SANDIA NATIONAL LABORATORIES

WASTE ISOLATION PILOT PLANT

Computational Code Execution and File Management for the 2019 Compliance Recertification Application Performance Assessment (CRA-2019 PA)

REVISION 0

	Print	Signature	Date
Author:	Jennifer Long	Jennifer Long	5/22/19
Technical Reviewer:	Sarah Brunell	Sarah Brunell	5/29/2019
QA Reviewer:	Shelly Nielsen	Shelly R. Nielsen	6-5-19
Management Reviewer:	Chris Camphouse	[signature]	5/29/2019

ERMS #571375

MAY 2019

WIPP:4.2.1:PA:QA-L:571155

This page intentionally left blank.

TABLE OF CONTENTS

LIST OF TABLES

This page intentionally left blank.

Executive Summary

The Land Withdrawal Act requires that the U.S. Department of Energy (DOE) apply for recertification of the Waste Isolation Pilot Plant (WIPP) every five years following the initial 1999 waste shipment. The 2019 Compliance Recertification Application (CRA-2019) is the fourth WIPP recertification application submitted for approval by the U.S. Environmental Protection Agency. A performance assessment (PA) has been executed by Sandia National Laboratories in support of the DOE submittal of the CRA-2019. Results found in the CRA-2019 PA are compared to those obtained in the 2014 Compliance Recertification Application (CRA-2014) in order to assess repository performance in terms of the current regulatory baseline. This report documents the computational code execution and file management component for the CRA-2019 PA. Changes incorporated into the CRA-2019 PA include repository planned changes, parameter updates, and refinements to PA implementation.

This page intentionally left blank.

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 (deep underground) disposal of transuranic (TRU) waste. Containment of TRU waste at the WIPP is regulated by the U.S. Environmental Protection Agency (EPA) according to the regulations set forth in Title 40 of the Code of Federal Regulations (CFR), Part 191. The DOE demonstrates compliance with the containment requirements according to the Certification Criteria in Title 40 CFR Part 194 by means of performance assessment (PA) calculations performed by Sandia National Laboratories (SNL). WIPP PA calculations estimate the probability and consequence of potential radionuclide releases from the repository to the accessible environment for a regulatory period of 10,000 years after facility closure. The models used in PA are maintained and updated with new information as part of an ongoing process. Improved information regarding important WIPP features, events, and processes typically results in refinements and modifications to PA models and the parameters used in them. Planned changes to the repository and/or the components therein also result in updates to WIPP PA models. WIPP PA models are used to support the repository recertification process that occurs at five-year intervals following the receipt of the first waste shipment at the site in 1999.

PA calculations were included in the 1996 Compliance Certification Application (CCA) (U.S. DOE 1996), and in a subsequent Performance Assessment Verification Test (PAVT) (MacKinnon and Freeze 1997a, 1997b and 1997c). Based in part on the CCA and PAVT PA calculations, the EPA certified that the WIPP met the regulatory containment criteria. The facility was approved for disposal of transuranic waste in May 1998 (U.S. EPA 1998). PA calculations were an integral part of the 2004 Compliance Recertification Application (CRA-2004) (U.S. DOE 2004). During their review of the CRA-2004, the EPA requested an additional PA calculation, referred to as the CRA-2004 Performance Assessment Baseline Calculation (PABC) (Leigh et al. 2005), be conducted with modified assumptions and parameter values (Cotsworth 2005). Following review of the CRA-2004 and the CRA-2004 PABC, the EPA recertified the WIPP in March 2006 (U.S. EPA 2006).

PA calculations were completed for the second WIPP recertification and documented in the 2009 Compliance Recertification Application (CRA-2009). The CRA-2009 PA resulted from continued review of the CRA-2004 PABC, including a number of technical changes and corrections, as well as updates to parameters and improvements to the PA computer codes (Clayton et al. 2008). To incorporate additional information which was received after the CRA-2009 PA was completed, but before the submittal of the CRA-2009, the EPA requested an additional PA calculation, referred to as the 2009 Compliance Recertification Application Performance Assessment Baseline Calculation (PABC-2009) (Clayton et al. 2010), be undertaken which included updated information (Cotsworth 2009). Following the completion and submission of the PABC-2009, the WIPP was recertified in 2010 (U.S. EPA 2010).

PA calculations were completed for the third WIPP recertification and documented in the 2014 Compliance Recertification Application (CRA-2014). Following the completion and submission of the CRA-2014, the WIPP was recertified in 2017 (U.S. EPA 2017).

The Land Withdrawal Act (U.S. Congress 1992) requires that the DOE apply for WIPP recertification every five years following the initial 1999 waste shipment. The 2019 Compliance Recertification Application (CRA-2019) is the fourth WIPP recertification application submitted by the DOE for EPA approval. The PA executed by SNL in support of the CRA-2019 is detailed in AP-181 (Zeitler 2019). The CRA-2019 PA includes repository planned changes, parameter updates, and refinements to PA implementation. Results found in the CRA-2019 PA are compared to those obtained in the CRA-2014 in order to assess repository performance in terms of the current regulatory baseline.

Additionally, a separate calculation ("CRA19_CL") was performed as a sensitivity study that investigates the impact of the assumption of a more rapid closure of open areas in the repository on calculationed releases. This sensitivity study is detailed in AP-181 (Zeitler 2019). The only difference between the analyses were the input files used. For simplicity in run control execution, no output from the CRA-2019 analysis were used in the CRA19_CL analysis. Identical code versions were executed for each analysis and identical parameters were pulled from the parameter database. The STEPWISE code was not executed for the CRA19_CL analysis.

Code execution, file locations, and file names for the CRA19_CL are identical to the CRA19 except for the _CL that differentiates the two.

This report documents the computational code execution and file management component for the CRA-2019 and CRA-2019_CL PA analyses.

1.1 Key Personnel

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

Table 1 – CRA-2019 Run Control Team

Function	Personnel
CRA-2019 PA Task Lead	Todd Zeitler
Run Control Coordinator and PA Parameter Database 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-2019 are shown in Table 2.

Table 2 – CRA-2019 Analysts

Major WIPP PA Codes	Analyst
EPAUNI	Dwayne Kicker
LHS	Todd Zeitler
BRAGFLO	Brad Day
NUTS	Ramesh Sarathi
PANEL	Ramesh Sarathi
CUTTINGS_S	Dwayne Kicker
BRAGFLO_DBR	James Bethune
CCDFGF	Sarah Brunell
STEPWISE	Todd Zeitler

2.0 WIPP PA SOLARIS CLUSTER

The CRA-2019 was performed using the WIPP PA Solaris cluster. The WIPP PA Solaris cluster consists of one head node (SAN) that distributes jobs to 20 other nodes. The node name, hardware description, CPU information, operating system and number of CPU's for each node are provided in Table 3.

Table 3 – WIPP PA Solaris Cluster Nodes Used in CRA-2019

Node	Hardware Type	CPU	Operating System	# CPU'S
GD	Oracle/SUN X6270 m2	x86 (GenuineIntel 206C2 family 6 model 44 step 2 clock 3458 MHz)	Oracle Solaris 11	24
IRON	Oracle/SUN X6270 m2	x86 (GenuineIntel 206C2 family 6 model 44 step 2 clock 3458 MHz)	Oracle Solaris 11	24
VH	Oracle/SUN X6270 m2	x86 (GenuineIntel 206C2 family 6 model 44 step 2 clock 3458 MHz)	Oracle Solaris 11	24
PF	Oracle/SUN X6270 m2	x86 (GenuineIntel 206C2 family 6 model 44 step 2 clock 3458 MHz)	Oracle Solaris 11	24
BLS	Oracle/SUN X6270 m2	x86 (GenuineIntel 206C2 family 6 model 44 step 2 clock 3458 MHz)	Oracle Solaris 11	24
BEP	Oracle/SUN X6270 m2	x86 (GenuineIntel 206C2 family 6 model 44 step 2 clock 3458 MHz)	Oracle Solaris 11	24

GFD	Oracle/SUN X6270 m2	x86 (GenuineIntel 206C2 family 6 model 44 step 2 clock 3458 MHz)	Oracle Solaris 11	24
DC5	Oracle/SUN X6270 m2	x86 (GenuineIntel 206C2 family 6 model 44 step 2 clock 3458 MHz)	Oracle Solaris 11	24
LZ	Oracle/SUN X6270 m2	x86 (GenuineIntel 206C2 family 6 model 44 step 2 clock 3458 MHz)	Oracle Solaris 11	24
ZP	Oracle/SUN X6270 m2	x86 (GenuineIntel 206C2 family 6 model 44 step 2 clock 3458 MHz)	Oracle Solaris 11	24
BC	Oracle/SUN X4-2B	x86 (GenuineIntel 306E4 family 6 model 62 step 4 clock 2693 MHz)	Oracle Solaris 11	48
BOS	Oracle/SUN X4-2B	x86 (GenuineIntel 306E4 family 6 model 62 step 4 clock 2693 MHz)	Oracle Solaris 11	48
CHI	Oracle/SUN X4-2B	x86 (GenuineIntel 306E4 family 6 model 62 step 4 clock 2693 MHz)	Oracle Solaris 11	48
FOG	Oracle/SUN X4-2B	x86 (GenuineIntel 306E4 family 6 model 62 step 4 clock 2693 MHz)	Oracle Solaris 11	48
HP	Oracle/SUN X4-2B	x86 (GenuineIntel 306E4 family 6 model 62 step 4 clock 2693 MHz)	Oracle Solaris 11	48
JA	Oracle/SUN X4-2B	x86 (GenuineIntel 306E4 family 6 model 62 step 4 clock 2693 MHz)	Oracle Solaris 11	48
ML	Oracle/SUN X4-2B	x86 (GenuineIntel 306E4 family 6 model 62 step 4 clock 2693 MHz)	Oracle Solaris 11	48
RE	Oracle/SUN X4-2B	x86 (GenuineIntel 306E4 family 6 model 62 step 4 clock 2693 MHz)	Oracle Solaris 11	48
UH	Oracle/SUN X4-2B	x86 (GenuineIntel 306E4 family 6 model 62 step 4 clock 2693 MHz)	Oracle Solaris 11	48
ZZ	Oracle/SUN X4-2B	x86 (GenuineIntel 306E4 family 6 model 62 step 4 clock 2693 MHz)	Oracle Solaris 11	48
SAN	Dell PowerEdge R820	x86 (GenuineIntel 206D7 family 6 model 45 step 7 clock 2400 MHz)	Oracle Solaris 11	64

3.0 WIPP PA SOFTWARE CONFIGURATION MANAGEMENT AND RUN CONTROL SYSTEMS

The computer simulations that form the core of the CRA-2019 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 code, 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 Solaris 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.

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 (Concurrent Versions System) 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 NP 19-1 (Long 2017b). These may include:

- source code
- 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 Solaris 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 Performance Assessment Parameter Database (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 ID
- 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

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-2019 calculations (CRA19 or CRA19_CL)
- 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 ID which is sometimes followed by the other code specific items used to give meaning to file names.

A file with the name GM_BF_CRA19.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.
CRA19	This is the calculation ID.
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 (Performance Assessment Parameter Database) 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 uncertainty distributions. See the PAPDB Design Document (Long 2019) for more information. PAPDB Version 3.00 was used for the CRA-2019.

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-2019.

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 2017a).

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 Solaris 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-2019 analysis are listed in Table 4. These codes will be executed on the WIPP PA Solaris 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 2017b).

Table 4 – WIPP PA Codes Used for the CRA-2019 PA

Code	Version	Executable[1]	Build Date
ALGEBRACDB	2.36	algebracdb	9/11/12
BRAGFLO	7.00	bragflo	8/14/18
CCDFGF	7.03	ccdfgf	5/3/17
CCDFVECTORSTATS	1.01	ccdfvectorstats	3/20/18
CUTTINGS_S	6.03	cuttings_s	1/15/13
EPAUNI	1.19	epauni	9/12/16
GENMESH	6.10	genmesh	1/12/15
ICSET	2.23	icset	9/11/12
LHS	2.44	lhs	6/2/15
MATSET	9.24	matset	10/11/16
NUTS	2.07	nuts	2/22/19
PANEL	5.00	panel	2/18/19
POSTBRAG	4.02	postbrag	1/10/13
POSTLHS	4.11	postlhs	6/2/16
PREBRAG	9.00	prebrag	8/16/18
PRECCDFGF	2.01	preccdfgf	9/9/13
PRELHS	2.44	prelhs	10/11/16
RELATE	1.45	relate	9/11/12
SCREEN_NUTS	1.02	screen_nuts	2/7/18
STEPWISE	2.22	stepwise	7/2/13
SUMMARIZE	3.02	summarize	10/31/12

[1] Executables are located in $CVSLIB/WIPP_CODES/PA_CODES/*CODE*/Build/Solaris

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 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/CRA19/EPAUNI	Python run control script
RunControl/EPAUNIlib.py	$REP1/CRA19/EPAUNI	Python run control script class modules
RunControl/rc.py	$REP1/CRA19/EPAUNI	Run control module
RunControl/Run.py	$REP1/CRA19/EPAUNI	Main control script

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

Table 6 – The EPAUNI input files used were:

File	Repository	Comment
Input/epu_CRA19_ch.inp	$REP1/CRA19/EPAUNI	Input file
Input/epu_CRA19_ch_misc.inp	$REP1/CRA19/EPAUNI	Input file
Input/epu_CRA19_rh.inp	$REP1/CRA19/EPAUNI	Input file
Input/epu_CRA19_rh_misc.inp	$REP1/CRA19/EPAUNI	Input file

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

Table 7 – The EPAUNI CVS repositories used were:

Repository
$CODE/EPAUNI
$REP1/CRA19/EPAUNI

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

Table 8 – The EPAUNI log files used were:

File	Repository	Comment
RunControl/EPAUNI.log	$REP1/CRA19/EPAUNI	Log file
RunControl/EPAUNI.rtf	$REP1/CRA19/EPAUNI	Formatted log file (Word file)

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

Table 9 – The EPAUNI output files produced were:

File	Repository	Comment
Output/epu_CRA19_ch.dat	$REP1/CRA19/EPAUNI	Results file
Output/epu_CRA19_ch.dia	$REP1/CRA19/EPAUNI	Diagnostic file (units)
Output/epu_CRA19_ch.out	$REP1/CRA19/EPAUNI	Results file (times)
Output/epu_CRA19_ch.out2	$REP1/CRA19/EPAUNI	Results file (format 2)
Output/epu_CRA19_ch_activity.dia	$REP1/CRA19/EPAUNI	Diagnostic file (activity)
Output/epu_CRA19_rh.dat	$REP1/CRA19/EPAUNI	Results file
Output/epu_CRA19_rh.dia	$REP1/CRA19/EPAUNI	Diagnostic file (units)
Output/epu_CRA19_rh.out	$REP1/CRA19/EPAUNI	Results file (times)
Output/epu_CRA19_rh.out2	$REP1/CRA19/EPAUNI	Results file (format 2)
Output/epu_CRA19_rh_activity.dia	$REP1/CRA19/EPAUNI	Diagnostic file (activity)

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

Table 10 – The EPAUNI executable file used was:

File	Repository	Comment
Build/Solaris/epauni (Ver:1.19)	$CODE/EPAUNI	Computes decay of radionuclide components in inventory

Where:
$CODE = /nfs/data/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/CRA19/LHS	Python run control script
RunControl/LHSlib.py	$REP1/CRA19/LHS	Python run control script class modules
RunControl/rc.py	$REP1/CRA19/LHS	Run control module
RunControl/Run.py	$REP1/CRA19/LHS	Main control script

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

Table 12 – The LHS input file used was:

File	Repository	Comment
Input/lhs1_CRA19_r*i*_con.inp	$REP1/CRA19/PRELHS	Input file

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

Table 13 – The LHS CVS repositories used were:

Repository
$CODE/LHS
$CODE/PRELHS
$REP1/CRA19/LHS
$REP1/CRA19/PRELHS

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

Table 14 – The LHS log files used were:

File	Repository	Comment
RunControl/LHS.log	$REP1/CRA19/LHS	Log file
RunControl/LHS.rtf	$REP1/CRA19/LHS	Formatted log file (Word file)

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

Table 15 – The LHS output files produced were:

File	Repository	Comment
Output/lhs1_CRA19_*ri*_con.dbg	$REP1/CRA19/PRELHS	Debug file
Output/lhs1_CRA19_*ri*_con.trn	$REP1/CRA19/PRELHS	Transfer file
Output/lhs2_CRA19_*ri*_con.dbg	$REP1/CRA19/LHS	Debug file
Output/lhs2_CRA19_*ri*_con.trn	$REP1/CRA19/LHS	Transfer file

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

Table 16 – The LHS executable files used were:

File	Repository	Comment
Build/Solaris/lhs (Ver:2.44)	$CODE/LHS	Code to sample uncertain parameters
Build/Solaris/prelhs (Ver:2.44)	$CODE/PRELHS	Pre-processes data for LHS

Where:
$CODE = /nfs/data/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 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.

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/CRA19/BRAGFLO	Python run control script
RunControl/BRAGFLOlib.py	$REP1/CRA19/BRAGFLO	Python run control script class modules
RunControl/rc.py	$REP1/CRA19/BRAGFLO	Run control module
RunControl/Run.py	$REP1/CRA19/BRAGFLO	Main control script

Where:

$REP1 = /nfs/data/CVSLIB/WIPP_ANALYSES

Table 19 – The BRAGFLO input files used were:

File	Repository	Comment
Input/alg1_bf_CRA19.inp	$REP1/CRA19/ALGEBRACDB	Input file
Input/alg2_bf_CRA19.inp	$REP1/CRA19/ALGEBRACDB	Input file
Input/bf1_CRA19_s*n*.inp	$REP1/CRA19/PREBRAG	Input file
Input/bf1_CRA19_s*n*_mod1.inp	$REP1/CRA19/PREBRAG	Input file
Input/bf1_CRA19_s*n*_mod2.inp	$REP1/CRA19/PREBRAG	Input file
Input/bf2_CRA19_closure.dat	$REP1/CRA19/BRAGFLO	Input file
Input/gm_bf_CRA19.inp	$REP1/CRA19/GENMESH	Input file
Input/ic_bf_CRA19.inp	$REP1/CRA19/ICSET	Input file
Input/ms_bf_CRA19.inp	$REP1/CRA19/MATSET	Input file

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

Table 20 – The BRAGFLO CVS repositories used were:

Repository
$CODE/ALGEBRACDB
$CODE/BRAGFLO
$CODE/GENMESH
$CODE/ICSET
$CODE/MATSET
$CODE/POSTBRAG
$CODE/POSTLHS
$CODE/PREBRAG
$REP1/CRA19/ALGEBRACDB
$REP1/CRA19/BRAGFLO
$REP1/CRA19/GENMESH
$REP1/CRA19/ICSET
$REP1/CRA19/MATSET
$REP1/CRA19/POSTLHS
$REP1/CRA19/PREBRAG

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

Table 21 – The BRAGFLO log files used were:

File	Repository	Comment
RunControl/BRAGFLO.log	$REP1/CRA19/BRAGFLO	Log file
RunControl/BRAGFLO.rtf	$REP1/CRA19/BRAGFLO	Formatted log file (Word file)

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

Table 22 – The BRAGFLO output files produced were:

File	Repository	Comment
alg1_bf_CRA19_r*i*_v*vvv*.cdb		Not Saved:CDB transfer file
alg2_bf_CRA19_r*i*_s*n*_v*vvv*.cdb		Not Saved:CDB transfer file
Output/bf2_CRA19_r*i*_s*n*_v*vvv*.inp	$REP1/CRA19/PREBRAG	Input file
Output/bf2_CRA19_r*i*_s*n*_v*vvv*.log	$REP1/CRA19/BRAGFLO	Log file
Output/bf2_CRA19_r*i*_s*n*_v*vvv*.sum	$REP1/CRA19/BRAGFLO	Summary file
bf3_CRA19_r*i*_s*n*_v*vvv*.cdb		Not Saved:CDB transfer file
gm_bf_CRA19.cdb		Not Saved:CDB transfer file
ic_bf_CRA19_r*i*_v*vvv*.cdb		Not Saved:CDB transfer file
lhs3_bf_CRA19_r*i*_v*vvv*.cdb		Not Saved:CDB transfer file
ms_bf_CRA19.cdb		Not Saved:CDB transfer file

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

Table 23 – The BRAGFLO executable files used were:

File	Repository	Comment
Build/Solaris/algebracdb (Ver:2.36)	$CODE/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Solaris/bragflo (Ver:7.00)	$CODE/BRAGFLO	Computes brine and gas flow in the repository
Build/Solaris/genmesh (Ver:6.10)	$CODE/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/icset (Ver:2.23)	$CODE/ICSET	Assigns initial conditions to the CAMDAT grid elements
Build/Solaris/matset (Ver:9.24)	$CODE/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postbrag (Ver:4.02)	$CODE/POSTBRAG	Post-processes data for bragflo
Build/Solaris/postlhs (Ver:4.11)	$CODE/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/prebrag (Ver:9.00)	$CODE/PREBRAG	Pre-processes data for bragflo

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

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 CRA19.

Table 24 – The PANEL run script files used were:

File	Repository	Comment
RunControl/PANEL.py	$REP1/CRA19/PANEL	Python run control script
RunControl/PANELlib.py	$REP1/CRA19/PANEL	Python run control script class modules
RunControl/rc.py	$REP1/CRA19/PANEL	Run control module
RunControl/Run.py	$REP1/CRA19/PANEL	Main control script

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

Table 25 – The PANEL input files used were:

File	Repository	Comment
Input/alg1_panel_CRA19.inp	$REP1/CRA19/ALGEBRACDB	Input file
Output/alg2_bf_CRA19_*ri*_*sn*_*vvvv*.cdb		Not Saved:CDB transfer file
Input/alg2_panel_CRA19_b1.inp	$REP1/CRA19/ALGEBRACDB	Input file
Input/alg2_panel_CRA19_b2.inp	$REP1/CRA19/ALGEBRACDB	Input file
Input/alg2_panel_CRA19_b3.inp	$REP1/CRA19/ALGEBRACDB	Input file
Input/alg2_panel_CRA19_b4.inp	$REP1/CRA19/ALGEBRACDB	Input file
Input/alg2_panel_CRA19_b5.inp	$REP1/CRA19/ALGEBRACDB	Input file
Input/alg3_panel_CRA19_b1.inp	$REP1/CRA19/ALGEBRACDB	Input file
Input/alg3_panel_CRA19_b2.inp	$REP1/CRA19/ALGEBRACDB	Input file
Input/alg3_panel_CRA19_b3.inp	$REP1/CRA19/ALGEBRACDB	Input file
Input/alg3_panel_CRA19_b4.inp	$REP1/CRA19/ALGEBRACDB	Input file
Input/alg3_panel_CRA19_b5.inp	$REP1/CRA19/ALGEBRACDB	Input file
Input/gm_panel_CRA19.inp	$REP1/CRA19/GENMESH	Input file
Input/ms_panel_CRA19.inp	$REP1/CRA19/MATSET	Input file
Input/sum_panel_con.inp	$REP1/CRA19/SUMMARIZE	Input file
Input/sum_panel_int.inp	$REP1/CRA19/SUMMARIZE	Input file
Input/sum_panel_st.inp	$REP1/CRA19/SUMMARIZE	Input file

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

Table 26 – The PANEL CVS repositories used were:

Repository
$CODE/ALGEBRACDB
$CODE/GENMESH
$CODE/MATSET
$CODE/PANEL
$CODE/POSTLHS
$CODE/SUMMARIZE
$REP1/CRA19/ALGEBRACDB
$REP1/CRA19/GENMESH
$REP1/CRA19/MATSET
$REP1/CRA19/PANEL
$REP1/CRA19/POSTLHS
$REP1/CRA19/SUMMARIZE

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

Table 27 – The PANEL log files used were:

File	Repository	Comment
RunControl/PANEL.log	$REP1/CRA19/PANEL	Log file
RunControl/PANEL.rtf	$REP1/CRA19/PANEL	Formatted log file (Word file)

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

Table 28 – The PANEL output files produced were:

File	Repository	Comment
alg1_panel_CRA19.cdb		Not Saved:CDB transfer file
alg2_panel_CRA19_b1.cdb		Not Saved:CDB transfer file
alg2_panel_CRA19_b2.cdb		Not Saved:CDB transfer file
alg2_panel_CRA19_b3.cdb		Not Saved:CDB transfer file
alg2_panel_CRA19_b4.cdb		Not Saved:CDB transfer file
alg2_panel_CRA19_b5.cdb		Not Saved:CDB transfer file
alg3_panel_CRA19_b1_*rj*_*vwww*.cdb		Not Saved:CDB transfer file

alg3_panel_CRA19_b2_*rj*_*vwww*.cdb		Not Saved:CDB transfer file
alg3_panel_CRA19_b3_*rj*_*vwww*.cdb		Not Saved:CDB transfer file
alg3_panel_CRA19_b4_*rj*_*vwww*.cdb		Not Saved:CDB transfer file
alg3_panel_CRA19_b5_*rj*_*vwww*.cdb		Not Saved:CDB transfer file
gm_panel_CRA19.cdb		Not Saved:CDB transfer file
lhs3_panel_CRA19_b1_*rj*_*vwww*.cdb		Not Saved:CDB transfer file
lhs3_panel_CRA19_b2_*rj*_*vwww*.cdb		Not Saved:CDB transfer file
lhs3_panel_CRA19_b3_*rj*_*vwww*.cdb		Not Saved:CDB transfer file
lhs3_panel_CRA19_b4_*rj*_*vwww*.cdb		Not Saved:CDB transfer file
lhs3_panel_CRA19_b5_*rj*_*vwww*.cdb		Not Saved:CDB transfer file
ms_panel_CRA19.cdb		Not Saved:CDB transfer file
panel_con_CRA19_b1_*rj*_*sq*_*vwww*.cdb		Not Saved:CDB transfer file
panel_con_CRA19_b2_*rj*_*sq*_*vwww*.cdb		Not Saved:CDB transfer file
panel_con_CRA19_b3_*rj*_*sq*_*vwww*.cdb		Not Saved:CDB transfer file
panel_con_CRA19_b4_*rj*_*sq*_*vwww*.cdb		Not Saved:CDB transfer file
panel_con_CRA19_b5_*rj*_*sq*_*vwww*.cdb		Not Saved:CDB transfer file
panel_decay_CRA19_*ri*_*sn*_*vvvv*.cdb		Not Saved:CDB transfer file
panel_int_CRA19_b1_*rj*_*so*_*tttttt*_*vwww*.cdb		Not Saved:CDB transfer file
panel_int_CRA19_b2_*rj*_*so*_*tttttt*_*vwww*.cdb		Not Saved:CDB transfer file
panel_int_CRA19_b3_*rj*_*so*_*tttttt*_*vwww*.cdb		Not Saved:CDB transfer file
panel_int_CRA19_b4_*rj*_*so*_*tttttt*_*vwww*.cdb		Not Saved:CDB transfer file
panel_int_CRA19_b5_*rj*_*so*_*tttttt*_*vwww*.cdb		Not Saved:CDB transfer file

Output/sum_panel_con_CRA19_b1_r*j*_s*p*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_con_CRA19_b2_r*j*_s*p*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_con_CRA19_b3_r*j*_s*p*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_con_CRA19_b4_r*j*_s*p*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_con_CRA19_b5_r*j*_s*p*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_int_CRA19_b1_r*j*_s*o*_t*tttt*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_int_CRA19_b2_r*j*_s*o*_t*tttt*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_int_CRA19_b3_r*j*_s*o*_t*tttt*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_int_CRA19_b4_r*j*_s*o*_t*tttt*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_int_CRA19_b5_r*j*_s*o*_t*tttt*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_st_CRA19_b1_r*j*_s*p*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_st_CRA19_b2_r*j*_s*p*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_st_CRA19_b3_r*j*_s*p*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_st_CRA19_b4_r*j*_s*p*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_st_CRA19_b5_r*j*_s*p*.tbl	$REP1/CRA19/SUMMARIZE	Table file

Where:
i is 1
j is 1-3
n is 1
o is 6
p is 1-2
q is 1-6
tttt is 00100, 00350, 01000, 02000, 04000, 06000, 09000
vvv is 001
www is 001-100
$REP1 = /nfs/data/CVSLIB/WIPP_ANALYSES

Table 29 – The PANEL executable files used were:

File	Repository	Comment
Build/Solaris/algebracdb (Ver:2.36)	$CODE/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Solaris/genmesh (Ver:6.10)	$CODE/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/matset (Ver:9.24)	$CODE/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/panel (Ver:5.00)	$CODE/PANEL	Computes release concentrations of nuclides from repository
Build/Solaris/postlhs (Ver:4.11)	$CODE/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/summarize (Ver:3.02)	$CODE/SUMMARIZE	Writes tables of data from many CAMDAT files

Where:
$CODE = /nfs/data/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 CRA19.

Table 30 – The NUTS run script files used were:

File	Repository	Comment
RunControl/NUTS.py	$REP1/CRA19/NUTS	Python run control script
RunControl/NUTSlib.py	$REP1/CRA19/NUTS	Python run control script class modules
RunControl/rc.py	$REP1/CRA19/NUTS	Run control module
RunControl/Run.py	$REP1/CRA19/NUTS	Main control script

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

Table 31 – The NUTS input files used were:

File	Repository	Comment
Input/alg_nut_iso_CRA19.inp	$REP1/CRA19/ALGEBRACDB	Input file
Input/alg_nut_scn_CRA19.inp	$REP1/CRA19/ALGEBRACDB	Input file
Output/bf2_CRA19_*ri*_*sn*_*vvvv*.inp	$REP1/CRA19/PREBRAG	Input file
bf3_CRA19_*ri*_*sn*_*vvvv*.cdb		Not Saved:CDB transfer file
Input/ms_nut_CRA19.inp	$REP1/CRA19/MATSET	Input file
Input/nut_int_CRA19_*so*_*ttttt*.inp	$REP1/CRA19/NUTS	Input file
Input/nut_iso_CRA19_*sn*.inp	$REP1/CRA19/NUTS	Input file
Input/nut_scn_CRA19_*sn*.inp	$REP1/CRA19/NUTS	Input file
panel_con_CRA19_b1_*ri*_*sn*_*vvvv*.cdb		Not Saved:CDB transfer file

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

Table 32 – The NUTS CVS repositories used were:

Repository
$CODE/ALGEBRACDB
$CODE/MATSET
$CODE/NUTS
$CODE/SCREEN_NUTS
$CODE/SUMMARIZE
$REP1/CRA19/ALGEBRACDB
$REP1/CRA19/BRAGFLO
$REP1/CRA19/MATSET
$REP1/CRA19/NUTS
$REP1/CRA19/PANEL
$REP1/CRA19/PREBRAG
$REP1/CRA19/SCREEN_NUTS
$REP1/CRA19/SUMMARIZE

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

Table 33 – The NUTS log files used were:

File	Repository	Comment
RunControl/NUTS.log	$REP1/CRA19/NUTS	Log file
RunControl/NUTS.rtf	$REP1/CRA19/NUTS	Formatted log file (Word file)

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

Table 34 – The NUTS output files produced were:

File	Repository	Comment
alg_nut_int_CRA19_r*i*_s*o*_t*tttt*_V*VVV*.cdb		Not Saved:CDB transfer file
alg_nut_iso_CRA19_r*i*_s*n*_V*VVV*.cdb		Not Saved:CDB transfer file
alg_nut_scn_CRA19_r*i*_s*n*_v*vvv*.cdb		Not Saved:CDB transfer file
ms_nut_CRA19_r*i*_s*n*_V*VVV*.cdb		Not Saved:CDB transfer file
nut_int_CRA19_r*i*_s*o*_t*tttt*_V*VVV*.cdb		Not Saved:CDB transfer file
nut_iso_CRA19_r*i*_s*n*_V*VVV*.cdb		Not Saved:CDB transfer file
nut_scn_CRA19_r*i*_s*n*_v*vvv*.cdb		Not Saved:CDB transfer file
Output/screen_nut_scn_CRA19_r*i*_EDIT.inp	$REP1/CRA19/SCREEN_NUTS	Input file
Output/screen_nut_scn_CRA19_r*i*_s*n*.out[2]	$REP1/CRA19/SCREEN_NUTS	Output file
Output/sum_nut_CRA19_r*i*_s*n*_t*uuuuu*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_nut_scn_CRA19_r*i*_s*n*.tbl	$REP1/CRA19/SUMMARIZE	Table file

Where:

i is 1-3
n is 1-5
o is 2-5
tttt is 0100 for S2, S4
03000, 05000, 07000, 09000 for S3, S5
uuuuu is 0100 for S1
00100, 00350 for S2,S4
01000, 03000, 05000, 07000, 09000 for S3, S5
vvv is 001-100
$REP1 = /nfs/data/CVSLIB/WIPP_ANALYSES
VVV are the screened-in vectors listed in Table 35.

[2] Contains a list of screened-in vectors.

Table 35 – The CRA19 NUTS screened-in vectors were:

Replicate	Scenario	Vectors
1	1	1,2,5,6,7,8,9,10,12,13,14,17,19,20,22,23,24,25,26,27,28,29,30,34,35,36,38,41,43,44,45,46,47,49,50,51,52,53,54,55,58,59,60,61,62,63,64,66,67,69,70,71,72,76,78,79,80,82,83,84,86,88,89,92,93,96,98
1	2	1,2,5,6,7,8,9,10,12,13,14,17,19,20,22,23,24,25,26,27,28,29,30,34,35,36,38,41,43,44,45,46,47,49,50,51,52,53,54,55,58,59,60,61,62,63,64,66,67,69,70,71,72,76,78,79,80,82,83,84,86,88,89,92,93,96,98
1	3	1,2,5,6,7,8,9,12,13,14,17,20,22,23,24,25,26,27,28,29,30,34,35,36,38,41,43,44,45,46,47,49,50,51,52,53,54,55,59,60,61,62,63,64,66,67,70,71,76,78,80,82,83,84,86,88,89,93,96,98
1	4	7,9,30,36,45,50,53,55
1	5	7,9,30,36,45,50,53,55
2	1	1,2,3,4,6,8,9,12,14,16,17,18,19,21,24,25,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,43,44,45,48,49,50,51,52,53,54,55,56,59,63,64,65,66,67,68,71,72,74,75,77,79,80,83,84,87,89,90,92,94,95,96,98,99,100
2	2	1,2,3,4,6,8,9,12,14,16,17,18,19,21,24,25,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,43,44,45,48,49,50,51,52,53,54,55,56,59,63,64,65,66,67,68,71,72,74,75,77,79,80,83,84,87,89,90,92,94,95,96,98,99,100
2	3	3,4,6,8,9,12,14,16,18,19,21,24,25,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,43,44,45,48,49,50,51,52,53,54,55,56,59,63,65,67,68,71,74,75,77,79,80,84,87,89,90,92,94,95,96,98,99,100
2	4	4,17,21,24,28,34,68
2	5	4,17,21,24,28,34,68
3	1	2,3,7,10,11,13,14,15,16,17,18,21,22,24,25,26,27,28,30,32,33,34,37,38,39,40,41,42,43,44,45,47,49,50,52,53,56,57,58,59,61,62,63,64,65,66,67,68,69,70,71,73,74,75,77,78,79,81,83,84,86,88,89,90,91,93,94,96,97,99,100
3	2	2,3,7,10,11,13,14,15,16,17,18,21,22,24,25,26,27,28,30,32,33,34,37,38,39,40,41,42,43,44,45,47,49,50,52,53,56,57,58,59,61,62,63,64,65,66,67,68,69,70,71,73,74,75,77,78,79,81,83,84,86,88,89,90,91,93,94,96,97,99,100
3	3	2,3,7,10,11,14,18,21,22,26,27,30,32,33,34,37,39,40,42,43,44,45,47,49,50,53,56,57,58,59,61,62,63,64,65,66,67,68,69,70,71,73,74,75,77,78,79,84,86,88,89,90,91,93,94,96,97,99,100
3	4	37,42,47,66,86,91,93
3	5	42,47,66,86,91,93

Table 36 – The NUTS executable files used were:

File	Repository	Comment
Build/Solaris/algebracdb (Ver:2.36)	$CODE/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Solaris/matset (Ver:9.24)	$CODE/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/nuts (Ver:2.07)	$CODE/NUTS	Nuclide Transport system model
Build/Solaris/screen_nuts (Ver:1.02)	$CODE/SCREEN_NUTS	Executable file
Build/Solaris/summarize (Ver:3.02)	$CODE/SUMMARIZE	Writes tables of data from many CAMDAT files

Where:

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

Table 377 – The CRA19_CL NUTS screened-in vectors were:

Replicate	Scenario	Vectors
1	1	1,2,3,5,6,7,8,9,10,12,13,14,17,19,20,22,23,24,25,26,27,28,29,30,34,35,36,41,43,44,45,46,47,50,51,52,53,54,55,58,59,60,61,62,63,64,66,67,69,70,71,76,77,78,79,80,82,83,84,86,88,89,90,93,95,96,98
1	2	1,2,3,5,6,7,8,9,10,12,13,14,17,19,20,22,23,24,25,26,27,28,29,30,34,35,36,41,43,44,45,46,47,50,51,52,53,54,55,58,59,60,61,62,63,64,66,67,69,70,71,76,77,78,79,80,82,83,84,86,88,89,90,93,95,96,98
1	3	1,2,3,5,6,7,9,12,13,14,17,19,20,22,23,24,25,26,27,28,29,30,34,35,36,41,43,44,45,46,47,50,51,52,54,55,58,59,60,61,62,63,64,66,67,70,76,77,78,79,80,82,83,84,88,89,90,93,95,96,98
1	4	7,9,22,30,36,50,53
1	5	7,9,22,36,50,53
2	1	2,3,4,6,8,9,10,11,12,14,16,17,18,19,21,23,24,25,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,43,44,45,48,49,50,51,52,53,54,55,56,58,59,63,65,66,67,68,69,71,72,74,75,77,79,80,81,84,86,87,88,89,90,92,94,95,96,98,99,100
2	2	2,3,4,6,8,9,10,11,12,14,16,17,18,19,21,23,24,25,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,43,44,45,48,49,50,51,52,53,54,55,56,58,59,63,65,67,68,69,71,72,74,75,77,79,80,81,84,86,87,88,89,90,92,94,95,96,98,99,100
2	3	2,4,6,8,9,10,12,14,16,17,18,19,21,24,25,27,28,29,30,32,33,34,35,36,37,38,39,40,41,43,44,45,48,49,50,51,52,54,55,59,63,65,66,67,68,69,71,74,75,77,79,80,81,84,87,89,90,95,96,98,99,100
2	4	4,17,21,24,28,34,40,68
2	5	4,17,21,24,28,34,40,68
3	1	2,3,5,7,10,11,13,14,17,18,21,22,24,25,26,27,28,30,32,33,34,35,37,39,40,42,43,44,45,46,47,49,50,51,52,53,56,57,58,59,61,62,63,64,65,66,67,68,69,70,71,73,74,75,76,77,78,79,81,82,83,84,86,88,89,90,91,93,94,96,97,98,99,100
3	2	2,3,5,7,10,11,13,14,17,18,21,22,24,25,26,27,28,30,32,33,34,35,37,39,40,42,43,44,45,46,47,49,50,51,52,53,56,57,58,59,61,62,63,64,65,66,67,68,69,70,71,73,74,75,76,77,78,79,81,82,83,84,86,88,89,90,91,93,94,96,97,98,99,100
3	3	2,3,5,7,10,11,14,18,21,22,24,26,27,30,32,33,34,35,37,39,40,42,43,44,45,46,47,49,50,51,52,53,56,57,58,59,61,62,63,64,65,66,67,68,69,70,71,74,75,76,77,78,83,86,88,89,90,91,93,94,96,97,98,99
3	4	30,37,42,66,91,93
3	5	30,37,42,66,91,93

5.6 Single-Intrusion Spallings Volume Calculations (DRSPALL)

DRSPALL Version 1.22 was not rerun for CRA19. 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 CRA19 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 38 – The CUTTINGS_S run script files used were:

File	Repository	Comment
RunControl/CUTTINGS_S.py	$REP1/CRA19/CUTTINGS_S	Python run control script
RunControl/CUTTINGS_Slib.py	$REP1/CRA19/CUTTINGS_S	Python run control script class modules
RunControl/rc.py	$REP1/CRA19/CUTTINGS_S	Run control module
RunControl/Run.py	$REP1/CRA19/CUTTINGS_S	Main control script

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

Table 39 – The CUTTINGS_S input files used were:

File	Repository	Comment
bf3_CRA19_*ri*_*sn*_*vvvv*.cdb		Not Saved:CDB transfer file
Input/cusp_CRA19.inp	$REP1/CRA19/CUTTINGS_S	Input file
Input/gm_cusp_CRA19.inp	$REP1/CRA19/GENMESH	Input file
Input/ms_cusp_CRA19.inp	$REP1/CRA19/MATSET	Input file
Output/mspall_drs_PABC09_*ri*.out	$REP1/PABC09/DRSPALL	Input file

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

Table 40 – The CUTTINGS_S CVS repositories used were:

Repository
$CODE/CUTTINGS_S
$CODE/GENMESH
$CODE/MATSET
$CODE/POSTLHS
$REP1/CRA19/BRAGFLO
$REP1/CRA19/CUTTINGS_S
$REP1/CRA19/GENMESH
$REP1/CRA19/MATSET
$REP1/CRA19/POSTLHS
$REP1/PABC09/DRSPALL

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

Table 41 – The CUTTINGS_S log files used were:

File	Repository	Comment
RunControl/CUTTINGS_S.log	$REP1/CRA19/CUTTINGS_S	Log file
RunControl/CUTTINGS_S.rtf	$REP1/CRA19/CUTTINGS_S	Formatted log file (Word file)

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

Table 42 – The CUTTINGS_S output files produced were:

File	Repository	Comment
Output/cusp_CRA19_master_*ri*.inp	$REP1/CRA19/CUTTINGS_S	Input file
Output/cusp_CRA19_*ri*.tbl	$REP1/CRA19/CUTTINGS_S	Table file
cusp_CRA19_*ri*_*sn*_*ttttt*_L_*vvvv*.cdb		Not Saved:CDB transfer file
cusp_CRA19_*ri*_*sn*_*ttttt*_M_*vvvv*.cdb		Not Saved:CDB transfer file
cusp_CRA19_*ri*_*sn*_*ttttt*_U_*vvvv*.cdb		Not Saved:CDB transfer file
gm_cusp_CRA19.cdb		Not Saved:CDB transfer file
lhs3_cusp_CRA19_*ri*_*vvvv*.cdb		Not Saved:CDB transfer file
ms_cusp_CRA19.cdb		Not Saved:CDB transfer 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 = /nfs/data/CVSLIB/WIPP_ANALYSES

Table 43 – The CUTTINGS_S executable files used were:

File	Repository	Comment
Build/Solaris/cuttings_s (Ver:6.03)	$CODE/CUTTINGS_S	Computes cuttings/spall generated by drilling
Build/Solaris/genmesh (Ver:6.10)	$CODE/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/matset (Ver:9.24)	$CODE/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postlhs (Ver:4.11)	$CODE/POSTLHS	Assigns sampled parameters to the grid blocks and elements

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

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 44 – The BRAGFLO_DBR run script files used were:

File	Repository	Comment
RunControl/BRAGFLO_DBR.py	$REP1/CRA19/BRAGFLO_DBR	Python run control script
RunControl/BRAGFLO_DBRlib.py	$REP1/CRA19/BRAGFLO_DBR	Python run control script class modules
RunControl/rc.py	$REP1/CRA19/BRAGFLO_DBR	Run control module
RunControl/Run.py	$REP1/CRA19/BRAGFLO_DBR	Main control script

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

Table 45 – The BRAGFLO_DBR input files used were:

File	Repository	Comment
Input/alg1_dbr_CRA19.inp	$REP1/CRA19/ALGEBRACDB	Input file
Input/alg2_dbr_CRA19_*so*.inp	$REP1/CRA19/ALGEBRACDB	Input file
Input/alg3_dbr_CRA19_L.inp	$REP1/CRA19/ALGEBRACDB	Input file
Input/alg3_dbr_CRA19_M.inp	$REP1/CRA19/ALGEBRACDB	Input file
Input/alg3_dbr_CRA19_U.inp	$REP1/CRA19/ALGEBRACDB	Input file
Input/bf1_dbr_CRA19_L.inp	$REP1/CRA19/PREBRAG	Input file
Input/bf1_dbr_CRA19_M.inp	$REP1/CRA19/PREBRAG	Input file
Input/bf1_dbr_CRA19_*sn*_100_L.inp	$REP1/CRA19/PREBRAG	Input file
Input/bf1_dbr_CRA19_*sn*_100_M.inp	$REP1/CRA19/PREBRAG	Input file
Input/bf1_dbr_CRA19_*sn*_100_U.inp	$REP1/CRA19/PREBRAG	Input file
Input/bf1_dbr_CRA19_U.inp	$REP1/CRA19/PREBRAG	Input file
bf3_CRA19_*ri*_*so*_*vvvv*.cdb		Not Saved:CDB transfer file
cusp_CRA19_*ri*_*so*_*ttttt*_L_*vvvv*.cdb		Not Saved:CDB transfer file
cusp_CRA19_*ri*_*so*_*ttttt*_M_*vvvv*.cdb		Not Saved:CDB transfer file
cusp_CRA19_*ri*_*so*_*ttttt*_U_*vvvv*.cdb		Not Saved:CDB transfer file
Input/gm_dbr_CRA19.inp	$REP1/CRA19/GENMESH	Input file
Input/ic_dbr_CRA19_*so*.inp	$REP1/CRA19/ICSET	Input file
Input/ms_dbr_CRA19.inp	$REP1/CRA19/MATSET	Input file
Input/rel1_dbr_CRA19.inp	$REP1/CRA19/RELATE	Input file
Input/rel2_dbr_CRA19_*so*.inp	$REP1/CRA19/RELATE	Input file
Input/sum_dbr.inp	$REP1/CRA19/SUMMARIZE	Input file

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 = /nfs/data/CVSLIB/WIPP_ANALYSES

Table 46 – 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/CRA19/ALGEBRACDB
$REP1/CRA19/BRAGFLO
$REP1/CRA19/BRAGFLO_DBR
$REP1/CRA19/CUTTINGS_S
$REP1/CRA19/GENMESH
$REP1/CRA19/ICSET
$REP1/CRA19/MATSET
$REP1/CRA19/PREBRAG
$REP1/CRA19/RELATE
$REP1/CRA19/SUMMARIZE

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

Table 47 – The BRAGFLO_DBR log files used were:

File	Repository	Comment
RunControl/BRAGFLO_DBR.log	$REP1/CRA19/BRAGFLO_DBR	Log file
RunControl/BRAGFLO_DBR.rtf	$REP1/CRA19/BRAGFLO_DBR	Formatted log file (Word file)

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

Table 48 – The BRAGFLO_DBR output files produced were:

File	Repository	Comment
alg1_dbr_CRA19_*ri*_*sn*_*ttttt*_*vvvv*.cdb		Not Saved:CDB transfer file
alg2_dbr_CRA19_*ri*_*sn*_*ttttt*_*vvvv*.cdb		Not Saved:CDB transfer file
alg3_dbr_CRA19_*ri*_*sn*_*ttttt*_L_*vvvv*.cdb		Not Saved:CDB transfer file
alg3_dbr_CRA19_*ri*_*sn*_*ttttt*_M_*vvvv*.cdb		Not Saved:CDB transfer file
alg3_dbr_CRA19_*ri*_*sn*_*ttttt*_U_*vvvv*.cdb		Not Saved:CDB transfer file
Output/bf2_dbr_CRA19_*ri*_*sn*_*ttttt*_L_*vvvv*.inp	$REP1/CRA19/BRAGFLO_DBR	Input file
Output/bf2_dbr_CRA19_*ri*_*sn*_*ttttt*_M_*vvvv*.inp	$REP1/CRA19/BRAGFLO_DBR	Input file
Output/bf2_dbr_CRA19_*ri*_*sn*_*ttttt*_U_*vvvv*.inp	$REP1/CRA19/BRAGFLO_DBR	Input file
bf3_dbr_CRA19_*ri*_*sn*_*ttttt*_L_*vvvv*.cdb		Not Saved:CDB transfer file
bf3_dbr_CRA19_*ri*_*sn*_*ttttt*_M_*vvvv*.cdb		Not Saved:CDB transfer file
bf3_dbr_CRA19_*ri*_*sn*_*ttttt*_U_*vvvv*.cdb		Not Saved:CDB transfer file
gm_dbr_CRA19.cdb		Not Saved:CDB transfer file
ic_dbr_CRA19_*ri*_*sn*_*ttttt*_*vvvv*.cdb		Not Saved:CDB transfer file
ms_dbr_CRA19.cdb		Not Saved:CDB transfer file
rel1_dbr_CRA19_*ri*_*sn*_*ttttt*_*vvvv*.cdb		Not Saved:CDB transfer file
rel2_dbr_CRA19_*ri*_*sn*_*ttttt*_*vvvv*.cdb		Not Saved:CDB transfer file
Output/sum_dbr_CRA19_*ri*_*sn*_*ttttt*_L.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_dbr_CRA19_*ri*_*sn*_*ttttt*_M.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_dbr_CRA19_*ri*_*sn*_*ttttt*_U.tbl	$REP1/CRA19/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 = /nfs/data/CVSLIB/WIPP_ANALYSES

Table 49 – The BRAGFLO_DBR executable files used were:

File	Repository	Comment
Build/Solaris/algebracdb (Ver:2.36)	$CODE/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Solaris/bragflo (Ver:7.00)	$CODE/BRAGFLO	Computes brine and gas flow in the repository
Build/Solaris/genmesh (Ver:6.10)	$CODE/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/icset (Ver:2.23)	$CODE/ICSET	Assigns initial conditions to the CAMDAT grid elements
Build/Solaris/matset (Ver:9.24)	$CODE/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postbrag (Ver:4.02)	$CODE/POSTBRAG	Post-processes data for bragflo
Build/Solaris/postlhs (Ver:4.11)	$CODE/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/prebrag (Ver:9.00)	$CODE/PREBRAG	Pre-processes data for bragflo
Build/Solaris/relate (Ver:1.45)	$CODE/RELATE	Transfers CAMDAT data to another CAMDAT file
Build/Solaris/summarize (Ver:3.02)	$CODE/SUMMARIZE	Writes tables of data from many CAMDAT files

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

5.9 Culebra Transport Calculations (SECOTP2D)

Culebra Transport calculations are identical to those found in the migration of computational codes to the Solaris system (Kirchner et al. 2014, Kirchner et al. 2015) because the parameters and conceptual model was not impacted by any of the updates or corrections implemented in the CRA19 analysis. Consequently, the transport results from the code migration using SECOTP2D version 1.43 (which were based on calculations performed to replicate PABC-2009 calculations; i.e., PABC09, Rev. 1) were used for the CRA-2019 PA.

5.10 CCDF Construction (CCDFGF)

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

Table 50 – The CCDFGF run script files used were:

File	Repository	Comment
RunControl/CCDFGF.py	$REP1/CRA19/CCDFGF	Python run control script
RunControl/CCDFGFlib.py	$REP1/CRA19/CCDFGF	Python run control script class modules
RunControl/rc.py	$REP1/CRA19/CCDFGF	Run control module
RunControl/Run.py	$REP1/CRA19/CCDFGF	Main control script

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

Table 51 – The CCDFGF input files used were:

File	Repository	Comment
Input/ccgf_CRA19_control_*ri*.inp	$REP1/CRA19/CCDFGF	Input file
Output/cusp_CRA19_*ri*.tbl	$REP1/CRA19/CUTTINGS_S	Table file
Output/epu_CRA19_ch.dat	$REP1/CRA19/EPAUNI	Table file
Output/epu_CRA19_rh.dat	$REP1/CRA19/EPAUNI	Table file
Input/gm_ccgf_CRA19.inp	$REP1/CRA19/GENMESH	Input file
Input/intrusiontimes.in	$REP1/CRA19/PRECCDFGF	Input file
Input/ms_ccgf_CRA19.inp	$REP1/CRA19/MATSET	Input file
Output/sum_dbr_CRA19_*ri*_*so*_*tvvvvv*_L.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_dbr_CRA19_*ri*_*so*_*tvvvvv*_M.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_dbr_CRA19_*ri*_*so*_*tvvvvv*_U.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_nut_CRA19_*ri*_*so*_*tuuuuu*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_con_CRA19_b1_*ri*_*sn*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_con_CRA19_b2_*ri*_*sn*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_con_CRA19_b3_*ri*_*sn*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_con_CRA19_b4_*ri*_*sn*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_con_CRA19_b5_*ri*_*sn*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_int_CRA19_b1_*ri*_*sp*_*tttttt*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_st_CRA19_b1_*ri*_*sn*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_st_CRA19_b2_*ri*_*sn*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_st_CRA19_b3_*ri*_*sn*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_st_CRA19_b4_*ri*_*sn*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_panel_st_CRA19_b5_*ri*_*sn*.tbl	$REP1/CRA19/SUMMARIZE	Table file
Output/sum_st2d_PABC09_*ri*_mf.tbl	$REP1/PABC09/SUMMARIZE	Table file
Output/sum_st2d_PABC09_*ri*_mp.tbl	$REP1/PABC09/SUMMARIZE	Table file

Where:

i is 1-3
n is 1-2
o is 1-5
p is 6
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 = /nfs/data/CVSLIB/WIPP_ANALYSES

Table 52 – The CCDFGF CVS repositories used were:

Repository
$CODE/CCDFGF
$CODE/CCDFVECTORSTATS
$CODE/GENMESH
$CODE/MATSET
$CODE/POSTLHS
$CODE/PRECCDFGF
$REP1/CRA19/CCDFGF
$REP1/CRA19/CUTTINGS_S
$REP1/CRA19/EPAUNI
$REP1/CRA19/GENMESH
$REP1/CRA19/MATSET
$REP1/CRA19/POSTLHS
$REP1/CRA19/PRECCDFGF
$REP1/CRA19/SUMMARIZE
$REP1/PABC09/SUMMARIZE

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

Table 53 – The CCDFGF log files used were:

File	Repository	Comment
RunControl/CCDFGF.log	$REP1/CRA19/CCDFGF	Log file
RunControl/CCDFGF.rtf	$REP1/CRA19/CCDFGF	Formatted log file (Word file)

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

Table 54 – The CCDFGF output files produced were:

File	Repository	Comment
Output/ccgf_CRA19_reltab_r*i*.dat	$REP1/CRA19/PRECCDFGF	Results file
Output/ccgf_CRA19_r*i*.out	$REP1/CRA19/CCDFGF	Results file
gm_ccgf_CRA19.cdb		Not Saved:CDB transfer file
lhs3_ccgf_CRA19_r*i*_*vvvv*.cdb		Not Saved:CDB transfer file
ms_ccgf_CRA19.cdb		Not Saved:CDB transfer file

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

Table 55 – The CCDFGF executable files used were:

File	Repository	Comment
Build/Solaris/ccdfgf (Ver:7.03)	$CODE/CCDFGF	Constructs complimentary cumulative distribution functions for radionuclide releases
Build/Solaris/ccdfvectorstats (Ver:1.01)	$CODE/CCDFVECTORSTATS	Executable file
Build/Solaris/genmesh (Ver:6.10)	$CODE/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/matset (Ver:9.24)	$CODE/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postlhs (Ver:4.11)	$CODE/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/preccdfgf (Ver:2.01)	$CODE/PRECCDFGF	Pre-processes data for ccdfgf

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

5.11 Sensitivity Analysis (STEPWISE)

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.

Input files for the STEPWISE code were generated using the *PA_AnalysisRemote.accdb* Microsoft Access database that has links to the official PA results (PA_Results) and parameter (ParamDB) databases located on the BTO machine. A copy of the *PA_AnalysisRemote.accdb* database is included in /nfs/data/CVSLIB/WIPP_ANALYSES/CRA19/STEPWISE/database. Input files were generated using this database on a machine using Microsoft Windows 10 by selecting the menu button in the MainForm form entitled "Create Stepwise Input Files (3)," selecting the CRA19 analysis, and then selecting the "Create Stepwise Files" button. Input files and run scripts were generated and the user then chose a destination folder for the input files.

Input files and run scripts were then transferred to the Solaris cluster. The input files and run scripts as-created on a Windows machine are not readable by the STEPWISE code on Solaris. After transfer to the Solaris cluster, a utility program dos2unix was used to remove end-of-line characters from each input file. Table 56 lists the input and run script filenames.

Table 56 – The STEPWISE input files used were:

File	Repository	Comment
RunSTEPWISE.sh	/home/run_mast/GD	Run script
Input/STP_CRA19_Raw_ALL_R*i*.sh	$REP1/CRA19/STEPWISE	Run script
Input/STP_CRA19_Rank_ALL_R*i*.sh	$REP1/CRA19/STEPWISE	Run script
Input/STP_CRA19_Raw_ALL_R*i*.inp	$REP1/CRA19/STEPWISE	Input file
Input/STP_CRA19_Rank_ALL_R*i*.inp	$REP1/CRA19/STEPWISE	Input file
Input/STP_CRA19_LHS_R*i*	$REP1/CRA19/STEPWISE	Input file
Input/STP_CRA19_MEANS_R*i*	$REP1/CRA19/STEPWISE	Input file

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

Table 57 – The STEPWISE CVS repositories used were:

Repository
$CODE/STEPWISE
$REP1/CRA19/STEPWISE

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

Table 58 – The STEPWISE output files produced were:

File	Repository	Comment
Output/STP_CRA19_Rank_ALL_R*i*.txt	$REP1/CRA19/STEPWISE	Output file
Output/STP_CRA19_Rank_ALL_R*i*.sigma	$REP1/CRA19/STEPWISE	Output file

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

Table 59 – The STEPWISE executable file used was:

File	Repository	Comment
Build/Solaris/stepwise (Ver:2.22)	$CODE/STEPWISE	Executable file

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

6.0 REFERENCES

Clayton, D.J., S. Dunagan, J.W. Garner, A.E. Ismail, T.B. Kirchner, G.R. Kirkes, M.B. Nemer. 2008. Summary Report of the 2009 Compliance Recertification Application Performance Assessment. Sandia National Laboratories, Carlsbad, NM. ERMS 548862.

Clayton, D.J., R.C. Camphouse, J.W. Garner, A.E. Ismail, T.B. Kirchner, K.L. Kuhlman, M.B. Nemer. 2010. Summary Report of the CRA-2009 Performance Assessment Baseline Calculation. Sandia National Laboratories, Carlsbad, NM. ERMS 553039.

Cotsworth, E. 2005. EPA Letter on Conducting the Performance Assessment Baseline Change (PABC) Verification Test. U.S. EPA, Office of Radiation and Indoor Air, Washington, D.C. ERMS 538858.

Cotsworth, E. 2009. EPA Letter on CRA-2009 First Set of Completeness Comments. U.S. EPA, Office of Radiation and Indoor Air, Washington, D.C. ERMS 551444.

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

Kirchner, T., A. Gilkey, and J. Long. 2014. Summary Report on the Migration of the WIPP PA Codes from VMS to Solaris, AP-162 Revision 1. Sandia National Laboratories, Carlsbad, NM. ERMS 561757.

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

Leigh, C.D., J.F. Kanney, L.H. Brush, J.W. Garner, G.R. Kirkes, T. Lowry, M.B. Nemer, J.S. Stein, E.D. Vugrin, S. Wagner, and T.B. Kirchner. 2005. 2004 Compliance Recertification Application Performance Assessment Baseline Calculation, Revision 0. Sandia National Laboratories, Carlsbad, NM. ERMS 541521.

Long, J.J. 2017a. Nuclear Waste Management Procedure NP 9-2, Parameters, Revision 2. Sandia National Laboratories, Carlsbad, NM. ERMS 568008.

Long, J.J. 2017b. Nuclear Waste Management Procedure NP 19-1, Software Requirements, Revision 18. Sandia National Laboratories, Carlsbad, NM. ERMS 567724.

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

MacKinnon, R.J., and G. Freeze. 1997a. Summary of EPA-Mandated Performance Assessment Verification Test (Replicate 1) and Comparison With the Compliance Certification Application Calculations, Revision 1. Sandia National Laboratories, Carlsbad, NM. ERMS 422595.

MacKinnon, R.J., and G. Freeze. 1997b. Summary of Uncertainty and Sensitivity Analysis Results for the EPA-Mandated Performance Assessment Verification Test, Rev. 1. Sandia National Laboratories, Carlsbad, NM. ERMS 420669.

MacKinnon, R.J., and G. Freeze. 1997c. Supplemental Summary of EPA-Mandated Performance Assessment Verification Test (All Replicates) and Comparison With the Compliance

Certification Application Calculations, Revision 1. Sandia National Laboratories, Carlsbad, NM. ERMS 414880.

U.S. Congress. 1992. WIPP Land Withdrawal Act, Public Law 102-579, 106 Stat. 4777, 1992; as amended by Public Law 104-201, 110 Stat. 2422, 1996.

U.S. Department of Energy (DOE) 1996. Title 40 CFR Part 191 Compliance Certification Application for the Waste Isolation Pilot. 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 (EPA). 1998. 40 CFR 194, Criteria for the Certification and Recertification of the Waste Isolation Pilot Plant's Compliance with the Disposal Regulations: Certification Decision: Final Rule, Federal Register. Vol. 63, 27354-27406.

U.S. Environmental Protection Agency (EPA). 2006. 40 CFR 194, Criteria for the Certification and Recertification of the Waste Isolation Pilot Plant's Compliance with the Disposal Regulations: Certification Decision: Final Rule, Federal Register. Vol. 71, 18010-18021.

U.S. Environmental Protection Agency (EPA). 2010. 40 CFR Part 194 Criteria for the Certification and Recertification of the Waste Isolation Pilot Plant's Compliance With the Disposal Regulations: Recertification Decision, Federal Register No. 222, Vol. 75, pp. 70584-70595, November 18, 2010.

U.S. Environmental Protection Agency (EPA). 2017. Criteria for the Certification and Recertification of the Waste Isolation Pilot Plant's Compliance with the Disposal Regulations; Recertification Decision. July 19, 2017. Office of Radiation and Indoor Air, Docket EPA-HQ-OAR-2014-0609-0079.

Zeitler, T.R. 2019. Analysis Plan for the 2019 WIPP Compliance Recertification Application Performance Assessment. Sandia National Laboratories, Carlsbad, NM. ERMS 571150.