordinator, Software Configuration Management (SCM) Coordinator, and PA Parameter Database (PAPDB) Administrator	Jennifer Long

The WIPP PA analysts were responsible for preparing input for the various WIPP PA codes and performing output data analysis and interpretation of the calculation results. The PA analysts for the CRA-2026 are shown in Table 2.

Table 2 – CRA-2026 FS Analysts

Major WIPP PA Codes	Analyst
EPAUNI	Sungtae Kim
LHS	Seth King
BRAGFLO	Brad Day
NUTS	Paul Dockerty
PANEL	Paul Dockerty
CUTTINGS_S	Sarah Brunell
BRAGFLO_DBR	Todd Zeitler
CCDFGF	Seth King
STEPWISE	Seth King

2.0 WIPP PA LINUX CLUSTER

The CRA-2026 FS was performed using the WIPP PA Linux cluster. The WIPP PA Linux cluster consists of one head node (FWM) that distributes jobs to 24 other nodes. The node name, hardware description, Central Processing Unit (CPU) information, operating system and number of CPU's for each node are provided in Table 3.

Information Only

Table 3 – WIPP PA Linux Cluster Nodes Used in CRA-2026 FS

Node	Hardware Type	CPU	Operating System	# CPU'S
FWM	Dell PowerEdge R640	Intel(R) Xeon(R) Gold 6132 CPU @ 2.60GHz	CentOS 7	56
CB001	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB002	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB003	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB004	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB005	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB006	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB007	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB008	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB009	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB010	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB011	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB012	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB013	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB014	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB015	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB016	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB017	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB018	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB019	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB020	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB021	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB022	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB023	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80
CB024	Dell PowerEdge C6420	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz	CentOS 7	80

3.0 WIPP PA SOFTWARE CONFIGURATION MANAGEMENT AND RUN CONTROL SYSTEMS

The computer simulations that form the core of the CRA-2026 FS are made fully traceable and reproducible through three key elements: 1) An archive or library system (CVS) for controlling, tracking changes, and monitoring user access for source codes, executables, simulation input and output files; 2) a scripting tool that interacts with the library or archive to fetch input files, execute codes, and store output files; and 3) an access control capability to allow only approved individuals to run official calculations and have write-access to areas where officical inputs and/or results are stored. The following sections briefly describe how these elements are implemented on the WIPP PA computing cluser. Additional information is available in Kirchner (2012).

Most calculations performed on the WIPP PA Linux Cluster also take advantage of the WIPP PA Parameter Database (PAPDB) to control the use of key modeling parameters. The PAPDB is discussed in Section 3.4.

Information Only

3.1 Library System

The general system used on UNIX platforms to implement, control, test, distribute, and ensure overall software quality assurance compliance for WIPP PA is referred to as Software Configuration Management under Unix (SCMU).

The CVS is the library system used for WIPP PA. CVS is a software package that identifies, controls, and tracks original files and their subsequent revisions. CVS is integral to the testing and validation program used by WIPP PA.

CVS contains relevant files as defined by SNL (Long 2024). These may include:

- source codes
- executable files
- associated input files
- make files and build scripts
- output files
- input and output files for test problems (those provided by the code sponsor and any created by the tester)
- test logs
- reports of test results
- and reference/user documentation

The SCMU provides utilities and tools to access and update controlled software such that various versions of files can be recovered and modifications or enhancements applied to the controlled software can be tracked. Once a code is placed in a CVS repository, the sponsor can check out the code for modification. An authorized user (typically the "run master") can then update (check in) the code when the modifications are completed. All CVS file creations and replacements are tracked and retained, providing an automatic, accurate history of the revisions and allowing any prior version of a file to be retrieved.

3.2 Run Control Scripts

The execution of WIPP PA codes on the Linux Cluster for regulatory calculations is orchestrated using a collection of scripts. The run control scripts export all input files and executables from access-controlled CVS libraries and parameters from the PAPDB. The scripts are modified and maintained by the Run Control Coordinator. The run control scripts also store all important output files to access-controlled CVS libraries. The run control scripts read from input files that specify:

- Analysis identification
- Analysis directory
- Replicate, scenario, intrusion times, intrusion locations and vector information, as appropriate
- Names and locations of all codes (executables) used in the run
- Code input files and their storage location
- Code output files and their post-run destination
- Log file name and post-run disposition

Information Only

3.3 File Naming Conventions

The file names contain as much meaningful information as possible. This was accomplished using a naming convention that provided the following information:

- The code associated with the file, if applicable
- The calculation type (e.g. BRAGFLO)
- An identifier indicating the file is part of the CRA-2026 calculations (CRA26_FS)
- The replicate number, if applicable
- The scenario number, if applicable
- The vector number, if applicable
- The time intrusion value, if applicable
- The intrusion location (upper, lower, or middle) used, if applicable
- The mining type (full or partial) represented, if applicable
- Other miscellaneous code specific file information
- The file format or file type (input text, binary, .cdb, debug, etc.,)

Underscores (_) are used to separate the distinct elements of identification embedded in a file name. The first item in the file name is typically used to designate which code is reading or creating the file. In some cases, the second item specifies the code prefix that a generic code is being run to support. The next item is the calculation identification which is sometimes followed by the other code specific items used to give meaning to file names.

A file with the name gm_bf_CRA26_FS.inp can be decoded as follows:

gm The file is used by the GENMESH code.

bf This file relates to the code BRAGFLO (bf) run stream.

CRA26_FS This is the calculation identification.

inp This is a code input file.

Many files also include replicate, scenario, and vector references as follows:

r1 The replicate number associated with this calculation is “1”.

s3 The scenario number associated with this calculation is “3”.

v007 The vector number associated with this calculation is “007”.

3.4 WIPP PA Parameter Database

The PAPDB contains data values, associated models, source information, usage, and additional information documenting parameter information. In addition to parameter management, the PAPDB allows certain WIPP PA codes (PRELHS and MATSET) to retrieve parameter data for use in the PA. PRELHS is used to retrieve parameter data for subjectively uncertain (sampled) parameters (e.g., range, mean, distribution, etc.). MATSET is used mainly to retrieve values for constant parameters; however, it also retrieves the median values for parameters modeled with

Information Only

uncertainty distributions. See the PAPDB Design Document (Long 2019) for more information. PAPDB Version 3.00 was used for the CRA-2026.

Codes that access the PAPDB require logicals for the database name, analysis name, computational code name, computational code version, and analysis revision be set. These items are defined in the run control script. The production PAPDB, "ParamDB", was used in the CRA-2026.

It should be noted that the WIPP PA codes only retrieve parameter values from the PAPDB and have no capability to modify the parameter values in the database. Changes to the PAPDB are only made by the PAPDB Administrator in accordance with Nuclear Waste Management Procedure NP 9-2: Parameters (Long 2025).

3.5 Access Control

Access to the various components under SCMU control will be provided to each member of the team. All users will be given "read" level access to the SCMU CVS repositories. Check-in of files will be restricted to authorized users only, primarily those assigned to be the SCM Coordinator and SCMU Librarians ("Run Master"). On the WIPP PA Linux cluster, a special "run_mast" account has been set up to perform official calculations. Only the Run Control Coordinator can access this account. All official calculations are run within an access-controlled working directory specific to the code being run. Only the "run_mast" can write to this working directory. In addition, only the "run_mast" account can write to the official libraries.

4.0 WIPP PA CODES

The major WIPP PA codes used for the CRA-2026 FS analysis are listed in Table 4. These codes will be executed on the WIPP PA Linux Cluster, the components of which are listed in Table 3. These codes have been qualified under Nuclear Waste Management Procedure NP 19-1: Software Requirements (Long 2024).

Information Only

Table 4 – WIPP PA Codes Used for the CRA-2026 FS PA

Code	Version	Executable[1]	Build Date
ALGEBRACDB	2.37	algebracdb	8/17/20
BRAGFLO	8.00	bragflo	1/14/25
CCDFGF	8.02	ccdfgf	9/16/24
CCDFVECTORSTATS	1.02	ccdfvectorstats	8/19/20
CUTTINGS_S	6.04	cuttings_s	11/13/20
DTRKMF	2.00	dtrkmf	9/25/24
EPAUNI	1.21	epauni	2/9/24
GENMESH	6.11	genmesh	8/20/20
GROPECDB	2.14	gropecdb	6/16/20
ICSET	2.24	icset	8/20/20
LHS	3.00	lhs	11/1/21
MATSET	9.25	matset	9/8/20
MODFLOW6	6.2.2	modflow6	5/16/22
NUTS	2.08	nuts	11/16/21
PANEL	5.03	panel	4/24/24
POSTBRAG	4.03	postbrag	9/8/20
POSTLHS	4.12	postlhs	9/17/20
POSTSECOTP2D	1.06	postsecotp2d	8/17/20
PREBRAG	9.01	prebrag	8/20/20
PRECCDFGF	3.01	preccdfgf	7/28/22
PRELHS	2.46	prelhs	8/29/22
PRESECOTP2D	1.25	presecotp2d	1/25/23
RELATE	1.46	relate	8/20/20
SCREEN_NUTS	1.03	screen_nuts	8/19/20
SECOTP2D	1.44	secotp2d	8/20/20
STEPWISE	2.23	stepwise	10/28/20
SUMMARIZE	3.03	summarize	8/17/20

[1] Executables are located in /data/cvs/CVSLIB/WIPP_CODES/PA_CODES/*CODE*/Build/Linux

5.0 CALCULATION FLOW

5.1 Inventory Decay Calculations (EPAUNI)

The EPAUNI code calculates the decay of the radionuclide components in each inventory waste stream over the 10,000-year regulatory period (for use in calculating direct solids releases). These

Information Only

calculations are deterministic, so multiple replicates and vectors of uncertain parameters are not used. Calculations are performed for both contact-handled (CH) and remote-handled (RH) waste.

Table 5 – The EPAUNI run script files used were:

File	Repository	Comment
RunControl/EPAUNI.py	$REP1/CRA26_FS/EPAUNI	Python run control script
RunControl/EPAUNIlib.py	$REP1/CRA26_FS/EPAUNI	Python run control script class modules
RunControl/RunPA.py	$REP1/CRA26_FS/EPAUNI	Main control script

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 6 – The EPAUNI input files used were:

File	Repository	Comment
Input/epu_CRA26_FS_ch.inp	$REP1/CRA26_FS/EPAUNI	Input file
Input/epu_CRA26_FS_ch_misc.inp	$REP1/CRA26_FS/EPAUNI	Input file
Input/epu_CRA26_FS_rh.inp	$REP1/CRA26_FS/EPAUNI	Input file
Input/epu_CRA26_FS_rh_misc.inp	$REP1/CRA26_FS/EPAUNI	Input file

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 7 – The EPAUNI CVS repositories used were:

CVS Repositories
$REP1/CRA26_FS/EPAUNI
$CODE/EPAUNI

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

Table 8 – The EPAUNI log files used were:

File	Repository	Comment
RunControl/EPAUNI.log	$REP1/CRA26_FS/EPAUNI	Log file
RunControl/EPAUNI.rtf	$REP1/CRA26_FS/EPAUNI	Formatted log file (Word file)

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 9 – The EPAUNI output files produced were:

File	Repository	Comment
Output/epu_CRA26_FS_ch.dat	$REP1/CRA26_FS/EPAUNI	Results file
Output/epu_CRA26_FS_ch.dia	$REP1/CRA26_FS/EPAUNI	Diagnostic file (units)

Information Only

File	Repository	Comment
Output/epu_CRA26_FS_ch.out	$REP1/CRA26_FS/EPAUNI	Results file (times)
Output/epu_CRA26_FS_ch.out2	$REP1/CRA26_FS/EPAUNI	Results file (format 2)
Output/epu_CRA26_FS_ch_activity.dia	$REP1/CRA26_FS/EPAUNI	Diagnostic file (activity)
Output/epu_CRA26_FS_rh.dat	$REP1/CRA26_FS/EPAUNI	Results file
Output/epu_CRA26_FS_rh.dia	$REP1/CRA26_FS/EPAUNI	Diagnostic file (units)
Output/epu_CRA26_FS_rh.out	$REP1/CRA26_FS/EPAUNI	Results file (times)
Output/epu_CRA26_FS_rh.out2	$REP1/CRA26_FS/EPAUNI	Results file (format 2)
Output/epu_CRA26_FS_rh_activity.dia	$REP1/CRA26_FS/EPAUNI	Diagnostic file (activity)

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 10 – The EPAUNI executable file used was:

File	Repository	Comment
Build/Linux/epauni (Ver:1.21)	$CODE/EPAUNI	Computes decay of radionuclide components in inventory

Where:
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

5.2 Sampling of Uncertain Parameters (LHS)

Sampling of the uncertain parameters used by the various process model codes is performed with the PRELHS and LHS codes. PRELHS reads information about the ranges and distributions of the uncertain parameters from the PAPDB and formats this information for LHS. The LHS code implements the sampling algorithms. LHS is executed once per replicate.

Table 11 – The LHS run script files used were:

File	Repository	Comment
RunControl/LHS.py	$REP1/CRA26_FS/LHS	Python run control script
RunControl/LHSlib.py	$REP1/CRA26_FS/LHS	Python run control script class modules
RunControl/RunPA.py	$REP1/CRA26_FS/LHS	Main control script

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 12 – The LHS input file used was:

File	Repository	Comment
Input/lhs1_CRA26_FS_r*i*_con.inp	$REP1/CRA26_FS/PRELHS	Input file
Input/lhs2_CRA26_FS_rep1.txt	$REP1/CRA26_FS/LHS	Input file
Input/lhs2_CRA26_FS_rep2.txt	$REP1/CRA26_FS/LHS	Input file
Input/lhs2_CRA26_FS_rep3.txt	$REP1/CRA26_FS/LHS	Input file

Where:

Information Only

i is 1-3
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 13 – The LHS CVS repositories used were:

CVS Repositories
$REP1/CRA26_FS/LHS
$REP1/CRA26_FS/PRELHS
$CODE/LHS
$CODE/PRELHS

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

Table 14 – The LHS log files used were:

File	Repository	Comment
RunControl/LHS.log	$REP1/CRA26_FS/LHS	Log file
RunControl/LHS.rtf	$REP1/CRA26_FS/LHS	Formatted log file (Word file)

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 15 – The LHS output files produced were:

File	Repository	Comment
Output/lhs1_CRA26_FS_r*i*_con.dbg	$REP1/CRA26_FS/PRELHS	Debug file
Output/lhs1_CRA26_FS_r*i*_con.trn	$REP1/CRA26_FS/PRELHS	Transfer file
Output/lhs2_CRA26_FS_r*i*_con.dbg	$REP1/CRA26_FS/LHS	Debug file
Output/lhs2_CRA26_FS_r*i*_con.trn	$REP1/CRA26_FS/LHS	Transfer file

Where:
i is 1-3
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 16 – The LHS executable files used were:

File	Repository	Comment
Build/Linux/lhs (Ver:3.00)	$CODE/LHS	Code to sample uncertain parameters
Build/Linux/prelhs (Ver:2.46)	$CODE /PRELHS	Pre-processes data for lhs

Where:
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

5.3 Salado Flow Calculations (BRAGFLO)

Brine and gas flow in and around the repository and in overlying formations is calculated using the BRAGFLO suite of codes (PREBRAG, BRAGFLO, and POSTBRAG) in conjunction with

Information Only

several utility codes. The entire set of calculations is performed for three replicates. Each replicate includes six scenarios (S1-S6) designed to cover a range of drilling intrusion types and times. For each replicate/scenario combination, calculations are performed for 100 vectors of uncertain model input parameters.

The statistical code, STEPWISE, evaluates variable importance by developing regression models between the observed response and input variables using either a forward (which is a special case of stepwise regression), backward, or stepwise regression procedure on the raw or ranked data.

Table 17 – The BRAGFLO Scenarios

BRAGFLO Scenario	Description[1,2]
S1-BF	Undisturbed
S2-BF	E1 intrusion at 350 years
S3-BF	E1 intrusion at 1000 years
S4-BF	E2 intrusion at 350 years
S5-BF	E2 intrusion at 1000 years
S6-BF	E2 intrusion at 1000 years, E1 intrusion at 2000 years

[1] E1 intrusion penetrates the repository and intersects a brine pocket in the underlying Castile Formation.

[2] E2 intrusion penetrates the repository but does not encounter a Castile brine pocket.

Table 18 – The BRAGFLO run script files used were:

File	Repository	Comment
RunControl/BRAGFLO.py	$REP1/CRA26_FS/BRAGFLO	Python run control script
Source/preSTEPWISE.py	$CODE/PRESTEPWISE	Python run control script
RunControl/BRAGFLOlib.py	$REP1/CRA26_FS/BRAGFLO	Python run control script class modules
RunControl/RunPA.py	$REP1/CRA26_FS/BRAGFLO	Main control script

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

Table 19 – The BRAGFLO input files used were:

File	Repository	Comment
Input/alg1_bf_CRA26_FS.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/alg2_bf_CRA26_FS.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/bf1_CRA26_FS_s*n*.inp	$REP1/CRA26_FS/PREBRAG	Input file
Input/bf1_CRA26_FS_s*n*_mod1.inp	$REP1/CRA26_FS/PREBRAG	Input file
Input/bf1_CRA26_FS_s*n*_mod2.inp	$REP1/CRA26_FS/PREBRAG	Input file
Input/bf2_CRA26_FS_closure.dat	$REP1/CRA26_FS/BRAGFLO	Input file
Input/gm_bf_CRA26_FS.inp	$REP1/CRA26_FS/GENMESH	Input file
Input/ic_bf_CRA26_FS.inp	$REP1/CRA26_FS/ICSET	Input file

Information Only

File	Repository	Comment
Input/ms_bf_CRA26_FS.inp	$REP1/CRA26_FS/MATSET	Input file
Input/pstp_bf_CRA26_FS.inp	$REP1/CRA26_FS/STEPWISE	Input file
Input/pstp_CRA26_FS_names.txt	$REP1/CRA26_FS/STEPWISE	Input file
Input/stp_bf_CRA26_FS.tmp	$REP1/CRA26_FS/STEPWISE	Input file
Input/sum_bf_CRA26_FS.inp	$REP1/CRA26_FS/SUMMARIZE	Input file

Where:
n is 1-6
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 20 – The BRAGFLO CVS repositories used were:

CVS Repositories
$REP1/CRA26_FS/ALGEBRACDB
$REP1/CRA26_FS/BRAGFLO
$REP1/CRA26_FS/GENMESH
$REP1/CRA26_FS/ICSET
$REP1/CRA26_FS/MATSET
$REP1/CRA26_FS/PREBRAG
$REP1/CRA26_FS/PRESTEPWISE
$REP1/CRA26_FS/STEPWISE
$REP1/CRA26_FS/SUMMARIZE
$CODE/ALGEBRACDB
$CODE/BRAGFLO
$CODE/GENMESH
$CODE/ICSET
$CODE/MATSET
$CODE/POSTBRAG
$CODE/POSTLHS
$CODE/PREBRAG
$CODE/PRESTEPWISE
$CODE/STEPWISE
$CODE/SUMMARIZE

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

Table 21 – The BRAGFLO log files used were:

File	Repository	Comment
RunControl/BRAGFLO.log	$REP1/CRA26_FS/BRAGFLO	Log file
RunControl/BRAGFLO.rtf	$REP1/CRA26_FS/BRAGFLO	Formatted log file (Word file)

Where:

Information Only

$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 22 – The BRAGFLO output files produced were:

File	Repository	Comment
Output/bf2_CRA26_FS_r*i*_s*n*_v*vvv*.inp	$REP1/CRA26_FS/PREBRAG	Input file
Output/bf2_CRA26_FS_r*i*_s*n*_v*vvv*.log	$REP1/CRA26_FS/BRAGFLO	Log file
Output/bf2_CRA26_FS_r*i*_s*n*_v*vvv*.sum	$REP1/CRA26_FS/BRAGFLO	Summary file
Output/pstp_bf_CRA26_FS_r*i*_s*n*_ddata.txt	$REP1/CRA26_FS/PRESTEPWISE	Stepwise variable file
Output/pstp_bf_CRA26_FS_r*i*_s*n*_idata.txt	$REP1/CRA26_FS/PRESTEPWISE	Stepwise variable file
Output/stp_bf_CRA26_FS.inp	$REP1/CRA26_FS/PRESTEPWISE	STEPWISE user input
Output/stp_bf_CRA26_FS_r*i*_s*n*.txt	$REP1/CRA26_FS/STEPWISE	Corr file
Output/sum_bf_CRA26_FS_r*i*_s*n*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file

Where:
i is 1-3
n is 1-6
vvv is 001-100
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 23 – The BRAGFLO executable files used were:

File	Repository	Comment
Build/Linux/algebracdb (Ver:2.37)	$CODE/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Linux/bragflo (Ver:8.00)	$CODE/BRAGFLO	Computes brine and gas flow in the repository
Build/Linux/genmesh (Ver:6.11)	$CODE/GENMESH	Generates the CAMDAT computational grid
Build/Linux/icset (Ver:2.24)	$CODE/ICSET	Assigns initial conditions to the CAMDAT grid elements
Build/Linux/matset (Ver:9.25)	$CODE/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Linux/postbrag (Ver:4.03)	$CODE/POSTBRAG	Post-processes data for bragflo
Build/Linux/postlhs (Ver:4.12)	$CODE/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Linux/prebrag (Ver:9.01)	$CODE/PREBRAG	Pre-processes data for bragflo
Build/Linux/stepwise (Ver:2.23)	$CODE/STEPWISE	Statistical program for stepwise regression
Build/Linux/summarize (Ver:3.03)	$CODE/SUMMARIZE	Writes tables of data from many CAMDAT files

Where:
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

Information Only

5.4 Actinide Mobilization Calculations (PANEL)

Actinide mobilization in WIPP brines as dissolved species and sorbed to colloids is calculated with the PANEL code, run in "concentration" mode (referred to here as PANEL_CON). This information is needed to compute the amount of radionuclides available for transport away from the repository (in direct brine releases to the surface, or in brines migrating to the accessible environment via subsurface pathways). Three replicates of the mobilization calculations are performed for CRA26 FS.

Table 24 – The PANEL run script files used were:

File	Repository	Comment
RunControl/PANEL.py	$REP1/CRA26_FS/PANEL	Python run control script
RunControl/PANELlib.py	$REP1/CRA26_FS/PANEL	Python run control script class modules
RunControl/RunPA.py	$REP1/CRA26_FS/PANEL	Main control script

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 25 – The PANEL input files used were:

File	Repository	Comment
Input/alg1_panel_CRA26_FS.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Output/alg2_bf_CRA26_FS_*ri*_s*n*_v*vvv*.cdb	$REP1/CRA26_FS/ALGEBRACDB	CDB transfer file
Input/alg2_panel_CRA26_FS_b1.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/alg2_panel_CRA26_FS_b2.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/alg2_panel_CRA26_FS_b3.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/alg2_panel_CRA26_FS_b4.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/alg2_panel_CRA26_FS_b5.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/alg3_panel_CRA26_FS_b1.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/alg3_panel_CRA26_FS_b2.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/alg3_panel_CRA26_FS_b3.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/alg3_panel_CRA26_FS_b4.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/alg3_panel_CRA26_FS_b5.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/gm_panel_CRA26_FS.inp	$REP1/CRA26_FS/GENMESH	Input file
Input/ms_panel_CRA26_FS.inp	$REP1/CRA26_FS/MATSET	Input file
Input/sum_panel_con.inp	$REP1/CRA26_FS/SUMMARIZE	Input file
Input/sum_panel_int.inp	$REP1/CRA26_FS/SUMMARIZE	Input file
Input/sum_panel_st.inp	$REP1/CRA26_FS/SUMMARIZE	Input file

Where:
i is 1-3
n is 1-6
vvv is 001-100
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Information Only

Table 26 – The PANEL CVS repositories used were:

CVS Repositories
$REP1/CRA26_FS/ALGEBRACDB
$REP1/CRA26_FS/GENMESH
$REP1/CRA26_FS/MATSET
$REP1/CRA26_FS/PANEL
$REP1/CRA26_FS/SUMMARIZE
$CODE/ALGEBRACDB
$CODE/GENMESH
$CODE/MATSET
$CODE/PANEL
$CODE/POSTLHS
$CODE/SUMMARIZE

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

Table 27 – The PANEL log files used were:

File	Repository	Comment
RunControl/PANEL.log	$REP1/CRA26_FS/PANEL	Log file
RunControl/PANEL.rtf	$REP1/CRA26_FS/PANEL	Formatted log file (Word file)

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 28 – The PANEL output files produced were:

File	Repository	Comment
Output/sum_panel_con_CRA26_FS_b1_r*i*_s*n*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_panel_con_CRA26_FS_b2_r*i*_s*n*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_panel_con_CRA26_FS_b3_r*i*_s*n*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_panel_con_CRA26_FS_b4_r*i*_s*n*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_panel_con_CRA26_FS_b5_r*i*_s*n*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_panel_int_CRA26_FS_r*i*_s*o*_t*tttt*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_panel_st_CRA26_FS_r*i*_s*n*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file

Where:
i is 1-3
n is 1-2
o is 6
ttttt is 00100, 00350, 01000, 02000, 04000, 06000, 09000
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Information Only

Table 29 – The PANEL executable files used were:

File	Repository	Comment
Build/Linux/algebracdb (Ver:2.37)	$CODE/ ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Linux/genmesh (Ver:6.11)	$CODE/GENMESH	Generates the CAMDAT computational grid
Build/Linux/matset (Ver:9.25)	$CODE/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Linux/panel (Ver:5.03)	$CODE/PANEL	Computes release concentrations of nuclides from repository
Build/Linux/postlhs (Ver:4.12)	$CODE/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Linux/summarize (Ver:3.03)	$CODE/ SUMMARIZE	Writes tables of data from CAMDAT files

Where:
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

5.5 Salado Transport Calculations (NUTS)

Radionuclide transport in the Salado for single intrusion conditions is calculated using the NUTS code. Three replicate calculations are performed for CRA26 FS.

Table 30 – The NUTS run script files used were:

File	Repository	Comment
RunControl/NUTS.py	$REP1/CRA26_FS/NUTS	Python run control script
RunControl/NUTSlib.py	$REP1/CRA26_FS/NUTS	Python run control script class modules
RunControl/RunPA.py	$REP1/CRA26_FS/NUTS	Main control script

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 31 – The NUTS input files used were:

File	Repository	Comment
Input/alg_nut_iso_CRA26_FS.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/alg_nut_scn_CRA26_FS.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Output/bf2_CRA26_FS_*ri*_*sn*_*vvvv*.inp	$REP1/CRA26_FS/PREBRAG	Input file
Output/bf3_CRA26_FS_*ri*_*sn*_*vvvv*.cdb	$REP1/CRA26_FS/BRAGFLO	CDB transfer file
Input/ms_nut_CRA26_FS.inp	$REP1/CRA26_FS/MATSET	Input file
Input/nut_int_CRA26_FS_*so*_*ttttt*.inp	$REP1/CRA26_FS/NUTS	Input file
Input/nut_iso_CRA26_FS_*sn*.inp	$REP1/CRA26_FS/NUTS	Input file
Input/nut_scn_CRA26_FS_*sn*.inp	$REP1/CRA26_FS/NUTS	Input file
Output/panel_con_CRA26_FS_b1_*ri*_*sn*_*vvvv*.cdb	$REP1/CRA26_FS/PANEL	CDB transfer file

Where:

Information Only

i is 1-3
n is 1-5
o is 2-5
ttttt is 0100 for S2, S4
03000, 05000, 07000, 09000 for S3, S5
vvv is 001-100
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 32 – The NUTS CVS repositories used were:

Repository
$CODE/ALGEBRACDB
$CODE/MATSET
$CODE/NUTS
$CODE/SCREEN_NUTS
$CODE/SUMMARIZE
$REP1/CRA26_FS/ALGEBRACDB
$REP1/CRA26_FS/BRAGFLO
$REP1/CRA26_FS/MATSET
$REP1/CRA26_FS/NUTS
$REP1/CRA26_FS/PANEL
$REP1/CRA26_FS/PREBRAG
$REP1/CRA26_FS/SCREEN_NUTS
$REP1/CRA26_FS/SUMMARIZE

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

Table 33 – The NUTS log files used were:

File	Repository	Comment
RunControl/NUTS.log	$REP1/CRA26_FS/NUTS	Log file
RunControl/NUTS.rtf	$REP1/CRA26_FS/NUTS	Formatted log file (Word file)

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 34 – The NUTS output files produced were:

File	Repository	Comment
Output/screen_nut_scn_CRA26_FS_r*i*_EDIT.inp	$REP1/CRA26_FS/SCREEN_NUTS	Input file
Output/screen_nut_scn_CRA26_FS_r*i*_s*n*.out	$REP1/CRA26_FS/SCREEN_NUTS	Output file
Output/sum_nut_CRA26_FS_r*i*_s*n*_t*ttttt*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_nut_scn_CRA26_FS_r*i*_s*n*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file

Information Only

Where:
i is 1-3
n is 1-5
ttttt is 0100 for S1
00100, 00350 for S2,S4
01000, 03000, 05000, 07000, 09000 for S3, S5
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 35 – The NUTS screened-in vectors were:

Replicate	Scenario	Vectors
1	1	1,2,6,7,8,9,13,14,22,23,24,28,30,34,36,41,50,52,55,59,61,62,66,70,80,83
1	2	1,2,6,7,9,13,14,22,23,24,28,30,34,36,41,52,55,59,61,62,66,70,80,83
1	3	1,7,9,13,14,22,23,30,36,41,55,59,61,62,66,70,80,83
1	4	8,50
1	5	50
2	1	4,16,21,24,25,27,29,33,34,35,36,37,38,40,41,63,68,80,84,87,92,98
2	2	4,16,21,24,25,27,29,33,34,35,36,37,38,40,41,63,68,80,84,87,92,98
2	3	4,21,24,25,29,33,34,36,37,38,68,80,98
2	4	63
2	5	63
3	1	10,18,26,32,37,39,42,44,45,47,58,64,65,66,71,74,75,78,79,86,91,93,96,97
3	2	10,26,32,37,39,42,44,45,47,64,65,66,71,74,75,78,79,86,91,93,96,97
3	3	10,18,26,32,37,39,42,45,58,64,65,66,71,74,75,86,91,93,97
3	4	10
3	5	10

Table 36 – The NUTS executable files used were:

File	Repository	Comment
Build/Linux/algebracdb (Ver:2.37)	$CODE/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Linux/matset (Ver:9.25)	$CODE/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Linux/nuts (Ver:2.08)	$CODE/NUTS	Nuclide Transport system model
Build/Linux/screen_nuts (Ver:1.03)	$CODE/SCREEN_NUTS	Executable file
Build/Linux/summarize (Ver:3.03)	$CODE/SUMMARIZE	Writes tables of data from many CAMDAT files

Where:
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

Information Only

5.6 Single-Intrusion Spallings Volume Calculations (DRSPALL)

DRSPALL Version 1.22 was not rerun for CRA26_FS. Instead, the DRSPALL Version 1.22 output results from a previous run (Kirchner et al. 2015) were used as input to the CUTTINGS_S code in the CRA26_FS calculations.

5.7 Single-Intrusion Solids Volume Calculation (CUTTINGS_S)

The total volume of radionuclide-contaminated solids that may reach the surface during a drilling intrusion event is calculated by the CUTTINGS_S code.

Table 37 – The CUTTINGS_S run script files used were:

File	Repository	Comment
RunControl/CUTTINGS_S.py	$REP1/CRA26_FS/CUTTINGS_S	Python run control script
RunControl/CUTTINGS_Slib.py	$REP1/CRA26_FS/CUTTINGS_S	Python run control script class modules
RunControl/RunPA.py	$REP1/CRA26_FS/CUTTINGS_S	Main control script

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 38 – The CUTTINGS_S input files used were:

File	Repository	Comment
Output/bf3_CRA26_FS_r*i*_s*n*_v*vvv*.cdb	$REP1/CRA26_FS/BRAGFLO	CDB transfer file
Input/cusp_CRA26_FS.inp	$REP1/CRA26_FS/CUTTINGS_S	Input file
Input/gm_cusp_CRA26_FS.inp	$REP1/CRA26_FS/GENMESH	Input file
Input/ms_cusp_CRA26_FS.inp	$REP1/CRA26_FS/MATSET	Input file
Output/mspall_drs_PABC09_r*i*.out	$REP1/PABC09/DRSPALL	Input file

Where:
i is 1-3
n is 1-5
vvv is 001-100
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 39 – The CUTTINGS_S CVS repositories used were:

Repository
$CODE/CUTTINGS_S
$CODE/GENMESH
$CODE/MATSET
$CODE/POSTLHS
$REP1/CRA26_FS/BRAGFLO
$REP1/CRA26_FS/CUTTINGS_S

Information Only

Repository
$REP1/CRA26_FS/GENMESH
$REP1/CRA26_FS/MATSET
$REP1/CRA26_FS/POSTLHS
$REP1/PABC09/DRSPALL

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

Table 40 – The CUTTINGS_S log files used were:

File	Repository	Comment
RunControl/CUTTINGS_S.log	$REP1/CRA26_FS/CUTTINGS_S	Log file
RunControl/CUTTINGS_S.rtf	$REP1/CRA26_FS/CUTTINGS_S	Formatted log file (Word file)

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 41 – The CUTTINGS_S output files produced were:

File	Repository	Comment
Output/cusp_CRA26_FS_master_r*i*.inp	$REP1/CRA26_FS/CUTTINGS_S	Input file
Output/cusp_CRA26_FS_r*i*.tbl	$REP1/CRA26_FS/CUTTINGS_S	Table file

Where:
i is 1-3
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 42 – The CUTTINGS_S executable files used were:

File	Repository	Comment
Build/Linux/cuttings_s (Ver:6.04)	$CODE/CUTTINGS_S	Computes cuttings/spall generated by drilling
Build/Linux/genmesh (Ver:6.11)	$CODE/GENMESH	Generates the CAMDAT computational grid
Build/Linux/matset (Ver:9.25)	$CODE/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Linux/postlhs (Ver:4.12)	$CODE/POSTLHS	Assigns sampled parameters to the grid blocks and elements

Where:
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

Information Only

5.8 Single-Intrusion Direct Brine Release Calculations (BRAGFLO-DBR)

Single-intrusion direct brine release volumes are calculated using the BRAGFLO suite of codes (PREBRAG, BRAGFLO, POSTBRAG).

Table 43 – The BRAGFLO_DBR run script files used were:

File	Repository	Comment
RunControl/BRAGFLO_DBR.py	$REP1/CRA26_FS/BRAGFLO_DBR	Python run control script
RunControl/BRAGFLO_DBRlib.py	$REP1/CRA26_FS/BRAGFLO_DBR	Python run control script modules
RunControl/RunPA.py	$REP1/CRA26_FS/BRAGFLO_DBR	Main control script

Where:

$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 44 – The BRAGFLO_DBR input files used were:

File	Repository	Comment
Input/alg1_dbr_CRA26_FS.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/alg2_dbr_CRA26_FS_*so*.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/alg3_dbr_CRA26_FS_L.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/alg3_dbr_CRA26_FS_M.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/alg3_dbr_CRA26_FS_O.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/alg3_dbr_CRA26_FS_U.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/bf1_dbr_CRA26_FS_L.inp	$REP1/CRA26_FS/PREBRAG	Input file
Input/bf1_dbr_CRA26_FS_M.inp	$REP1/CRA26_FS/PREBRAG	Input file
Input/bf1_dbr_CRA26_FS_O.inp	$REP1/CRA26_FS/PREBRAG	Input file
Input/bf1_dbr_CRA26_FS_s*n*_100_L.inp	$REP1/CRA26_FS/PREBRAG	Input file
Input/bf1_dbr_CRA26_FS_s*n*_100_M.inp	$REP1/CRA26_FS/PREBRAG	Input file
Input/bf1_dbr_CRA26_FS_s*n*_100_O.inp	$REP1/CRA26_FS/PREBRAG	Input file
Input/bf1_dbr_CRA26_FS_s*n*_100_U.inp	$REP1/CRA26_FS/PREBRAG	Input file
Input/bf1_dbr_CRA26_FS_U.inp	$REP1/CRA26_FS/PREBRAG	Input file
Output/bf3_CRA26_FS_r*i*_s*o*_v*vvv*.cdb	$REP1/CRA26_FS/BRAGFLO	CDB transfer file
Output/cusp_CRA26_FS_r*i*_s*o*_t*tttt*_L_v*vvv*.cdb	$REP1/CRA26_FS/CUTTINGS_S	CDB transfer file
Output/cusp_CRA26_FS_r*i*_s*o*_t*tttt*_M_v*vvv*.cdb	$REP1/CRA26_FS/CUTTINGS_S	CDB transfer file
Output/cusp_CRA26_FS_r*i*_s*o*_t*tttt*_O_v*vvv*.cdb	$REP1/CRA26_FS/CUTTINGS_S	CDB transfer file
Output/cusp_CRA26_FS_r*i*_s*o*_t*tttt*_U_v*vvv*.cdb	$REP1/CRA26_FS/CUTTINGS_S	CDB transfer file
Input/gm_dbr_CRA26_FS.inp	$REP1/CRA26_FS/GENMESH	Input file
Input/ic_dbr_CRA26_FS_s*o*.inp	$REP1/CRA26_FS/ICSET	Input file
Input/ms_dbr_CRA26_FS.inp	$REP1/CRA26_FS/MATSET	Input file
Input/rel1_dbr_CRA26_FS.inp	$REP1/CRA26_FS/RELATE	Input file
Input/rel2_dbr_CRA26_FS_s*o*.inp	$REP1/CRA26_FS/RELATE	Input file
Input/sum_dbr.inp	$REP1/CRA26_FS/SUMMARIZE	Input file

Information Only

Where:
i is 1-3
n is 1
o is 1-5
ttttt is 00100, 00350, 01000, 03000, 05000, 10000 for S1
00550, 00750, 02000, 04000, 10000 for S2, S4
01200, 01400, 03000, 05000, 10000 for S3, S5
vvv is 001-100
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 45 – The BRAGFLO_DBR CVS repositories used were:

Repository
$CODE/ALGEBRACDB
$CODE/BRAGFLO
$CODE/GENMESH
$CODE/ICSET
$CODE/MATSET
$CODE/POSTBRAG
$CODE/POSTLHS
$CODE/PREBRAG
$CODE/RELATE
$CODE/SUMMARIZE
$REP1/CRA26_FS/ALGEBRACDB
$REP1/CRA26_FS/BRAGFLO
$REP1/CRA26_FS/BRAGFLO_DBR
$REP1/CRA26_FS/CUTTINGS_S
$REP1/CRA26_FS/GENMESH
$REP1/CRA26_FS/ICSET
$REP1/CRA26_FS/MATSET
$REP1/CRA26_FS/PREBRAG
$REP1/CRA26_FS/RELATE
$REP1/CRA26_FS/SUMMARIZE

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

Table 46 – The BRAGFLO_DBR log files used were:

File	Repository	Comment
RunControl/BRAGFLO_DBR.log	$REP1/CRA26_FS/BRAGFLO_DBR	Log file
RunControl/BRAGFLO_DBR.rtf	$REP1/CRA26_FS/BRAGFLO_DBR	Formatted log file (Word file)

Information Only

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 47 – The BRAGFLO_DBR output files produced were:

File	Repository	Comment
Output/bf2_dbr_CRA26_FS_r*i*_s*n*_t*tttt*_L_v*vvv*.inp	$REP1/CRA26_FS/BRAGFLO_DBR	Input file
Output/bf2_dbr_CRA26_FS_r*i*_s*n*_t*tttt*_M_v*vvv*.inp	$REP1/CRA26_FS/BRAGFLO_DBR	Input file
Output/bf2_dbr_CRA26_FS_r*i*_s*n*_t*tttt*_O_v*vvv*.inp	$REP1/CRA26_FS/BRAGFLO_DBR	Input file
Output/bf2_dbr_CRA26_FS_r*i*_s*n*_t*tttt*_U_v*vvv*.inp	$REP1/CRA26_FS/BRAGFLO_DBR	Input file
Output/sum_dbr_CRA26_FS_r*i*_s*n*_t*tttt*_L.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_dbr_CRA26_FS_r*i*_s*n*_t*tttt*_M.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_dbr_CRA26_FS_r*i*_s*n*_t*tttt*_O.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_dbr_CRA26_FS_r*i*_s*n*_t*tttt*_U.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file

Where:
i is 1-3
n is 1-5
ttttt is 00100, 00350, 01000, 03000, 05000, 10000 for S1
00550, 00750, 02000, 04000, 10000 for S2, S4
01200, 01400, 03000, 05000, 10000 for S3, S5
vvv is 001-100
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 48 – The BRAGFLO_DBR executable files used were:

File	Repository	Comment
Build/Linux/algebracdb (Ver:2.37)	$CODE/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Linux/bragflo (Ver:8.00)	$CODE/BRAGFLO	Computes brine and gas flow in the repository
Build/Linux/genmesh (Ver:6.11)	$CODE/GENMESH	Generates the CAMDAT computational grid
Build/Linux/icset (Ver:2.24)	$CODE/ICSET	Assigns initial conditions to the CAMDAT grid elements
Build/Linux/matset (Ver:9.25)	$CODE/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Linux/postbrag (Ver:4.03)	$CODE/POSTBRAG	Post-processes data for bragflo
Build/Linux/postlhs (Ver:4.12)	$CODE/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Linux/prebrag (Ver:9.01)	$CODE/PREBRAG	Pre-processes data for bragflo
Build/Linux/relate (Ver:1.46)	$CODE/RELATE	Transfers CAMDAT data to another CAMDAT file
Build/Linux/summarize (Ver:3.03)	$CODE/SUMMARIZE	Writes tables of data from many CAMDAT files

Information Only

Where:
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

5.9 Culebra Flow Calculations (MODFLOW6 and DTRKMF)

The mining-modified Culebra flow fields were calculated using MODFLOW6 (MF6) and DTRKMF.

Table 49– The MODFLOW6 and DTRKMF script files used were:

File	Repository	Comment
RunControl/MODFLOW6.py	$REP1/CRA26_FS/MODFLOW6	Python run control script
RunControl/MODFLOW6lib.py	$REP1/CRA26_FS/MODFLOW6	Python run control script class modules
RunControl/RunPA.py	$REP1/CRA26_FS/MODFLOW6	Main control script
Source/100x100_to_50x50.py	$CODE/MININGMOD	Python Code
Source/enlarge_mining_areas.py	$CODE/MININGMOD	Python Code
Source/mining_mod.py	$CODE/MININGMOD	Python Code
Source/summarize_dtrkmf.py	$CODE/SUMMARIZE_DTRKMF	Python Code

Where:

$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

Table 50 – The MODFLOW6 and DTRKMF input files used were:

File	Repository	Comment
Input/MiningMod/100x100_to_50x50.in	$REP1/CRA26_FS/MODFLOW6	ancillary script
Input/dtrkmf_CRA26_FS_crp1.inp	$REP1/CRA26_FS/DTRKMF	Input file
Input/dtrkmf_CRA26_FS_crp2.inp	$REP1/CRA26_FS/DTRKMF	Input file
Input/dtrkmf_CRA26_FS_crp3.inp	$REP1/CRA26_FS/DTRKMF	Input file
Input/dtrkmf_CRA26_FS_crp4.inp	$REP1/CRA26_FS/DTRKMF	Input file
Inputs/enlarge_mining_areas.in	$REP2/MiningMod/	Input file
Input/gm_mf6_CRA26_FS.inp	$REP1/CRA26_FS/GENMESH	Input file
Input/keepers	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_files/mf6_head.chd	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_50/mf6_head.chd.50	$REP1/CRA26_FS/MODFLOW6	Input file
wipp_tfields_2021/Outputs/result_2023.02.21.itr5/r005/mf6_head.chd_stress_period_data_1.txt	$REP2/WIPP_EXTERNAL/ap203	mf input file
Input/mf6_files/mf6_head.dis	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_50/mf6_head.dis.50	$REP1/CRA26_FS/MODFLOW6	Input file
wipp_tfields_2021/Outputs/result_2023.02.21.itr5/r005/mf6_head.dis_botm.txt	$REP2/WIPP_EXTERNAL/ap203	mf input file

Information Only

File	Repository	Comment
wipp_tfields_2021/Outputs/result_2023.02.21.itr5/r005/mf6_head.dis_idomain.txt	$REP2/WIPP_EXTERNAL/ap203	mf input file
wipp_tfields_2021/Outputs/result_2023.02.21.itr5/r005/mf6_head.dis_top.txt	$REP2/WIPP_EXTERNAL/ap203	mf input file
Input/mf6_files/mf6_head.ic	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_50/mf6_head.ic.50	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_files/mf6_head.ims	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_50/mf6_head.ims.50	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_files/mf6_head.init_head.mod	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_files/mf6_head.nam	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_50/mf6_head.nam.50	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_mining/mf6_head.nam.mining	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_files/mf6_head.npf	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_50/mf6_head.npf.50	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_mining/mf6_head.npf.mining	$REP1/CRA26_FS/MODFLOW6	Input file
wipp_tfields_2021/Outputs/result_2023.02.21.itr5/r005/mf6_head.npf_icelltype.txt	$REP2/WIPP_EXTERNAL/ap203	mf input file
wipp_tfields_2021/Outputs/result_2023.02.21.itr5/r005/mf6_head.npf_k.txt	$REP2/WIPP_EXTERNAL/ap203	tfield mf input file
wipp_tfields_2021/Outputs/result_2023.02.21.itr5/r005/mf6_head.npf_k22.txt	$REP2/WIPP_EXTERNAL/ap203	tfield mf input file
Input/mf6_files/mf6_head.oc	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_50/mf6_head.oc.50	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_files/mf6_head.rcha	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_50/mf6_head.rcha.50	$REP1/CRA26_FS/MODFLOW6	Input file
wipp_tfields_2021/Outputs/result_2023.02.21.itr5/r005/mf6_head.rcha_recharge_1.txt	$REP2/WIPP_EXTERNAL/ap203	tfield mf input file
Input/mf6_files/mf6_head.sto	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_50/mf6_head.sto.50	$REP1/CRA26_FS/MODFLOW6	Input file
wipp_tfields_2021/Outputs/result_2023.02.21.itr5/r005/mf6_head.sto_iconvert.txt	$REP2/WIPP_EXTERNAL/ap203	mf input file
wipp_tfields_2021/Outputs/result_2023.02.21.itr5/r005/mf6_head.sto_ss.txt	$REP2/WIPP_EXTERNAL/ap203	tfield mf input file
wipp_tfields_2021/Outputs/result_2023.02.21.itr5/r005/mf6_head.sto_sy.txt	$REP2/WIPP_EXTERNAL/ap203	mf input file
Input/mf6_files/mf6_head.tdis	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_50/mf6_head.tdis.50	$REP1/CRA26_FS/MODFLOW6	Input file
Input/mf6_50/mfsim.nam	$REP1/CRA26_FS/MODFLOW6	Input file
Inputs/mining_areas_xyz.dat	$REP2/MiningMod/	Input file
Input/MiningMod/mining_mod.in	$REP1/CRA26_FS/MODFLOW6	ancillary script
Input/ms_mf6_CRA26_FS.inp	$REP1/CRA26_FS/MATSET	Input file
Input/sum_mf6_CRA26_FS.inp	$REP1/CRA26_FS/SUMMARIZE	Input file

Information Only

Where:

$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES
$REP2 = /data/cvs/CVSLIB

Table 51 – The MODFLOW6 and DTRKMF CVS repositories used were:

CVS Repositories
$REP1/CRA26_FS/DTRKMF
$REP1/CRA26_FS/GENMESH
$REP1/CRA26_FS/MATSET
$REP1/CRA26_FS/MODFLOW6
$REP1/CRA26_FS/SUMMARIZE
$REP2/MiningMod/
$REP2/WIPP_EXTERNAL/ap203
$CODE/DTRKMF
$CODE/GENMESH
$CODE/MATSET
$CODE/MININGMOD
$CODE/MODFLOW6
$CODE/POSTLHS
$CODE/SUMMARIZE
$CODE/SUMMARIZE_DTRKMF

Where:

$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES
$REP2 = /data/cvs/CVSLIB
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

Table 52 – The MODFLOW6 and DTRKMF log files produced were:

File	Repository	Comment
RunControl/MODFLOW6.log	$REP1/CRA26_FS/MODFLOW6	Log file
RunControl/MODFLOW6.rtf	$REP1/CRA26_FS/MODFLOW6	Formatted log file (Word file)

Where:

$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 53 – The MODFLOW6 and DTRKMF output files produced were:

File	Repository	Comment
Output/dtrk_mf6_CRA26_FS_r*i*_m*m*_c*c*_v*vvv*.out	$REP1/CRA26_FS/DTRKMF	particle track file
Output/sum_mf6_CRA26_FS_r*i*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_mf6_CRA26_FS_r*i*_m*m*_c*c*.tbl	$REP1/CRA26_FS/DTRKMF	particle summary

Where:

i is 1-3

Information Only

m is (f)ull or (p)art
c is 1-4
vvv is 001-100

$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 54 – The MODFLOW6 and DTRKMF executable files used were:

File	Repository	Comment
Build/Linux/dtrkmf (Ver:2.00)	$CODE/DTRKMF	Executable file
Build/Linux/genmesh (Ver:6.11)	$CODE/GENMESH	Generates the CAMDAT computational grid
Build/Linux/matset (Ver:9.25)	$CODE/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Linux/modflow6 (Ver:6.2.2)	$CODE/MODFLOW6	Executable file
Build/Linux/postlhs (Ver:4.12)	$CODE/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Linux/summarize (Ver:3.03)	$CODE/SUMMARIZE	Writes tables of data from many CAMDAT files

Where:
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

5.10 Culebra Transport Calculations (SECOTP2D)

Culebra Transport calculations are calculated using the SECOTP2D suite of codes (PRESECOTP2D, SECOTP2D, POSTSECOTP2D).

The statistical code, STEPWISE, evaluates variable importance by developing regression models between the observed response and input variables using either a forward (which is a special case of stepwise regression), backward, or stepwise regression procedure on the raw or ranked data.

Table 55 – The SECOTP2D run script files used were:

File	Repository	Comment
RunControl/RunPA.py	$REP1/CRA26_FS/SECOTP2D	Main control script
Source/preSTEPWISE.py	$CODE/PRESETPWISE	Python run control script
RunControl/SECOTP2D.py	$REP1/CRA26_FS/SECOTP2D	Python run control script
RunControl/SECOTP2Dlib.py	$REP1/CRA26_FS/SECOTP2D	Python run control script class modules

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

Information Only

Table 56 – The SECOTP2D input files used were:

File	Repository	Comment
Input/alg_st2d_CRA26_FS.inp	$REP1/CRA26_FS/ALGEBRACDB	Input file
Input/gm_st2d_CRA26_FS.inp	$REP1/CRA26_FS/GENMESH	Input file
Input/grope_st2d_CRA26_FS.inp	$REP1/CRA26_FS/GROPECDB	Input file
Output/R1/full/v001/mf6_head.cbc.50	$REP1/CRA26_FS/MODFLOW50m	MODFLOW6 file
Output/R1/full/v001/mf6_head.dis.50.grb	$REP1/CRA26_FS/MODFLOW50m	MODFLOW6 file
Input/ms_st2d_CRA26_FS.inp	$REP1/CRA26_FS/MATSET	Input file
Input/pstp_CRA26_FS_names.txt	$REP1/CRA26_FS/STEPWISE	Input file
Input/pstp_st2d_CRA26_FS.inp	$REP1/CRA26_FS/STEPWISE	Input file
Input/rel_st2d_CRA26_FS.inp	$REP1/CRA26_FS/RELATE	Input file
Input/st2d1_CRA26_FS_*cc*.inp	$REP1/CRA26_FS/PRESECOTP2D	Input file
Input/st2d1_CRA26_FS_*cc*_mod.inp	$REP1/CRA26_FS/PRESECOTP2D	Input file
Input/stp_st2d_CRA26_FS.tmp	$REP1/CRA26_FS/STEPWISE	Input file
Input/sum1_st2d_CRA26_FS.inp	$REP1/CRA26_FS/SUMMARIZE	Input file
Output/sum_mf6_CRA26_FS_r*i*_mf_c*c*.tbl	$REP1/CRA26_FS/DTRKMF	Particle track
Output/sum_mf6_CRA26_FS_r*i*_mp_c*c*.tbl	$REP1/CRA26_FS/DTRKMF	Particle track
Input/sum_st2d_CRA26_FS.inp	$REP1/CRA26_FS/SUMMARIZE	Input file

Where:

i is 1-3
c is 1-4
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 57 – The SECOTP2D CVS repositories used were:

CVS Repositories
$REP1/CRA26_FS/ALGEBRACDB
$REP1/CRA26_FS/DTRKMF
$REP1/CRA26_FS/GENMESH
$REP1/CRA26_FS/GROPECDB
$REP1/CRA26_FS/MATSET
$REP1/CRA26_FS/MODFLOW50m
$REP1/CRA26_FS/PRESECOTP2D
$REP1/CRA26_FS/PRESTEPWISE
$REP1/CRA26_FS/RELATE
$REP1/CRA26_FS/SECOTP2D
$REP1/CRA26_FS/STEPWISE
$REP1/CRA26_FS/SUMMARIZE
$CODE/ALGEBRACDB
$CODE/GENMESH
$CODE/GROPECDB
$CODE/MATSET

Information Only

CVS Repositories
$CODE/POSTLHS
$CODE/POSTSECOTP2D
$CODE/PRESECOTP2D
$CODE/PRESTEPWISE
$CODE/RELATE
$CODE/SECOTP2D
$CODE/STEPWISE
$CODE/SUMMARIZE

Where:

$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

Table 58 – The SECOTP2D log files used were:

File	Repository	Comment
RunControl/SECOTP2D.log	$REP1/CRA26_FS/SECOTP2D	Log file
RunControl/SECOTP2D.rtf	$REP1/CRA26_FS/SECOTP2D	Formatted log file (Word file)

Where:

$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 59 – The SECOTP2D output files produced were:

File	Repository	Comment
Output/pstp_st2d_CRA26_FS_r*i*_mf_c*c*_ddata.txt	$REP1/CRA26_FS/PRESTEPWISE	Stepwise variable file
Output/pstp_st2d_CRA26_FS_r*i*_mf_c*c*_idata.txt	$REP1/CRA26_FS/PRESTEPWISE	Stepwise variable file
Output/pstp_st2d_CRA26_FS_r*i*_mp_c*c*_ddata.txt	$REP1/CRA26_FS/PRESTEPWISE	Stepwise variable file
Output/pstp_st2d_CRA26_FS_r*i*_mp_c*c*_idata.txt	$REP1/CRA26_FS/PRESTEPWISE	Stepwise variable file
Output/stp_st2d_CRA26_FS.inp	$REP1/CRA26_FS/PRESTEPWISE	STEPWISE user input
Output/stp_st2d_CRA26_FS_r*i*_mf_c*c*.txt	$REP1/CRA26_FS/STEPWISE	Corr file
Output/stp_st2d_CRA26_FS_r*i*_mp_c*c*.txt	$REP1/CRA26_FS/STEPWISE	Corr file
Output/sum1_st2d_CRA26_FS_r*i*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_st2d_CRA26_FS_r*i*_mf_c*c*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_st2d_CRA26_FS_r*i*_mp_c*c*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file

Where:

i is 1-3
c is 1-4
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Information Only

Table 60 – The SECOTP2D executable files used were:

File	Repository	Comment
Build/Linux/algebracdb (Ver:2.37)	$CODE/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Linux/genmesh (Ver:6.11)	$CODE/GENMESH	Generates the CAMDAT computational grid
Build/Linux/gropecdb (Ver:2.14)	$CODE/GROPECDB	Displays CAMDAT data in text form
Build/Linux/matset (Ver:9.25)	$CODE/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Linux/postlhs (Ver:4.12)	$CODE/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Linux/postsecotp2d (Ver:1.06)	$CODE/POSTSECOTP2D	Post-processes data for secotp2d
Build/Linux/presecotp2d (Ver:1.25)	$CODE/PRESECOTP2D	Pre-processes data for secotp2d
Build/Linux/relate (Ver:1.46)	$CODE/RELATE	Transfers CAMDAT data to another CAMDAT file
Build/Linux/secotp2d (Ver:1.44)	$CODE/SECOTP2D	SECO 2D transport code
Build/Linux/stepwise (Ver:2.23)	$CODE/STEPWISE	Statistical program for stepwise regression
Build/Linux/summarize (Ver:3.03)	$CODE/SUMMARIZE	Writes tables of data from many CAMDAT files

Where:

$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

5.11 CCDF Construction (CCDFGF)

The complimentary cumulative distribution functions (CCDFs) for radionuclide releases to the accessible environment are constructed using the PRECCDFGF/CCDFGF code suite.

The statistical code, STEPWISE, evaluates variable importance by developing regression models between the observed response and input variables using either a forward (which is a special case of stepwise regression), backward, or stepwise regression procedure on the raw or ranked data.

Table 61 – The CCDFGF run script files used were:

File	Repository	Comment
Source/preSTEPWISE.py	$CODE/PRESTEPWISE	Python Code
Source/pstp_merge_sum_dtrkmf.py	$CODE/merge_DTRKMF	Python Code
RunControl/CCDFGF.py	$REP1/CRA26_FS/CCDFGF	Python run control script
RunControl/CCDFGFlib.py	$REP1/CRA26_FS/CCDFGF	Python run control script class modules
RunControl/RunPA.py	$REP1/CRA26_FS/CCDFGF	Main control script

Where:

$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

Information Only

Table 62 – The CCDFGF input files used were:

File	Repository	Comment
Input/ccgf_CRA26_FS_control_r*i*.inp	$REP1/CRA26_FS/CCDFGF	Input file
Output/cusp_CRA26_FS_r*i*.tbl	$REP1/CRA26_FS/CUTTINGS_S	Table file
Output/epu_CRA26_FS_ch.dat	$REP1/CRA26_FS/EPAUNI	Table file
Output/epu_CRA26_FS_rh.dat	$REP1/CRA26_FS/EPAUNI	Table file
Input/gm_ccgf_CRA26_FS.inp	$REP1/CRA26_FS/GENMESH	Input file
Input/intrusiontimes.in	$REP1/CRA26_FS/PRECCDFGF	Input file
Input/ms_ccgf_CRA26_FS.inp	$REP1/CRA26_FS/MATSET	Input file
Input/pstp_ccgf_CRA26_FS.inp	$REP1/CRA26_FS/STEPWISE	Input file
Input/pstp_CRA26_FS_names.txt	$REP1/CRA26_FS/STEPWISE	Input file
Input/stp_ccgf_CRA26_FS.tmp	$REP1/CRA26_FS/STEPWISE	Input file
Output/sum_dbr_CRA26_FS_r*i*_so_t*vvvvv*_L.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_dbr_CRA26_FS_r*i*_so_t*vvvvv*_M.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_dbr_CRA26_FS_r*i*_so_t*vvvvv*_O.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_dbr_CRA26_FS_r*i*_so_t*vvvvv*_U.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_mf6_CRA26_FS_r*i*_mf_c*c*.tbl	$REP1/CRA26_FS/DTRKMF	Particle track
Output/sum_mf6_CRA26_FS_r*i*_mp_c*c*.tbl	$REP1/CRA26_FS/DTRKMF	Particle track
Output/sum_nut_CRA26_FS_r*i*_so_t*uuuuu*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_panel_con_CRA26_FS_b*b*_r*i*_s*n*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_panel_int_CRA26_FS_r*i*_s6_t*ttttt*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_panel_st_CRA26_FS_r*i*_s*n*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_st2d_CRA26_FS_r*i*_mf_c*c*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file
Output/sum_st2d_CRA26_FS_r*i*_mp_c*c*.tbl	$REP1/CRA26_FS/SUMMARIZE	Table file

Where:

i is 1-3
n is 1-2
o is 1-5
b is 1-5
c is 1-4
ttttt is 00100, 00350, 01000, 02000, 04000, 06000, 09000
uuuuu is 0100 for S1
00100, 00350 for S2,S4
01000, 03000, 05000, 07000, 09000 for S3, S5
vvvvv is 00100, 00350, 01000, 03000, 05000, 10000 for S1
00550, 00750, 02000, 04000, 10000 for S2, S4
01200, 01400, 03000, 05000, 10000 for S3, S5

$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Information Only

Table 63 – The CCDFGF CVS repositories used were:

CVS Repositories
$REP1/CRA26_FS/CCDFGF
$REP1/CRA26_FS/CUTTINGS_S
$REP1/CRA26_FS/DTRKMF
$REP1/CRA26_FS/EPAUNI
$REP1/CRA26_FS/GENMESH
$REP1/CRA26_FS/MATSET
$REP1/CRA26_FS/PRECCDFGF
$REP1/CRA26_FS/PRESTEPWISE
$REP1/CRA26_FS/STEPWISE
$REP1/CRA26_FS/SUMMARIZE
$CODE/merge_DTRKMF
$CODE/PRESTEPWISE
$CODE/CCDFGF
$CODE/CCDFVECTORSTATS
$CODE/GENMESH
$CODE/MATSET
$CODE/POSTLHS
$CODE/PRECCDFGF
$CODE/STEPWISE

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

Table 64 – The CCDFGF log files used were:

File	Repository	Comment
RunControl/CCDFGF.log	$REP1/CRA26_FS/CCDFGF	Log file
RunControl/CCDFGF.rtf	$REP1/CRA26_FS/CCDFGF	Formatted log file (Word file)

Where:
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 65 – The CCDFGF output files produced were:

File	Repository	Comment
Output/ccgf_CRA26_FS_reltab_r*i*.dat	$REP1/CRA26_FS/PRECCDFGF	Results file
Output/ccgf_CRA26_FS_r*i*.out	$REP1/CRA26_FS/CCDFGF	Results file
Output/pstp_ccgf_CRA26_FS_r*i*_ddata.txt	$REP1/CRA26_FS/PRESTEPWISE	Stepwise variable file
Output/pstp_ccgf_CRA26_FS_r*i*_idata.txt	$REP1/CRA26_FS/PRESTEPWISE	Stepwise variable file
Output/stp_ccgf_CRA26_FS_r*i*.inp	$REP1/CRA26_FS/PRESTEPWISE	STEPWISE user input
Output/stp_ccgf_CRA26_FS_r*i*.txt	$REP1/CRA26_FS/STEPWISE	Corr file

Information Only

File	Repository	Comment
Output/sum_dtrk_CRA26_FS_r*i*_mf.tbl	$REP1/CRA26_FS/PRESTEPWISE	Merged DTRKMF file
Output/sum_dtrk_CRA26_FS_r*i*_mp.tbl	$REP1/CRA26_FS/PRESTEPWISE	Merged DTRKMF file

Where:
i is 1-3
$REP1 = /data/cvs/CVSLIB/WIPP_ANALYSES

Table 66 – The CCDFGF executable files used were:

File	Repository	Comment
Build/Linux/ccdfgf (Ver:8.02)	$CODE/CCDFGF	Constructs complimentary cumulative distribution functions for radionuclide releases
Build/Linux/ccdfvectorstats (Ver:1.02)	$CODE/CCDFVECTORSTATS	Executable file
Build/Linux/genmesh (Ver:6.11)	$CODE/GENMESH	Generates the CAMDAT computational grid
Build/Linux/matset (Ver:9.25)	$CODE/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Linux/postlhs (Ver:4.12)	$CODE/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Linux/preccdfgf (Ver:3.01)	$CODE/PRECCDFGF	Pre-processes data for ccdfgf
Build/Linux/stepwise (Ver:2.23)	$CODE/STEPWISE	Statistical program for stepwise regression

Where:
$CODE = /data/cvs/CVSLIB/WIPP_CODES/PA_CODES

Information Only

6.0 REFERENCES

Bollinger, M. 2024. Letter to Ms. L. Veal. Subject: Planned Change Request for the use of Replacement Panels 11 and 12. March 12, 2024. CBFO:ERCD:MG:SV:24-0168. Carlsbad, NM: U.S. DOE, Carlsbad Field Office.

Hansen, C., S. Brunell, B. Day, S. King, and J. Wilgus. 2025. Analysis Plan for the 2026 WIPP Compliance Recertification Application Performance Assessment. AP-214. EIMS-0004440569. Carlsbad, NM: Sandia National Laboratories.

Kirchner, T. 2012. WIPP Performance Assessment Software Configuration Management Under UNIX (SCMU) Plan. ERMS 558444. Carlsbad, N.M:Sandia National Laboratories.

Kirchner, T., A. Gilkey, and J. Long. 2015. Addendum to the Summary Report on the Migration of the WIPP PA Codes. ERMS 564675. Carlsbad, NM:Sandia National Laboratories.

Long, J.J. 2025. Nuclear Waste Management Procedure NP 9-2: Parameters, Revision 4. EIMS-0004430022. Carlsbad, NM:Sandia National Laboratories.

Long, J.J. 2024. Nuclear Waste Management Procedure NP 19-1: Software Requirements, Revision 20. ERMS 581776. Carlsbad, NM:Sandia National Laboratories.

Long, J.J. 2019. Design Document/User's Manual/Implementation Document for PAPDB Version 3.00, Document Version 3.00. ERMS 570734. Carlsbad, NM:Sandia National Laboratories.

U.S. Department of Energy (DOE) 1996. Title 40 CFR Part 191 Compliance Certification Application for the Waste Isolation Pilot Plant. U.S. Department of Energy Waste Isolation Pilot Plant, Carlsbad Area Office, Carlsbad, NM. DOE/CAO-1996-2184.

U.S. Department of Energy (DOE) 2004. Title 40 CFR Part 191 Compliance Recertification Application for the Waste Isolation Pilot Plant, 10 vols., U.S. Department of Energy Waste Isolation Pilot Plant, Carlsbad Area Office, Carlsbad, NM. DOE/WIPP 2004-3231.

U.S. Environmental Protection Agency (U.S. EPA). 1998. 40 CFR Part 194: Criteria for the Certification and Recertification of the Waste Isolation Pilot Plant's Compliance with the Disposal Regulations: Certification Decision. Federal Register, vol. 63 (May 18, 1998): 27354 – 27406.

U.S. Environmental Protection Agency (U.S. EPA). 2025. Letter from Abigale Tardif to Mark Bollinger, Subject: EPA's Review and Approval of DOE's RPPCR for Waste Panels 11 and 12 at WIPP. July 31, 2025. Washington, DC: U.S. EPA.

Information Only