561757

g 4/3/14

SANDIA NATIONAL LABORATORIES
WASTE ISOLATION PILOT PLANT

Summary Report on the Migration of the WIPP PA Codes From VMS to Solaris, AP-162 Revision 1

Authored by:	Thomas Kirchner Print Name	Thomas Kirchner Signature	4/2/2014 Date
Authored by:	Amy Gilkey Print Name	Amy Gilkey Signature	4/2/2014 Date
Authored by:	Jennifer Long Print Name	Jennifer Long Signature	4/2/2014 Date
Reviewed by:	Todd Zeitler Print Name Technical Reviewer	Todd Zeitler Signature	4/2/2014 Date
Reviewed by:	Shelly Nielsen Print Name Quality Assurance Reviewer	Shelly R. Nielsen Signature	4-2-14 Date
Approved by:	Sean Dunagan Print Name Department Manager	Sean Dunagan Signature	4/2/2014 Date

WIPP:4.4.1.3.1:PA:QA-L:561456

Table of Contents

Table of Figures

Acronyms

ASCII	American Standard Code for Information Interchange, defines "text" files
CDB	CAMDAT Database
CCDF	Complementary cumulative distribution function
CVS	Code Versioning System, an application used to track and tag versions of files
DD	Design document
EPA Units	Curies of releases normalized to the repository inventory
ID	Implementation document
OpenVMS	Open Virtual Memory System, the operating system used on the HP Alpha computers
PA	Performance assessment
RD	Requirements document
SPR	Software problem report
UM	User's manual
VD	Validation document
VVP	Validation and verification plan
WIPP	Waste Isolation Pilot Plant

Introduction

This report summarizes the migration of the WIPP performance assessment (PA) computational suite from an OpenVMS (Open Virtual Memory System) Alpha Cluster to a Sun Solaris Blade Server running with Intel processors. This migration was contemplated for many years but not implemented because of the manpower costs of requalifying the codes and concerns that the differences in floating point formats between computer architectures would require extensive investigations to explain differences in PA results. However, the diminishing technical and hardware support for OpenVMS and the Alpha servers coupled with rising costs for such support provided additional motivation to change computing platforms. Additionally, augmentation of the current capabilities is restricted due to the availability of additional hardware and the cost. Because of these reasons, SNL WIPP decided to move towards Sun Solaris/Intel based platforms. The primary objective of this effort was to port and qualify those codes needed to perform PA on the Sun Solaris cluster. In addition, the run control system is being simplified using improved scripting methods. The port of the LHS and CCFDGF codes to Solaris also included modifications to have their output data exported directly to database tables for easier access by the analysts.

The current strategy for maintaining files used or produced in PA is to keep versions sufficient to perform the last two WIPP compliance recertification analyses. Therefore, versions of codes and associated input files that were used for the PABC-2009 analyses were ported followed by those used in the CRA-2014 analyses. In addition, CDCFGF 7.00 was ported because it will likely to be used in upcoming assessments involving the reconfiguration of the layout of panels in the repository.

This revision of the report corrects two errors made in running the PABC09 and CRA-2014 integration tests. These errors are inconsequential in terms of the conclusions because they both resulted in differences in the releases from CCDFGF in the fourth significant digit. The PABC-2009 analysis was run twice, the first time using DRSPALL tables from the CRA-2004 PA as was done on VMS and the second time using DRSPALL tables created on Solaris. For the second run CUTTINGS_S, BRAGFLO_DBR and CCDFGF should have been rerun because DRSPALL output is used by CUTTINGS_S and CUTTINGS_S output is used by BRAGFLO_DBR and CCDFGF. However, BRAGFLO_DBR was not rerun resulting in differences in the fourth significant digit of DBR release values for two vectors.

BRAGFLO calculations use a numerical method for solving the partial differential equations that depends on achieving convergence in the solution. Convergence is defined using tolerance criteria specified in the input file to the code. If BRAGFLO fails to converge using the standard criteria it is rerun using an alternative set of convergence criteria. If the rerun fails to converge then BRAGFLO is run once more using a third set of convergence criteria. The client scripts written by the BRAGFLOlib.py script used to run BRAGFLO were intended to automatically do the reruns using the alternative tolerance criteria should BRAGFLO fail to converge. The error in the one CRA14BL run (vector 81, scenario 2) and the dependent CRA14TP and CRA14BV runs occurred because the BRAGFLO code failed immediately for some unknown reason. Failures of jobs running on the cluster have been seen in the past and seem to be related to the system reporting "insufficient resources", although the limited resource has not yet been identified. The script detected the failure and treated it as a failure to converge. This type of error is specific to

BRAGFLO and SECOTP2D because their slave scripts, written to run on the cluster nodes, automatically do reruns with alternative input files when the solutions of the equations fail to converge whereas failure of a program in the slave scripts for the other codes is detected as a failure of the script. The scripts for BRAGFLO and SECOTP2D have been modified to treat as failure to converge only those cases where the code returns an error code of 1 which is the error code returned due to a failure to converge.

The differences arising from these two errors have no impact on the conclusions and the graphs of results are indistinguishable. However, these analyses were rerun using improved scripts in order to provide a clean, reproducible set of baselines. The original repositories were moved from /nfs/data/CVSLIB/WIPP_Analyses to /nfs/data/CVSLIB/WIPP_ARCHIVE. The results from the original analysis are stored in the MySQL database JanPAResults.

Another difference between Revision 1 and Revision 0 of this report is that the comparisons made between the results from VMS and those from Solaris now use the results that include the output from DRSPALL that was run on Solaris, whereas previously the comparison was made using the Solaris results which included the DRSPALL output from the CRA-2004 analysis run on VMS. On VMS the DRSPALL results from the CRA of 2004 were used in the PABC-2009 and CRA14 analyses because there were no changes in parameters that required rerunning DRSPALL. Thus the most equivalent runs on Solaris also used the CRA-2004 DRSPALL outputs. However, we cannot exactly reproduce the DRSPALL results from VMS on Solaris and we wanted to use as our new baseline results that could be reproduced without having to use the VMS/Alpha system to reproduce the DRSPALL outputs. This change in Revision 1 of this report was made to provide detailed comparisons between the VMS and Solaris outputs for what we intend to use as the Solaris baseline calculations for PABC09 and CRA14 from this point forward. As will be seen below (Figs. 23-25) the differences between the Solaris results based on using the DRSPALL outputs generated on VMS and the Solaris results bases on using the DRSPALL outputs generated on Solaris are negligible.

Procedures

The port of the codes followed the procedure defined in NP 19-1 (Long 2012) and AP-162 (Kirchner 2012a). There were five tasks for the migration of PA codes to the new UNIX platform.

Task 1: Migration of PA modules to the new UNIX platform. A Change Control form (NP 19-1-9) was completed for each code. The codes were modified as needed to run under the new operating system. Outstanding software problem reports were identified and resolved during the migration. In addition, the variables in the codes were converted to double precision, FORTRAN standard dynamic arrays replaced many of the system-dependent memory allocation methods used with the FORTRAN-77 codes on VMS, and storage of output was enhanced by inserting some results (from LHS and CCDFGF) into database tables. The codes were compiled by an authorized "run master", Amy Gilkey, and the executable files ("builds") stored with the source code and run control files in the Code Versioning System (CVS) repositories. The products of this task were a Change Control form and an executable file, stored in CVS, for each code that was converted.

Task 2: Results from prior validation tests on VMS were converted to ASCII (text) files to allow for comparisons across the platforms and to transfer those results to the Solaris platform. The VMS results were stored in the CVS repositories with their associated code along with the results from the Solaris tests. The products of this task were the test results from the most current VMS validation test installed in CVS. The VMS results were not stored if they were not used in validation testing (see Task 4).

Task 3: Scripts were created for running the tests. The scripts for managing the use of the CVS repositories were written using Python and UNIX shell script "wrappers" were created for executing the Python scripts. The scripts were stored in the CVS repository for the associated code.

Task 4: Run tests for validation. VMS uses a different floating point format than do the Intel processors. In addition, many of the codes were compiled on VMS using single precision variables whereas on Solaris they were compiled using double precision. Similar results were expected for tests of the utility codes, and the pre- and post-processing codes, but those codes which require numerically intensive calculations involving convergence on solutions of differential equations were expected to show differences as these kinds of problems can be sensitive to the design of the hardware and the floating point representation. Therefore, allowances for differences were made when comparing numerical results using regression tests. Where regression testing was difficult to implement, validation against the criteria of the VVP was used. Results of validation tests were stored in the CVS repository of the code and documented in the code's Validation Document (VD).

Task 5: Installation and Checkout forms were completed for each code that was validated.

Storage in CVS

The builds, source code and files related to validation testing, including the scripts, for each code were stored in a CVS repository named for that code. The convention used was to name the repository in upper case, e.g. BRAGFLO, and the executable file in lower case. The repositories were stored in a file system that is shared with the various Solaris and Linux computers used for PA calculations. The path to the directory containing the CVS repositories is on the current set of computers used for PA is /nfs/data/CVSLIB/WIPP_CODES/PA_CODES. The environmental variable $CVSLIB stores the path down to CVSLIB on each machine as a contingency for mounting the file system differently in the future. $CVSLIB is set on each machine to point to the "root" of the CVS repository files no matter what the mount point is. All code repositories have the same structure. The executable file is stored in the module (upper level directory) called Build under the subdirectory Solaris. The scripts associated with compiling the code are in RunControl/Solaris. The source code is in the module Source. The Test module has the directories Auxiliary, Input, Output and RunControl to store auxiliary files, input files, output files and the scripts, respectively, for the validation tests.

Deviations from Procedures

The suite of PA codes includes a number of utilities, preprocessor codes and post processor codes designed to facilitate the transfer of data between the major codes such as LHS and BRAGFLO. During the migration to the extent possible such codes were compiled and validated prior to their use in testing other codes. However, sometimes codes are co-dependent, as when the output from one code is needed to verify a second code, and the second code is needed to verify the first. For example, GROPECDB, which extracts data from the binary CDB files, was used to validate ALGEBRACDB, GENMESH, ICSET, RELATE, and PANEL before its qualification was complete. The official build of GROPECDB was completed and that executable used in the test of other codes, then subsequently the validation of GROPECDB using that same executable was completed.

Two integration tests were conducted in addition to the validation of the individual codes. The runs for the PABC09 analysis and for the four CRA14 cases were executed on Solaris. The CCDFs for the major releases (direct brine, cutting and cavings, spallings, Culebra and total releases) were compared to those generated on VMS.

The output of LHS and CCDFGF are used extensively when performing analyses on the results of PA calculations. Previously these data had to be extracted from output files of the codes. These data are now additionally inserted into database tables in the MySQL PA_Results database. The PA_Results database has no inherent functionality in terms of queries or stored procedures; it is simply used to store output data from the analyses. Validation tests were performed using the results of the PABC-2009 integration test by comparing the values inserted into the database against the values written to the output files of LHS and CCDFGF. These tests are documented in the Validation Documents (VDs) for the codes (see Table 1 for ERMS numbers). The use and validation of PA_Results was not identified in AP-162 and therefore this task is a deviation from that plan.

Elimination of Utility Codes

Two utility codes were eliminated during the migration by combining their functionality with another code. LHS_EDIT was incorporated into LHS and some functions of VTRAN2 were incorporated into PRESECOTP2D. LHS_EDIT was used to post-process LHS files to enforce certain kinds of conditional relationship between variables that was not supported in LHS. VTRAN2 was used previously when binary files generated on Linux had to be converted to ASCII files then converted back to binary files on VMS. VTRAN2 also transformed the MODFLOW volume fluxes to SECOTP2D Darcy velocity for the appropriate transport sub-domain. PRESECOTP2D 1.23 reads the MODFLOW output file directly and handles the transformation of the MODFLOW volume fluxes to SECOTP2D Darcy velocity for the transport sub-domain. New parameters in the PRESECOTP2D input control file, previously input to the VTRAN2 utility, define the sub-domain and the cell area for the transformation.

Utility codes MERGESPALL and SCREEN_NUTS were migrated to Solaris. For these utility codes, an official build was performed and stored in CVS. These utilities have previously been qualified under NP 9-1 (Safely 2012) as routine calculations rather than under NP 19-1 (Long 2012), and hence would normally be qualified when next used. However, regression tests were

performed against the prior VMS test results and stored in CVS. Documentation of the tests was provided to the code sponsors, thus providing the documentation for NP 9-1 qualification. That documentation is reproduced in Appendix 1 for MERGESPALL and Appendix 2 for SCREEN_NUTS to support the use of these codes in the integration tests.

Documents Revised

Changes to the codes during the port included resolving Software Problem Reports (SPRs), changes to command line arguments and modification of VMS-specific code. New functionality was added to LHS and CCDFGF, enabling them to write data directly to a database. These changes required the update of some documents. Table 1 lists the various documents that were revised in accordance with NP19-1.

Table 1.Documents revised during the migration. The revision number of the document or "Add" (addendum) is shown with the ERMS numbers.

Code Name & Version	Executable Date	Document[1]						Regression Report
		RD	VVP	DD	ID	UM	VD	
ALGEBRACDB 2.36	9/11/12				2.36 (557794)	Add (557795)		2.36 (557796)
BLOTCDB 1.38	7/17/13	1.38 (560360)	1.38 (560361)		1.38 (560362)	Add (560363)	1.38 (560361)	
BRAGFLO 6.03	2/1/13		Add (558351)		6.03 (558350)	Add (558351)	6.03 (558352)	
CCDFGF 5.03	9/21/12	Add (558223)	Add (558223)		5.03 (558224)	Add (558223)	Add (558223)	
CCDFGF 6.01	10/17/13	Add (560537)	Add (560537)		6.01 (560538)	Add (560537)	Add (560537)	
CCDFGF 7.01	11/12/13	Add (561130)	Add (561130)		7.01 (561131)	Add (561130)	Add (561130)	
CUTTINGS_S 6.03	1/15/13				6.03 (558838)	Add (558839)		6.03 (558840)
DRSPALL	5/28/13			1.21	1.21	Add	1.21	

Code Name & Version	Executable Date	Document[1]						Regression Report
		RD	VVP	DD	ID	UM	VD	
1.21				(560045)	(560047)	(560048)	(560046)	
EPAUNI 1.16	3/25/13	Add (559075)	Add (559075)		1.16 (559074)	Add (559075)	Add (559075)	
FMT 2.41	9/27/13				2.41 (560967)	Add (560968)		2.41 (560969)
GENMESH 6.09	9/11/12				6.09 (557798)	Add (557799)		6.09 (557800)
GROPECDB 2.13	8/22/12	2.13 (558221)	2.13 (557790)		2.13 (557789)	Add (557792)	2.13 (557790)	
ICSET 2.23	9/11/12				2.23 (557802)	Add (557803)		2.23 (557805)
LHS 2.43	3/11/13	2.43 (559265)	2.43 (559266)		2.43 (559268)	Add (559267)	2.43 (559266)	
MATSET 9.21	11/6/12	Add (557809)	Add (557809)		9.21 (557807)	Add (557808)	Add (557809)	
NONLIN 2.02	11/4/13				2.02 (561134)	Add (561135)		2.02 (561136)
NUTS 2.06	03/27/13				2.06 (559594)	Add (559595)		2.06 (559596)
PANEL 4.04	9/26/12		Add (557787)		4.04 (557732)	Add (557787)	4.04 (557733)	
POSTBRAG 4.02	1/10/13				4.02 (557828)	Add (557829)		4.02 (557831)
POSTLHS 4.08	3/5/13	4.08 (559271)	4.08 (559272)	4.08 (559271)	4.08 (559274)	4.08 (559273)	4.08 (559272)	

Code Name & Version	Executable Date	Document[1]						Regression Report
		RD	VVP	DD	ID	UM	VD	
POSTSECOTP2D 1.05	4/18/13				1.05 (559787)	Add (559788)		1.05 (559789)
PREBRAG 8.03	1/23/13				8.03 (557816)	Add (557817)		8.03 (557818)
PRECCDFGF 2.01	9/9/13				2.01 (560550)	Add (560551)		2.01 (560552)
PRECCDFGF 1.06	10/16/12				1.06 (557820)	Add (557821)		1.06 (557822)
PRELHS 2.41	3/6/13	2.41 (559259)	2.41 (559260)		2.41 (559262)	Add (559261)	2.41 (559260)	
PRESECOTP2D 1.23	4/18/13	Add (559778)	Add (559778)		1.23 (559777)	Add (559778)	1.23 (559779)	
RELATE 1.45	9/11/12				1.45 (557824)	Add (557825)		1.45 (557826)
SECOTP2D 1.43	8/2/13				1.43 (559782)	Add (559783)	1.43 (559784)	
STEPWISE 2.22	7/2/13				2.22 (560367)	Add (560368)	2.22 (560366)	
SUMMARIZE 3.02	10/31/12				3.02 (557812)	Add (557813)		3.02 (557814)

[1] RD = Requirements Document, VVP = Validation and Verification Plan, DD = Design Document, ID = Implementation Document, UM = User's Manual and VD = Validation Document.

Software Problem Reports Resolved

There were several SPRs outstanding at the time of migration (Table 2). These SPRs were resolved during the migration.

Table 2. Software Problem Reports addressed during the migration.

Code	SPR Number	SPR Description	Status
BLOTCDB 1.38	02-004	Inability to plot logical grid legibly (font issue).	This problem was fixed in version 1.38 but still remains for version 1.37.
CCDFGF 5.03 and 6.01	10-001	The VVP set up tests for the observed frequencies of various events that computed a confidence interval (CI) on the expected value. However, the expected value is a constant, hence it has no uncertainty, whereas the observed frequencies are subject to random error and confidence intervals can be computed for them. This was an issue about the use of the term "confidence interval", not the test itself.	This problem was fixed in the documentation.
CCDFGF 5.03, 6.01 and 7.01	07-003	If no RH waste is encountered, as when the RH waste area parameter is set to 0, then a divide-by-zero error is encountered in the subroutine PrintStatistics.	This problem was fixed in the code for 6.01 and 7.01 but not for 5.03, and was changed in the documentation for all of them.
CCDFGF 5.03, 6.01 and 7.01	06-005	A fatal divide-by-zero error was encountered during a test involving changing the probability of encountering remote handled waste.	This problem was fixed in the code for 6.01 and 7.01 but not for 5.03, and was changed in the documentation for all of them.
DRSPALL 1.21	04-001	The Design Document only describes a simplification of the equations used and not the actual equations.	The documentation now describes the correct equations.
EPAUNI 1.16	07-002	EPAUNI failed to properly read an input file that consisted of more than one waste stream when the IFLAGI control parameter was set to 3. EPAUNI reads in the value of the waste unit factor (WUF) rather than getting it from the parameter database via MATSET.	This has been fixed with version 1.16 but remains for version 1.15A.

SECOTP2D 1.43	97-017	Error in the treatment of boundary conditions.	This was fixed in the documentation.
SUMMARIZE 3.02	06-001	If an attribute is requested in the input control file, SUMMARIZE could incorrectly determine that the attribute does not exist for the element block and abort.	This has been fixed with version 3.02 but remains for version 3.01

Integration Tests

Although the validation tests confirm that the performances of the individual codes are adequate, confidence in the new PA system requires a comparison of results from one or more PAs run on VMS and Solaris. Therefore, two integration tests were conducted. These integration tests are deviations from AP-162 in that the AP only required that the individual validation tests be performed. The integration tests consisted of running the PABC09 and CRA14 analyses on Solaris and then comparing the releases projected by CCDFGF from the VMS and Solaris runs. Both mean releases and releases by vector were compared. DRSPALL was last run for the CRA-2004 PABC and the tables it produced have been used since then. One deviation from the PABC09 analysis is that DRSPALL was run on Solaris and both the Solaris-generated tables and the CRA-2004 PABC tables from VMS were used, thus generating two sets of CCDFs for spallings and direct brine releases. The analysis name for the results from using the CRA1BC DRSPALL tables was changed from PABC09 to AP162_T1". Another difference between the VMS and Solaris analyses is that the Solaris script for DRSPALL implements automated checks for cases requiring longer runs and cases where a switch to cylinder mode is required. For the CRA-2004 PABC these checks and re-runs were done by the analysts. In the CRA-2004 PABC vector 59 at pressure level 4 was rerun for an extended time. In the integration test this case was not rerun because the criteria for achieving an asymptotic cavity size (based on the magnitude of the difference between 500 and 600 simulated seconds) could not be set to include this case without including several others.

Some changes were made in the running of the CRA14 cases related to sharing files between cases. BRAGFLO was run for all three replicates of the CRA14BL case to support running all three replicates of PANEL in the CRA14BV case. In addition, LHS was run for three replicates of the CRA14TP case, also to support the running of all three replicates of PANEL in the CRA14BV case. The three replicates of PANEL outputs were used in the CRA14-0 case rather than running PANEL there, as was done on VMS. These changes made the naming conventions more consistent (the PANEL files from the CRA14-0 case on VMS were all labeled using CRA14BV in their names), and also avoided an extra run of PANEL in the CRA14-0 case. The PANEL_INT step was not run in the CRA14BV case on VMS, so the PRECCDFGF input files for case CRA14BV included sum_panel_int_CRA14BL_r1_s6_T*tttt*.tbl files in the VMS analysis. On Solaris the PANEL_INT step was run in the BV case and the PANEL files used were named sum_panel_int_CRA14BV_b1_r1_s6_T*tttt*.tbl. The CRA14BL and CRA14BV_b1 versions of these files are interchangeable because both represent the same brine volume.

BRAGFLO 6.00 was used for the PABC09 and the CRA14BL analyses but only BRAGFLO 6.02 was ported, since the changes from version 6.00 to 6.02 were minor and 6.02 was backward compatible. However, using 6.02 did require some modification of the MATSET input files because there were two parameters that need to be specified for version 6.02 that were not required in 6.00. These parameters were REFCON:YRSEC and REFCON:SECYR. In addition the MATSET input file was modified by adding and setting the properties REFCON:DN_HYDRO and REFCON:MW_HYDRO. Updated versions of MATSET and PRELHS were also used in the PABC09 analysis because the parameter database, the PAPDB,

was migrated from Microsoft SQLServer to MySQL prior to the running of the CRA14 analysis (Kirchner 2012b).

Most of the input files used in these tests were identical to those used on VMS although the format of the names was changed. File names on Solaris are case sensitive and names were changed to lower case except for the analysis name component and some key characters, such as the location key (L, M, and U) in BRAGFLO_DBR input files. The names of the ALGEBRACDB input files for PANEL were modified by adding "_b*i*" following the analysis name component where *i* was an index of brine volume, even in case CRA14BL where only one brine volume was simulated.

The ALGEBRACDB input files for step 2 were modified for the CRA14BV runs. A change to the PANEL code made during the migration made it necessary to apply multipliers of 1,2,3,4 and 5 to the minimum brine volume (DBRMINBV) using ALGEBRACDB in order to input the five brine volumes into PANEL. This change was made in the alg2_panel_CRA14BL.inp and alg2_panel_CRA14BV.inp files using the statements:

```
! COMPUTE SCALEING FOR ONE PANEL AND PANEL MIN BRINE VOLUME LIMIT BLOCK 2
INVSCALE=MAKEPROP(VPANLEX/VREPOS)
PANDFVOL=MAKEPROP(DBRMINBV[B:1]*INVSCALE)
```

A change to SUMMARIZE eliminated a great many input files. Previously, the template for input files and output file name were contained in the SUMMARIZE input (user) file, so a file had to be created for each combination of replicates, scenarios, etc. These files were identical except for the file name and template. The Solaris version of SUMMARIZE takes the template and output file name from the command line, thus eliminating many files.

Previously the MODFLOW output files, which are in a binary format, had to be converted to ASCII, transferred to VMS, and used there by SECOTP2D. The Solaris version of SECOTP2D was modified to read the MODFLOW files directly.

Not all of the codes ported to Solaris were run during the integration tests. For example, NONLIN and FMT are used to support the calculation of actinide solubility but they are not used directly when running a PA. STEPWISE is used to identify the sensitivity of the outputs to sampled parameters using the data created during a PA. STEPWISE might be replaced with an off-the-shelf statistical analysis package in the future.

Logging Usage of Parameters

Normally extracting parameters from the Performance Assessment Parameter Database (PAPDB) by MATSET and PRELHS causes the parameter name, version and analysis information to be logged into the AnalysisRetrievalHistory table of the database. In Revision 0 of this report it was stated that the analysis names for the CRA14 analysis run on VMS were changed to allow the Solaris runs to log their retrieval information. However, that was an error. The analysis names were not changed and the parameter retrievals for the CRA14 runs on Solaris were not logged into the table. However, the versions of parameters pulled by a code for an analysis are now controlled using the AnalysisToVersion table, which ensured that the versions of the parameters used for the VMS runs are identical to those used in the Solaris runs. This was

not the case for the PABC-2009 analysis on VMS so those data in AnalysisRetrievalHistory were assigned the analysis name PABC09_VMS and the Solaris retrievals were assigned the analysis name PABC09. One side effect of this change is that the auto-generated LastModified data field we updated to the date on which the change in analysis name was made. Although having two sets of runs being preserved for a single analysis is expected to be uniquely associated with the migration from VMS to Solaris the best method for handling such cases is currently being reviewed with the goal of avoiding problems in logging parameter usage in the future.

Comparison of PABC09 Results

Overall the mean total, DBR, spallings, cuttings and cavings and Culebra releases were nearly indistinguishable between the PABC09 and CRA14 results (Figs. 1-3). These results (for both PABC09 and CRA14) were generated using the DRSPALL tables that were created on Solaris rather than the tables from the CRA-2004 that was run on VMS. There was a small difference in the replicate 1 releases from the Culebra at low probabilities that was due to a small deviation in a single vector (Fig. 4).

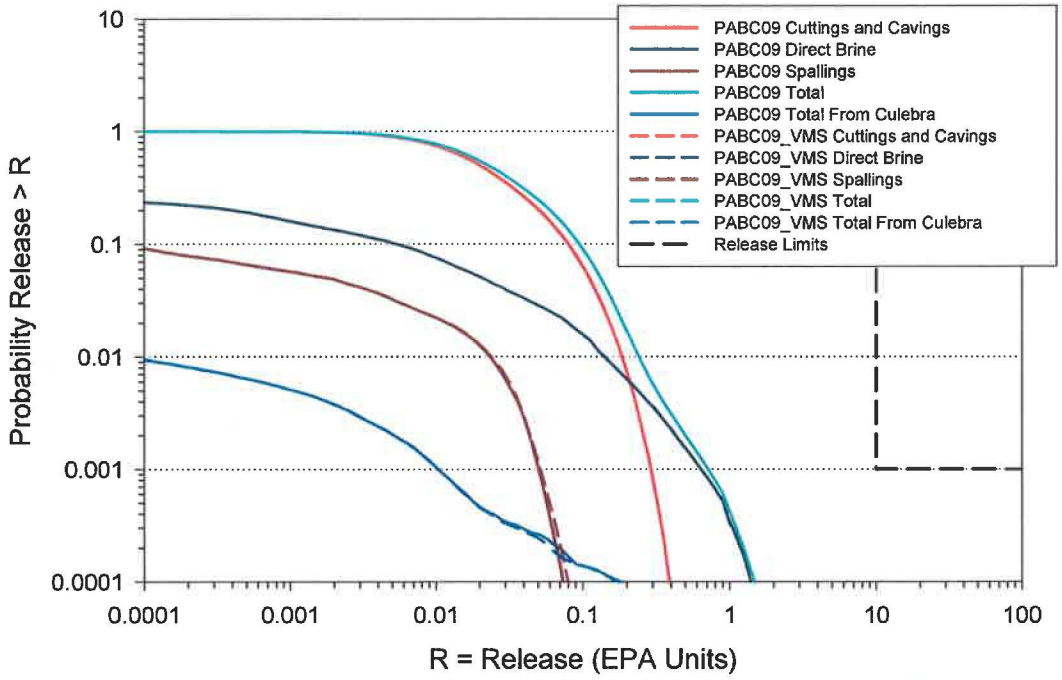

Figure 1. Comparison of mean releases replicate 1. Solaris results are displayed using solid lines and VMS results using dashed lines.

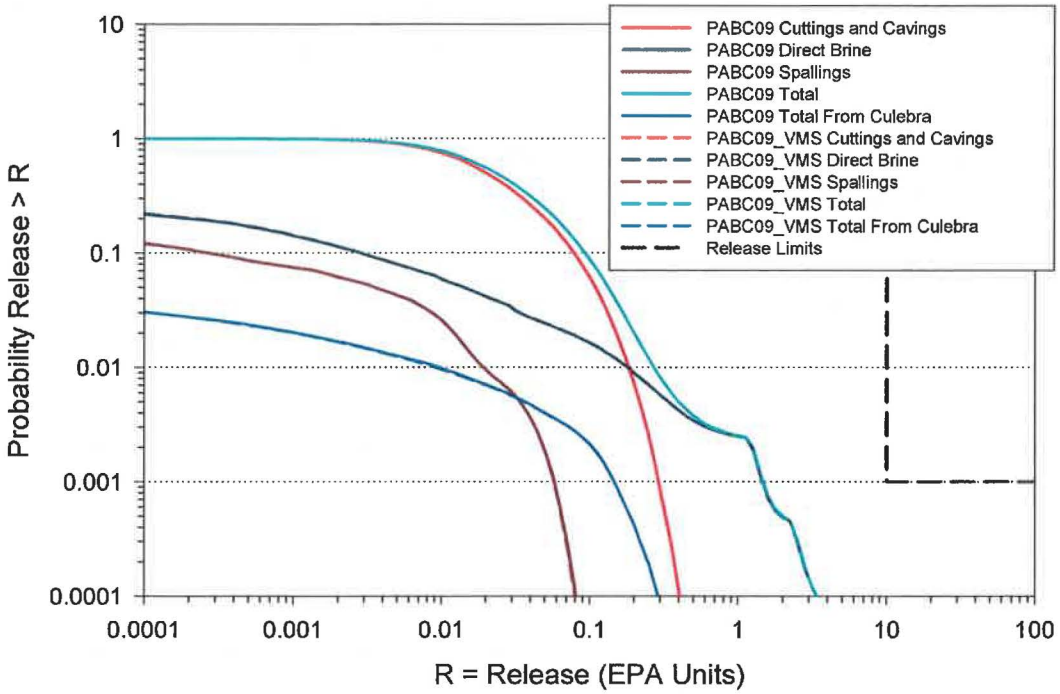

Figure 2. Comparison of mean releases replicate 2. Solaris results are displayed using solid lines and VMS results using dashed lines.

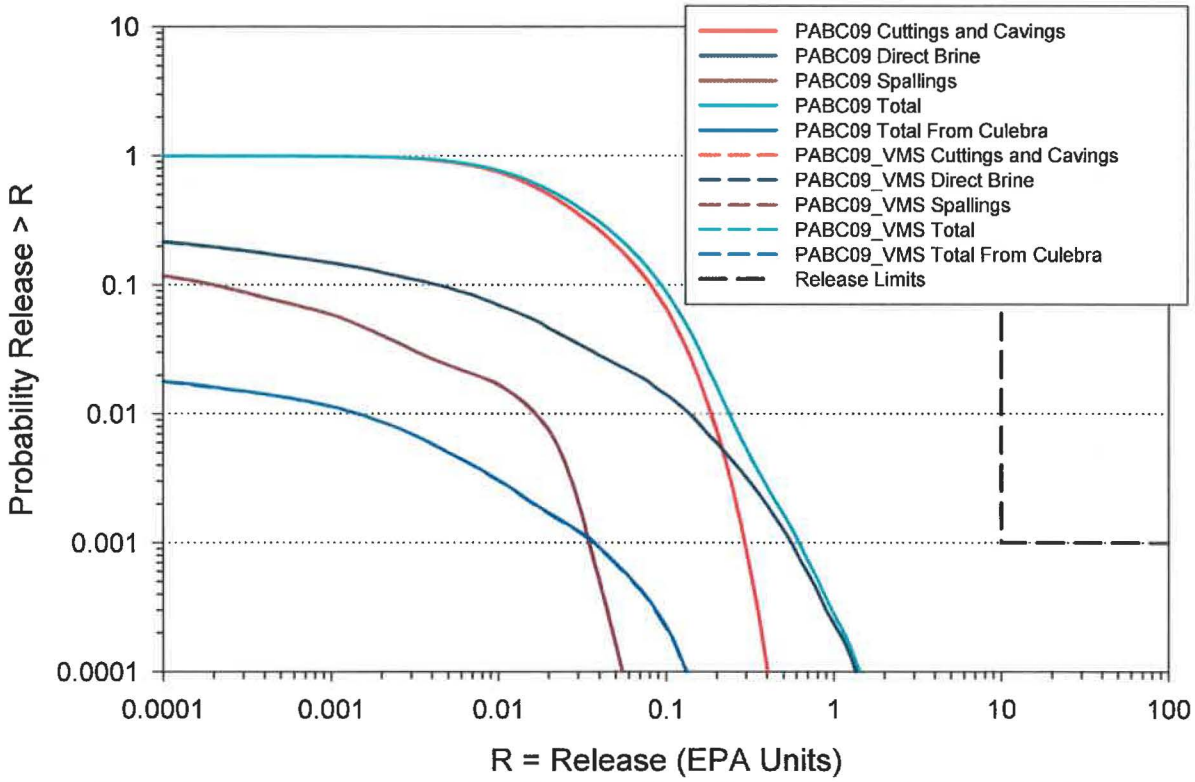

Figure 3. Comparison of mean releases replicate 3. Solaris results are displayed using solid lines and VMS results using dashed lines.

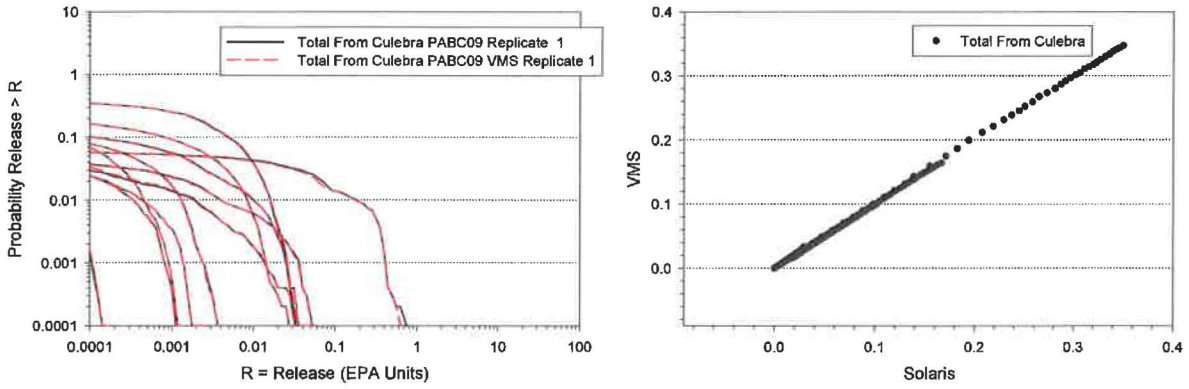

Figure 4. Comparison of CCDFs by vector for the releases from the Culebra as computed on VMS and Solaris for replicate 1. On the right are the probabilities of releases from the Culebra paired by release levels.

In order to facilitate the comparison of the Solaris results and the VMS results for the PABC09 analyses the vector data were paired by release values and the probabilities plotted. If the Solaris and VMS probabilities of release were identical then the points would lie on the diagonal of the plot. For example, Fig. 4 shows the Culebra releases by vector and a plot comparing the probabilities of releases paired by release levels across all vectors. Data compared in this way were restricted to values greater than or equal to 0.0001 in order to eliminate the region where numerical noise often dominates the release estimates. Similar plots for DBR releases, cuttings and cavings releases, spallings releases and total releases are shown in Figs. 5-8. In addition, a comparison was made of the releases to the Culebra to supplement the releases from the Culebra because of the low number of vectors having releases from the Culebra greater than 0.0001 EPA units (Fig. 9).

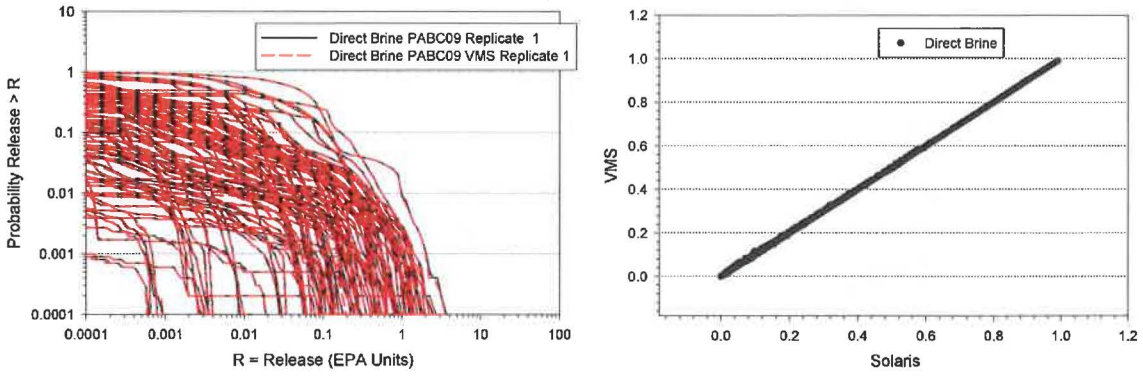

Figure 5. Comparison of VMS and Solaris probabilities of direct brine releases paired by release level. On the right are the probabilities of the releases paired by release levels.

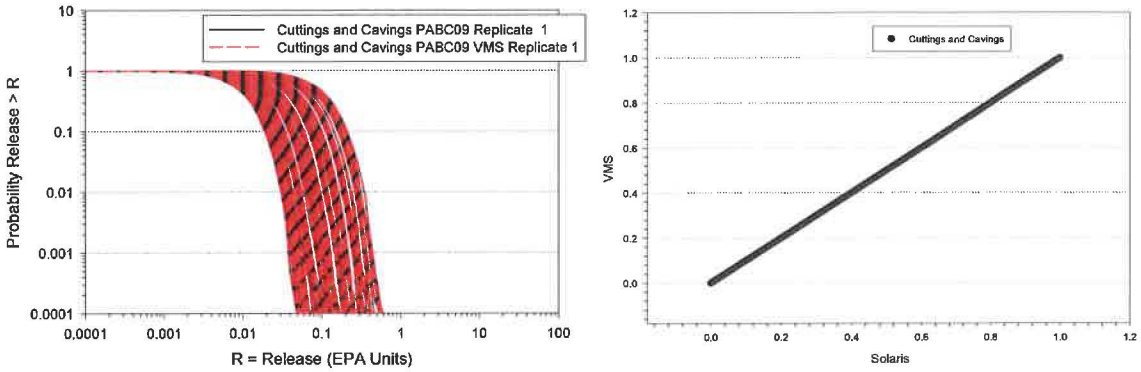

Figure 6. Comparison of CCDFs by vector for cuttings and cavings releases as computed on VMS and Solaris for replicate 1. On the right are the probabilities of the releases paired by release levels.

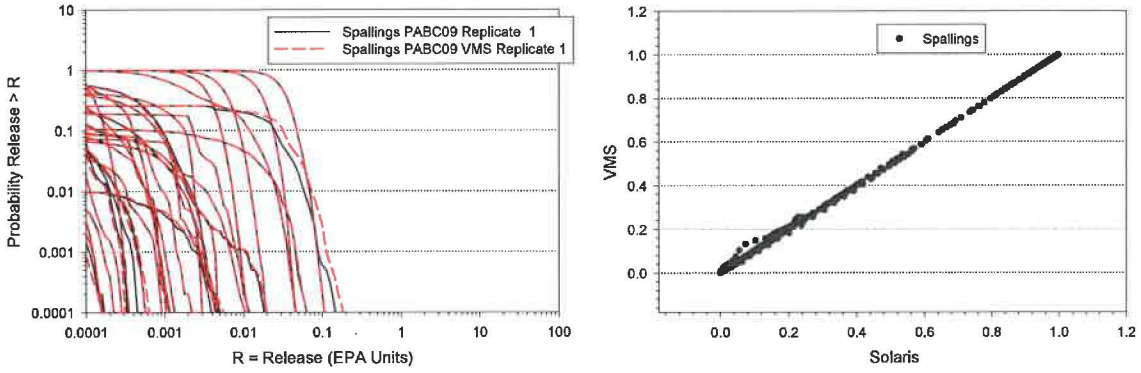

Figure 7. Comparison of CCDFs by vector for spallings releases as computed on VMS and Solaris for replicate 1. On the right are the probabilities of the releases paired by release levels.

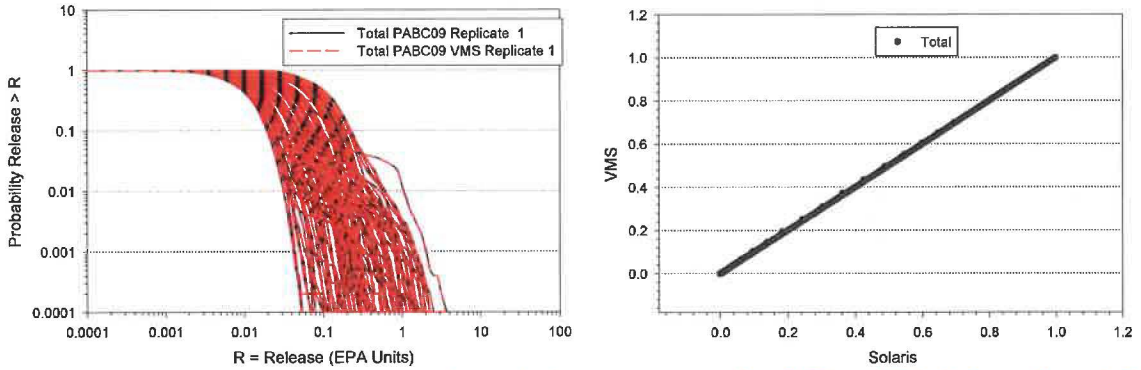

Figure 8. Comparison of CCDFs by vector for total releases as computed on VMS and Solaris for replicate 1. On the right are the probabilities of the releases paired by release levels.

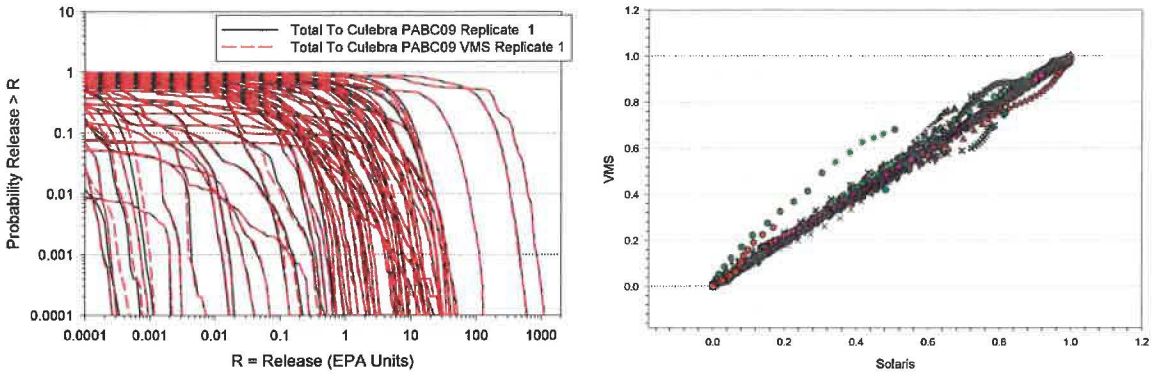

Figure 9. Comparison of CCDFs by vector for releases to the Culebra as computed on VMS and Solaris for replicate 1. On the right are the probabilities of the releases paired by release levels.

Overall the CCDF curves by vector from the VMS calculations and the Solaris calculations were nearly identical and the paired probabilities were tightly distributed around the diagonal of the plot. The releases to the Culebra showed the greatest degree of variability between the probabilities of the VMS and Solaris CCDFs (Fig. 9). This variability between the Solaris and VMS probabilities arises primarily due to the shape of the CCDF curves. The magnitude of the releases to the Culebra depend on the codes BRAGFLO, PANEL and NUTS and the differences in precision (single on VMS to double on Solaris) and floating point format are the likely cause. For example, the green hexagons lying above the diagonal are associated with Vector 25. The magnitude of the difference is due to a very slight change in the CCDF curve (Fig. 10). Because the data are paired based on the release values (X-axis of the left graph in Fig. 10) and because the curves slope steeply downward the small shift in the Solaris CCDF to the left yields a large difference in the vertical distance between the lines.

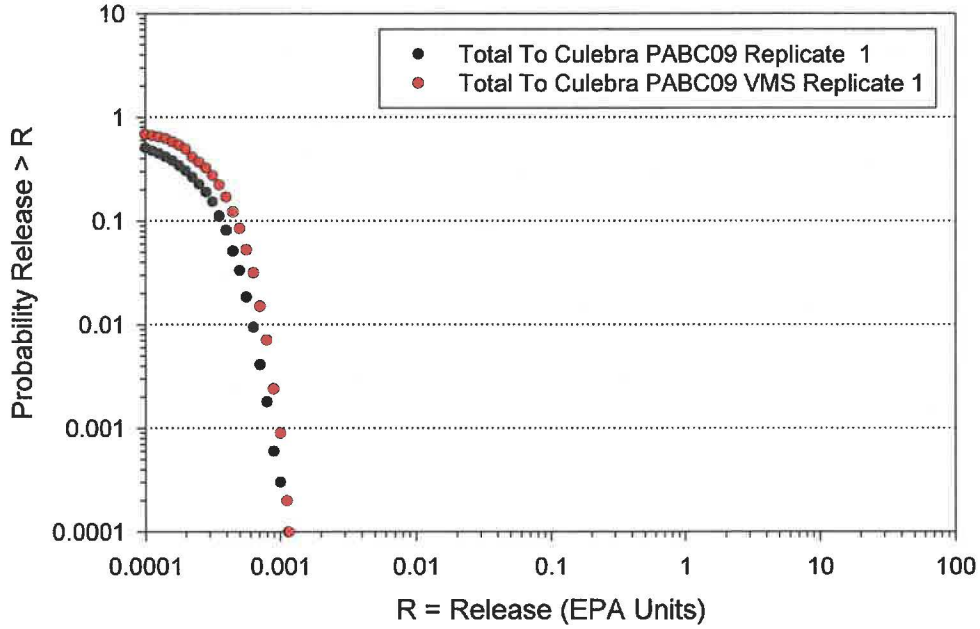

Figure 10. CCDFs for vector 25 Culebra releases from VMS and Solaris.

The results discussed above used the DRSPALL tables generated on Solaris. To provide a complete impact analysis for the port to Solaris DRSPALL results from the CRA-2004 (CRA1) analysis were used as well. The differences between the mean CCDF for spallings computed using the VMS tables versus using the Solaris-generated tables was small (Fig.11). The difference in the mean release at low probabilities is due to a difference in one vector. The results generated exclusively on Solaris will be considered the PABC09 baseline once the VMS system is retired.

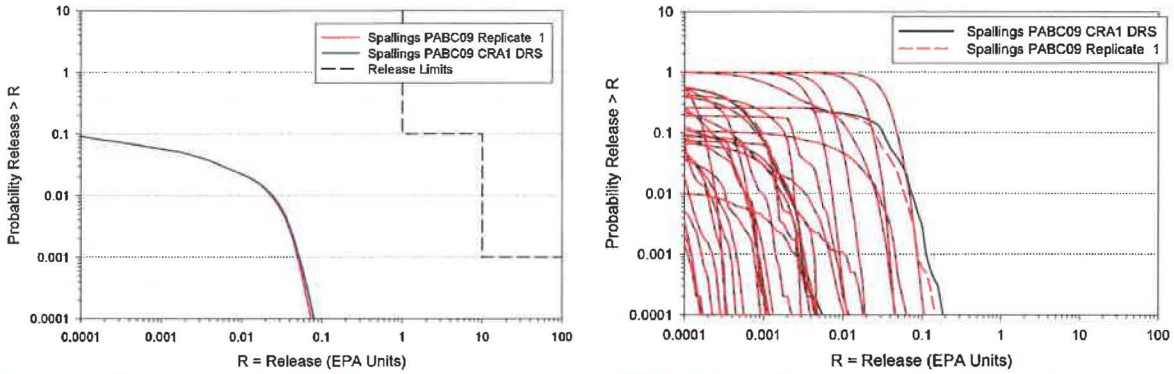

Figure 11. Mean and vector CCDFs resulting from the use of DRSPALL tables generated on VMS and Solaris.

Comparison of CRA14 Results

There are four sets of results for CRA14. These are denoted CRA14BL, CRA14BV, CRA14TP and CRA14-0 (Camphouse 2013). The CRA14BL is the "baseline" case and represents the minimum number of changes from the PABC09 analysis. These changes include the change in panel closures, impacts of mining in the experimental area, changes in the inventory, updates to solubilities, and updates to the drilling rate and distribution of plugging patterns. The CRA14TP adds to the baseline changes in the distribution of BOREHOLE:TAUFAIL, the waste shear strength, and GLOBAL:PBRINE, the probability of a drilling event encountering brine within the boundaries of the repository. The CRA14BV case adds to the TP case the impact of allowing brine volumes and radionuclide concentrations to vary. The CRA14-0 case is the new baseline case and includes all of the previous changes plus refinements to the steel corrosion rate and water balance. Only replicate 1 was run through CCDFGF for the first three CRA14 cases and all three replicates were run for CRA14-0. However, as described in the section "Deviations from Procedures", all three replicates were run for LHS and BRAGFLO in CRA14BL, LHS in CRA14TP and PANEL in CRA14BV in order to provide inputs for the CRA14-0 case.

A comparison of the mean results for all four cases showed few differences between the VMS CCDFs and the Solaris CCDFs (Fig. 12-17).

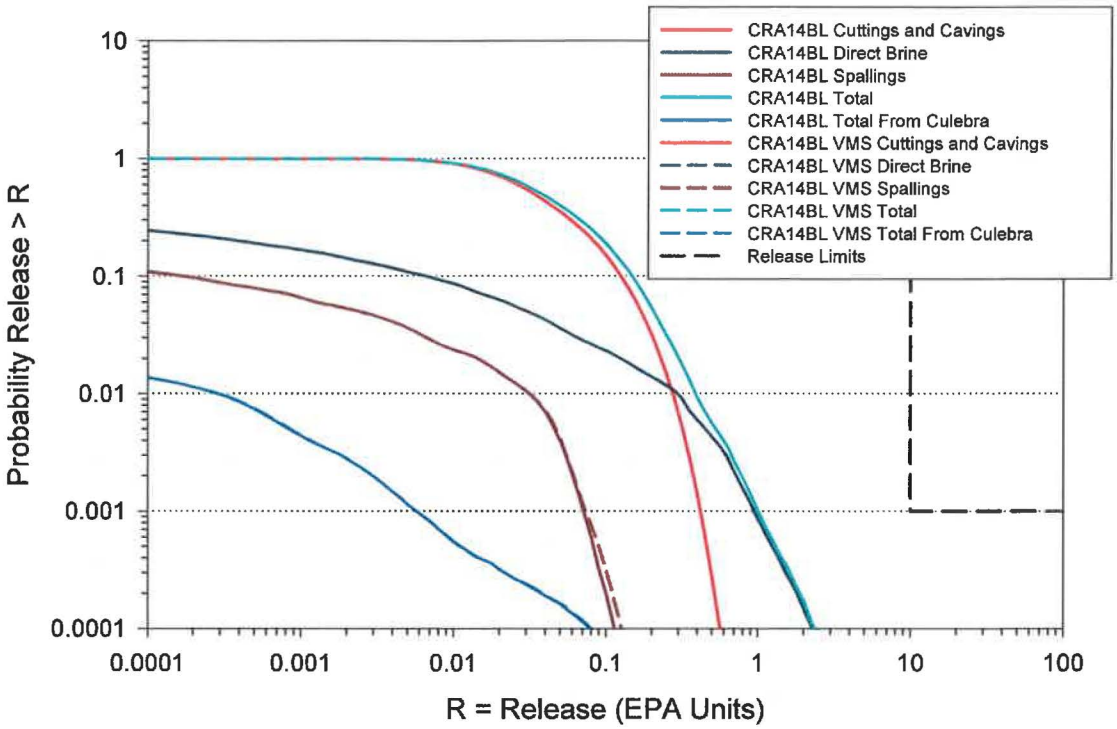

Figure 12. Solaris and VMS mean CCDFs for the CRA14BL case.

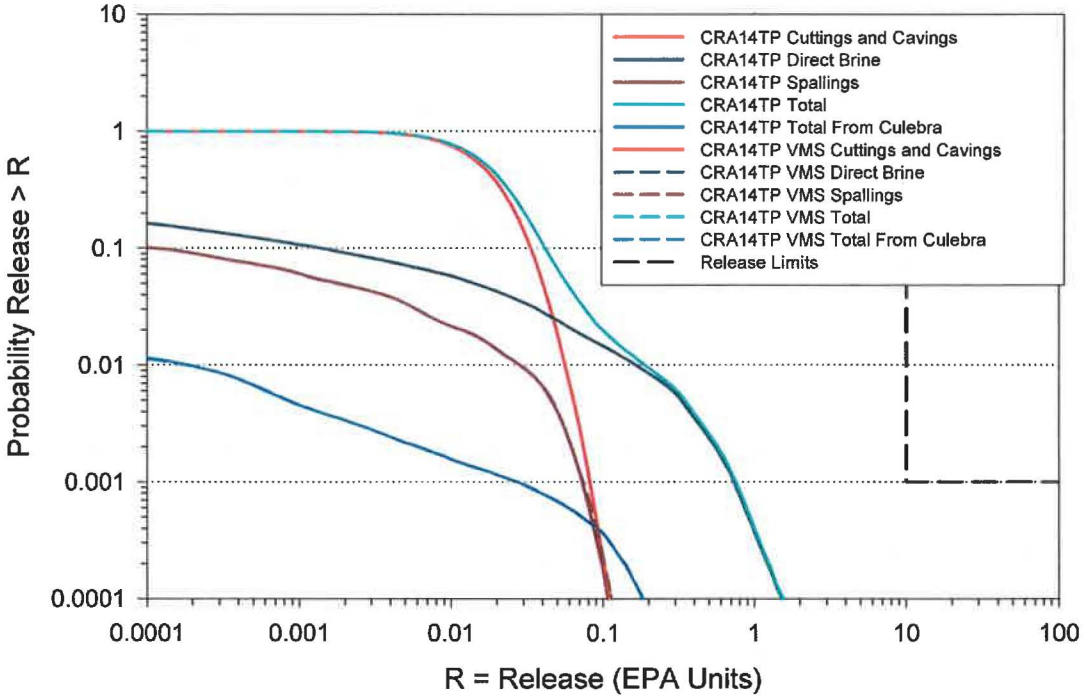

Figure 13. Solaris and VMS mean CCDFs for the CRA14TP case.

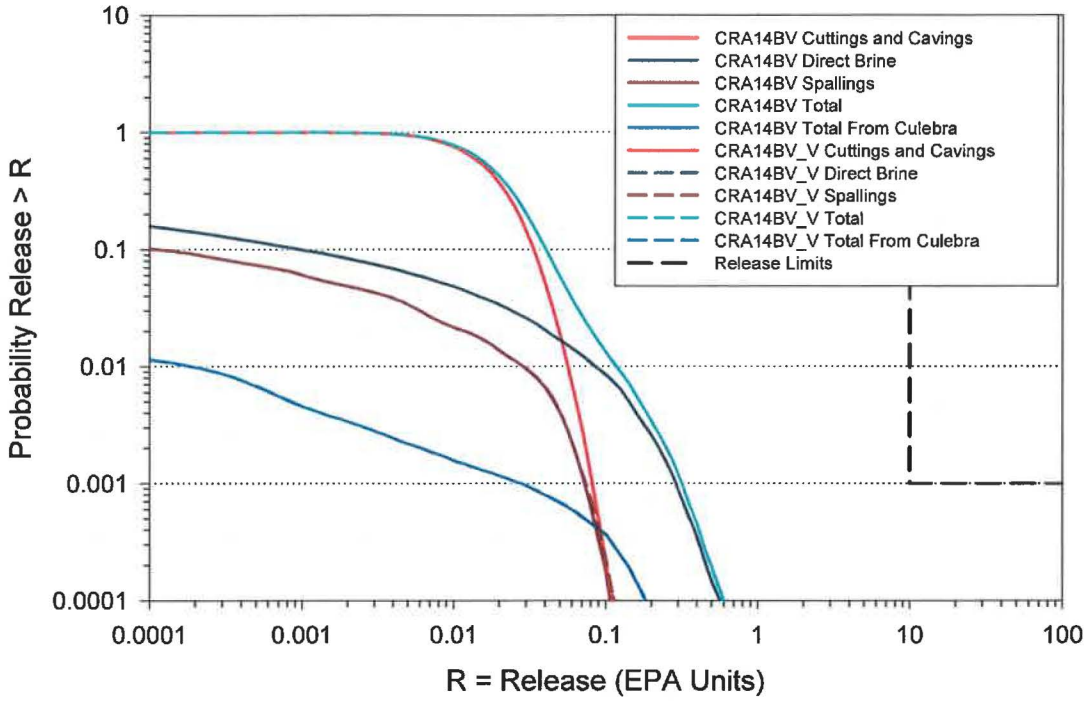

Figure 14. Solaris and VMS mean CCDFs for the CRA14BV case.

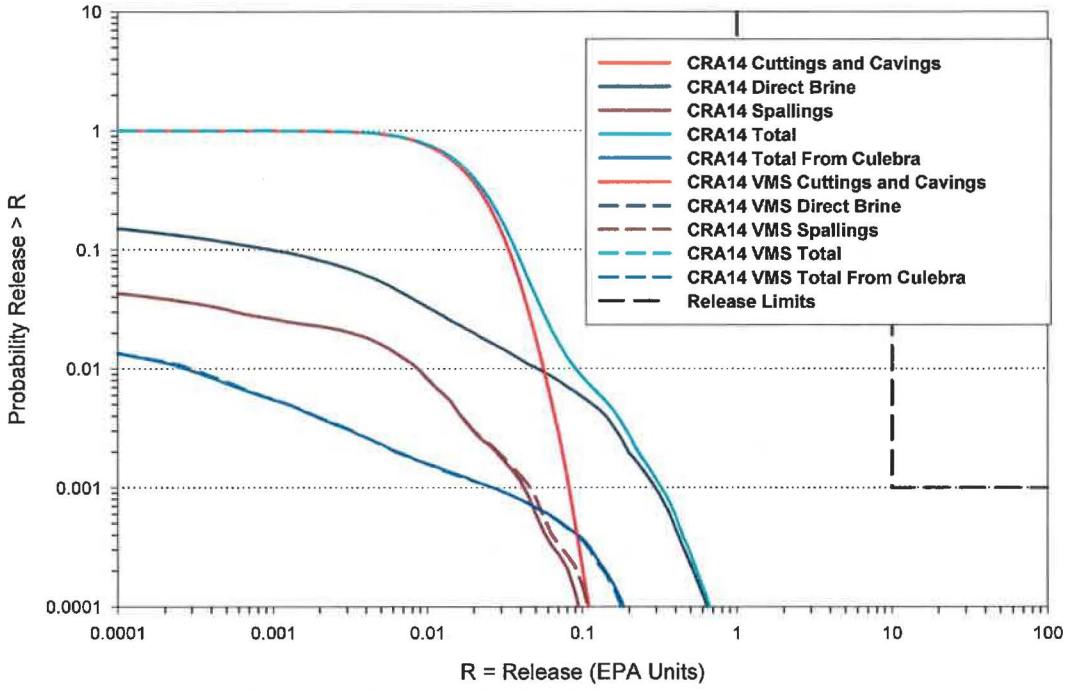

Figure 15. Solaris and VMS mean CCDFs for the CRA14-0 case replicate 1.

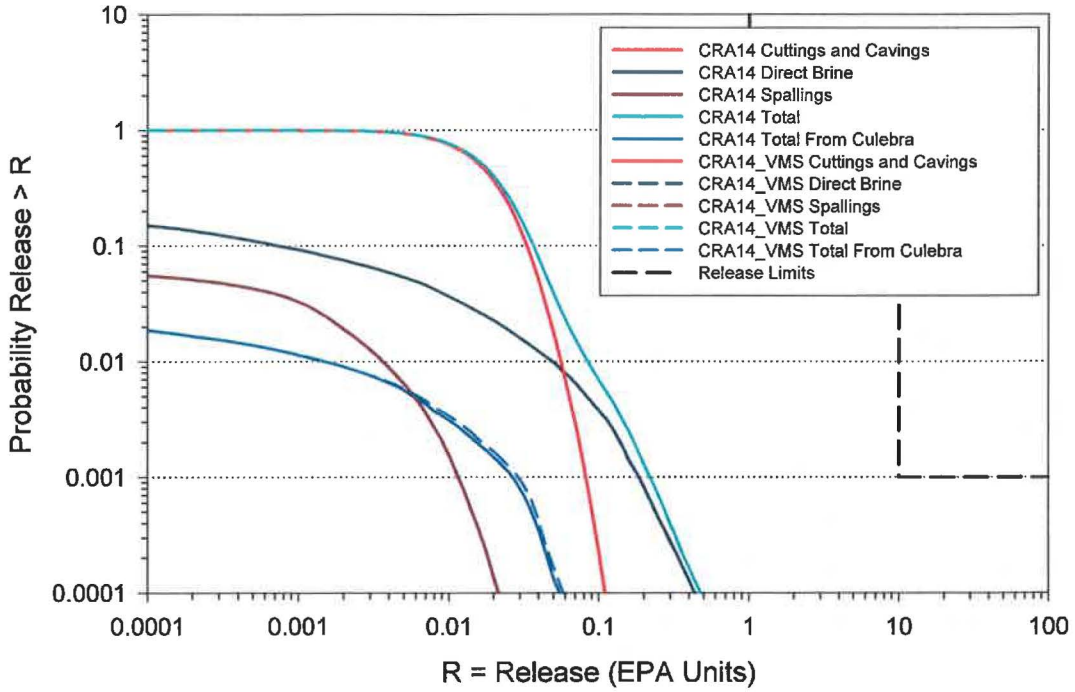

Figure 16. Solaris and VMS mean CCDFs for the CRA14-0 case replicate 2.

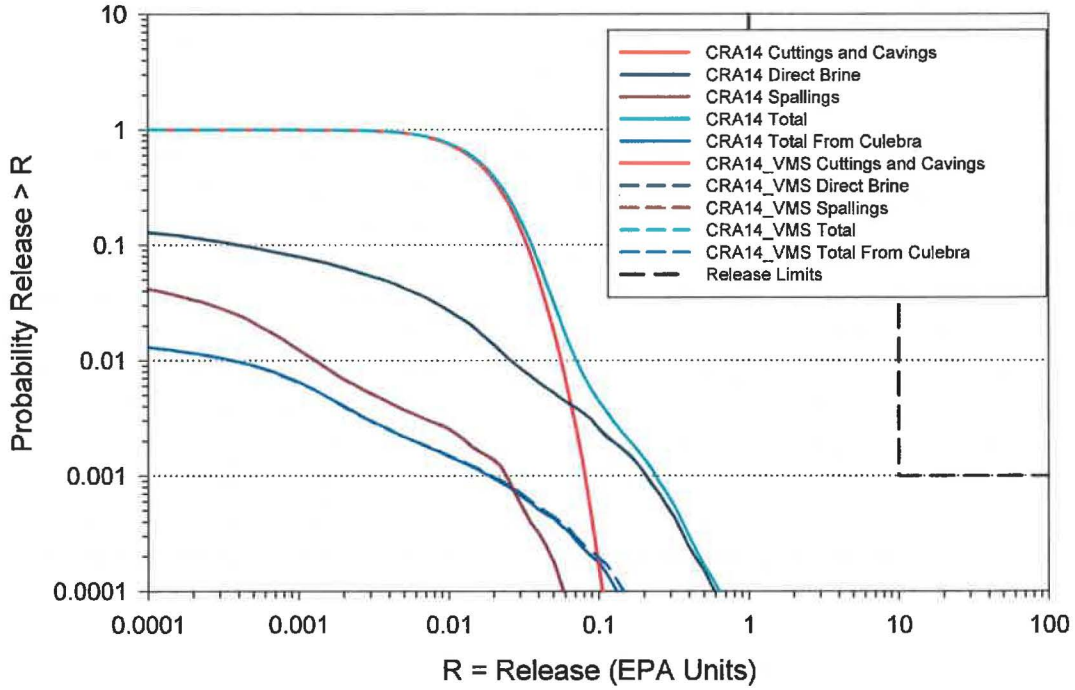

Figure 17. Solaris and VMS mean CCDFs for the CRA14-0 case replicate 3.

The differences at low probabilities in the CCDFs for spallings in Replicate 1of the CRA14-0 case are due to a slight shift toward lower releases of one vector (Fig. 18)

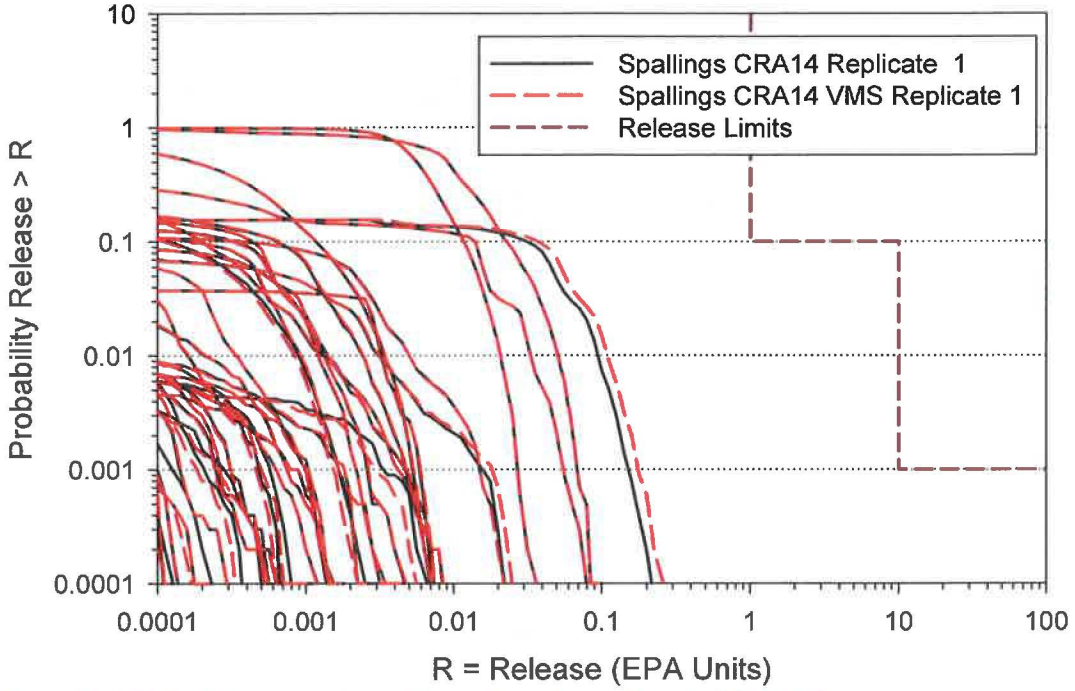

Figure 18. CCDFs of the vectors for spallings releases for replicate 1 of CRA14-0.

Scatter plots of the vector data for replicate 1 of the CRA14-0 analysis run on Solaris are similar to those for the CRA14-0 analysis run on VMS. Most show close agreement with the exception being releases to the Culebra (Fig. 19), as was the case for the PABC-2009 verification test. Figure 20 shows that some vectors are nearly identical while others show shifts in position. Differences in the CCDFs for individual vectors are expected due to the change in numerical precision, going from single precision on VMS to double precision on Solaris. These differences can impact convergence in the solution of the differential equations with the solution using double precision computation (Solaris) expected to be the better result. The magnitude of differences between paired points on the vector-level CCDFs can appear to be large simply due to computing the differences between the curves at fixed values of releases (x-axis) over regions where the curves are nearly vertical. Figure 21 shows the data for two vectors that have relatively large deviations from the diagonal in the plot of releases to the Culebra in Fig. 19; the CCDFs paired by vector number are nearly identical. Across the vectors the differences appears to be both negligible and random as illustrated by the absence of systematic deviation between the two curves (Fig. 22). Thus, while NUTS and PANEL can show somewhat larger differences in releases for some vectors than are seen in other codes, the differences tend to cancel out in the mean release curves and thus do not indicate any systematic errors exist in the Solaris version of the codes.

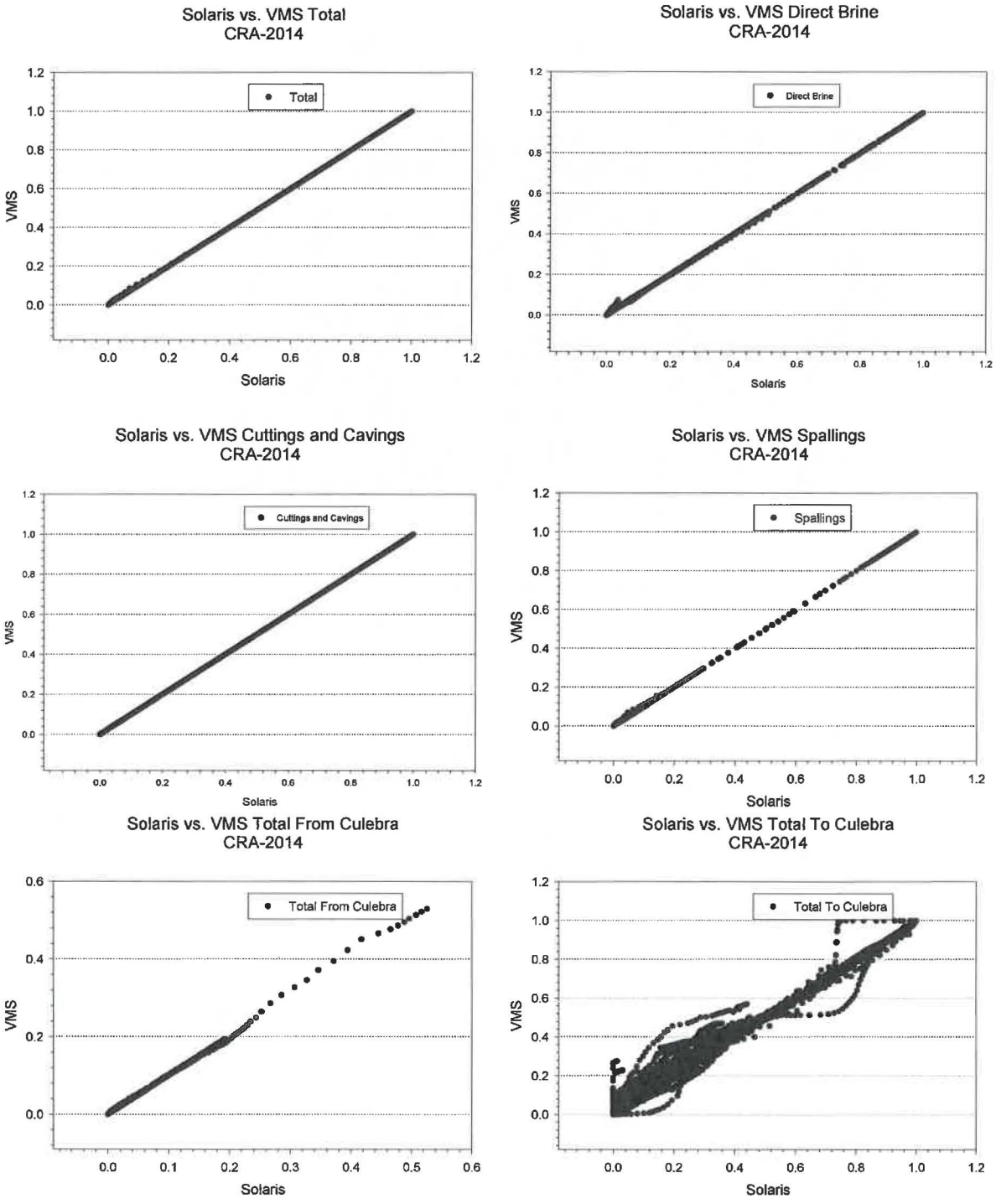

Figure 19. Scatter plots of the vector data comparing VMS to Solaris results.

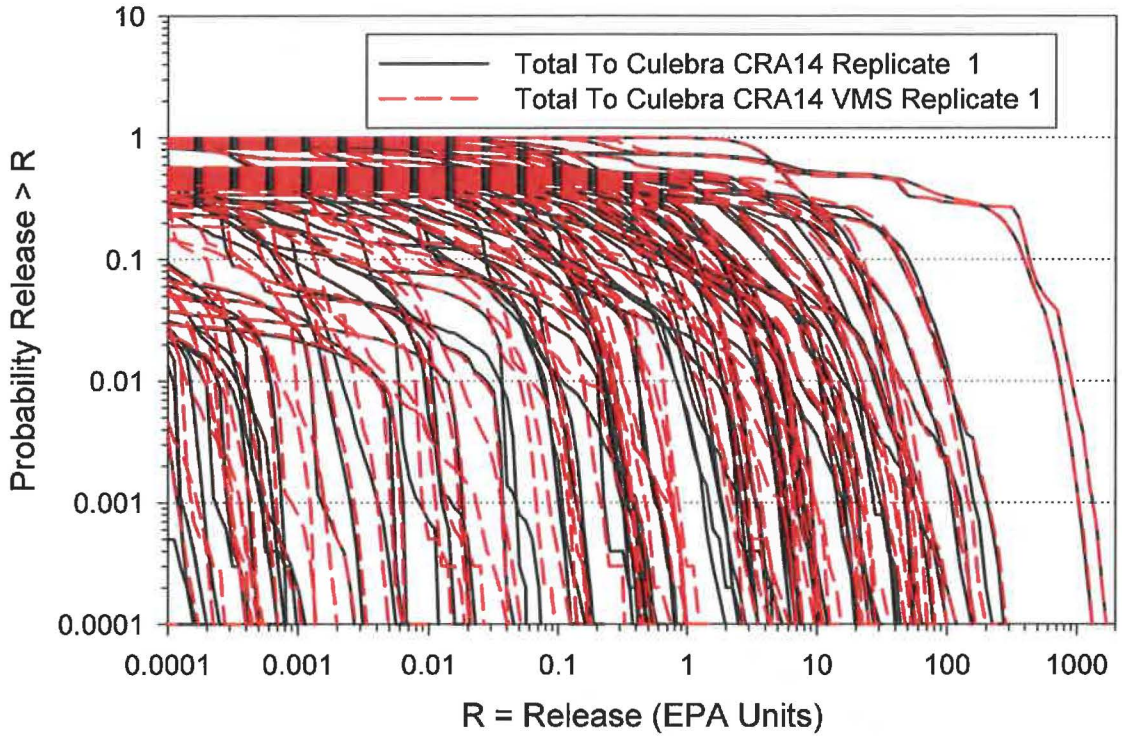

Figure 20. CCDFs by vector for the replicate 1 CRA14 releases to the Culebra.

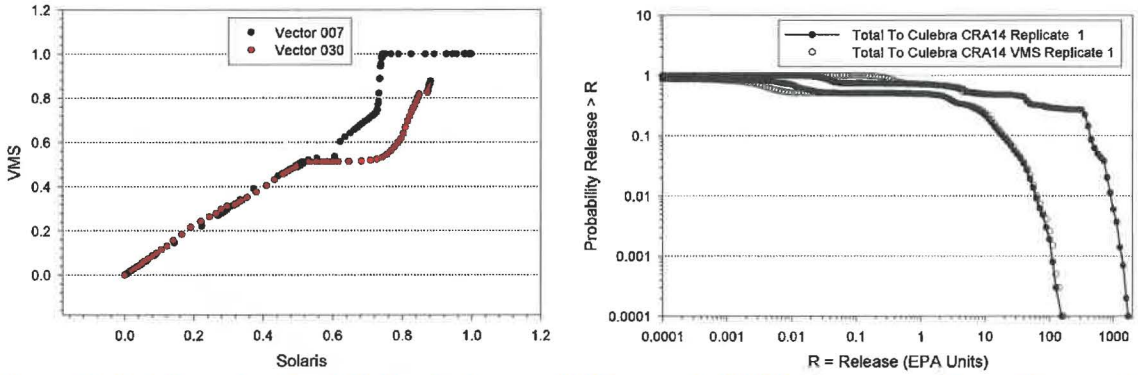

Figure 21. Data for vectors 7 and 30 showing how small differences in CCDFs can appear as large differences in the VMS versus Solaris plot.

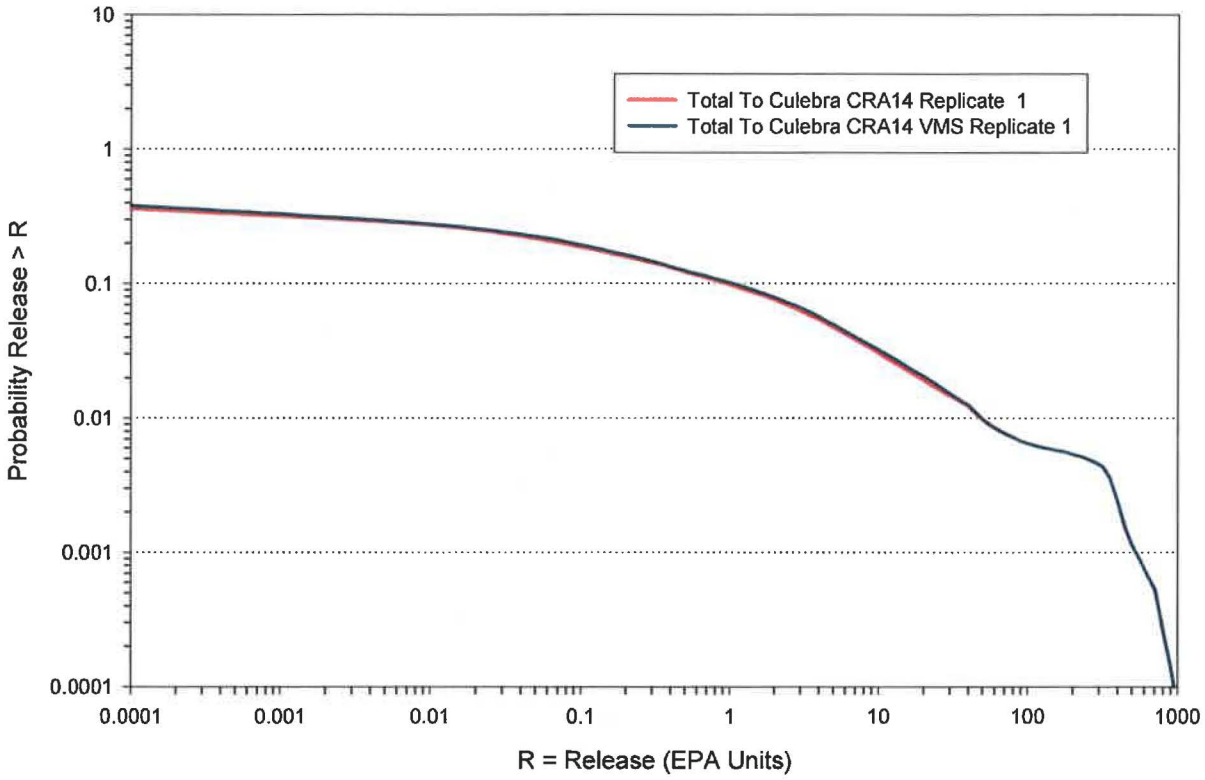

Figure 22. CRA14 releases to the Culebra.

The Solaris results above are based on running DRSPALL on Solaris. However, DRSPALL wasn't rerun for either the PABC09 or CRA-2014 analyses on VMS and those analyses used DRSPALL outputs from the 2004 CRA. Running the PABC09 analysis on Solaris using the DRSPALL output tables created on VMS showed that the differences with the new PABC09 baseline outputs were negligible (Figs. 23-25).

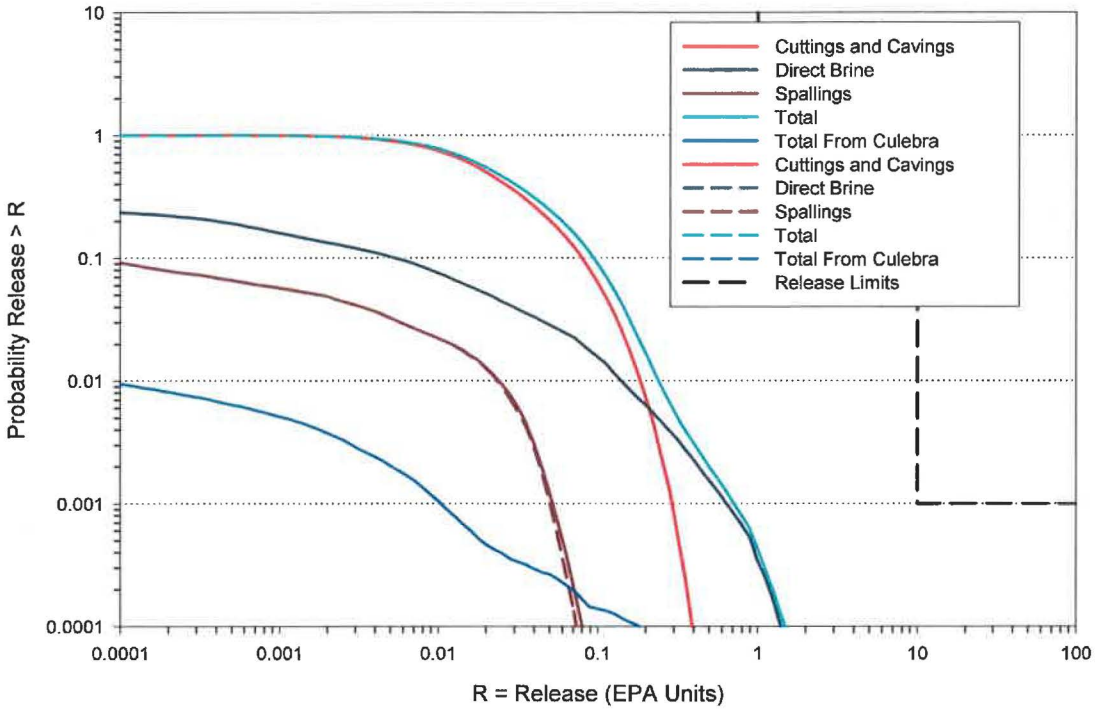

Figure 23. Comparison of mean releases for replicate 1 of the PABC09 based on using the CRA-2004 DRSPALL results (solid lines) and the new PABC09 baseline results based on using the Solaris DRSPALL outputs (dashed lines).

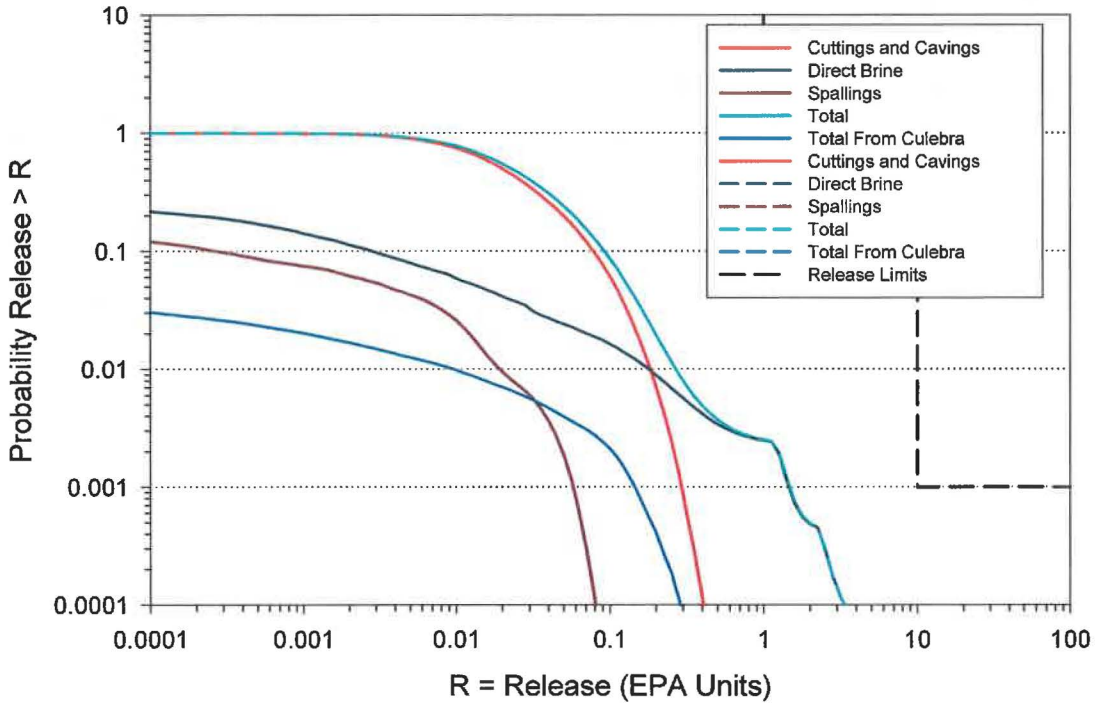

Figure 24. Comparison of mean releases for replicate 2 of the PABC09 based on using the CRA-2004 DRSPALL results (solid lines) and the new PABC09 baseline results based on using the Solaris DRSPALL outputs (dashed lines).

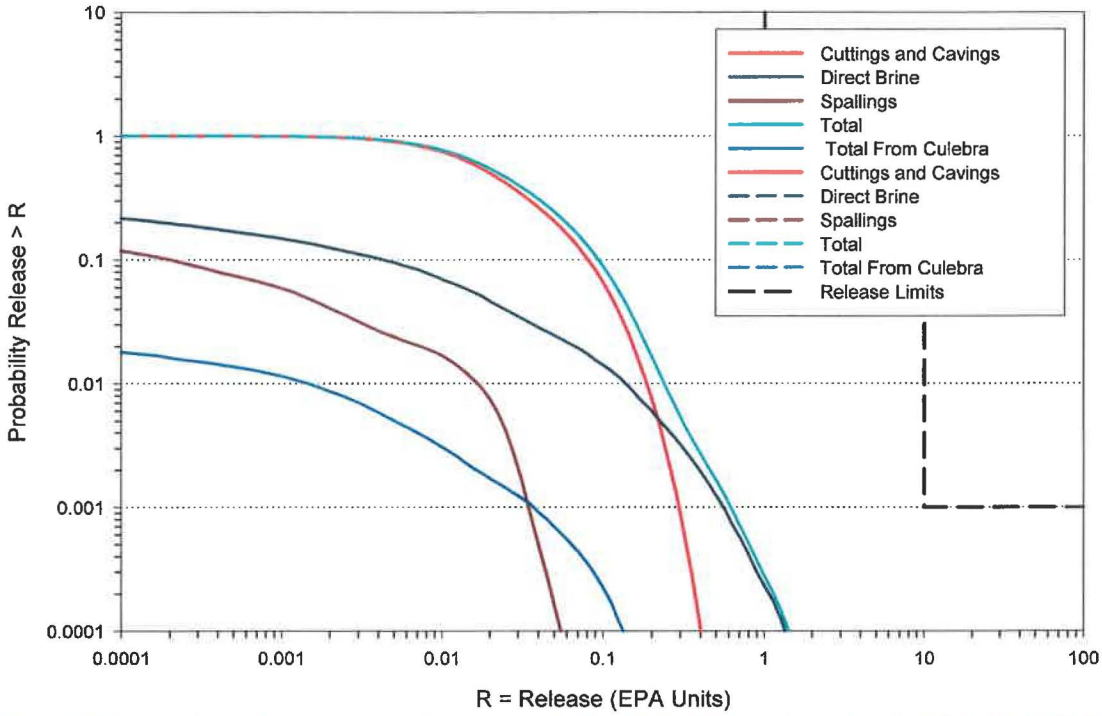

Figure 25. Comparison of mean releases for replicate 3 of the PABC09 based on using the CRA-2004 DRSPALL results (solid lines) and the new PABC09 baseline results based on using the Solaris DRSPALL outputs (dashed lines).

Run Control

The execution of the PABC09 and CRA14 analyses was run under run control by a user designated as the run master. Pertinent input and output files were stored in the CVS code versioning system. The original files were marked "read only" and left in place. Tables detailing the location of output files and examples of the scripts used to execute the codes are found in Appendix 3.

Conclusions

All of the relevant PA codes were ported to and qualified on Solaris as specified in NP 19-1. In addition, the comparison of the results for the PABC09 and CRA14 analyses showed few differences at the vector level, and the mean results were nearly identical. The largest differences observed were due to small horizontal (release) shifts in the CCDF curve near the right tail (low probability) of the distributions. Execution times were significantly reduced in all cases on Solaris compared to VMS. The saving of LHS and CCDFGF results directly to a database will improve both accessibility and traceability of the results. The Python scripts used to run these analyses are written for general application by both the run master for analyses run under QA and by analysts running jobs outside of QA. Because the scripts automatically determine whether a user is authorized to write files into CVS, analysts can use the same scripts as the run_master user without fear of corrupting the CVS repositories. Thus PAs can now be run faster and with a lower risk of human error than was possible on VMS. Based on the results included herein, the

Solaris system is operational, can reproduce to an acceptable level the last two compliance recertification PAs run on VMS and is fully capable of running future PAs.

References

Camphouse, R.C. 2013. Analysis Plan for the 2014 WIPP Compliance Recertification Application Performance Assessment. Sandia National Laboratories, Carlsbad, NM. ERMS #559198.

Kirchner, T. 2012a. AP-162 Revision 0: Analysis Plan for Migration of the Performance Assessment Codes to the Sun Solaris Blade Server Running with Intel Processors. Sandia National Laboratories, Carlsbad, NM. ERMS #557765

Kirchner, T. 2012b. Report on the Migration of the WIPP PA Parameter Database from SQL Server to MySQL. ERMS #557128

Long, J. 2012. Nuclear Waste Management Procedure NP 9-1: Analyses Revision 14. Carlsbad, NM: Sandia National Laboratories. ERMS #558215

Safely, G. 2012. Nuclear Waste Management Procedure NP 9-1: Analyses Revision 9. Carlsbad, NM: Sandia National Laboratories. ERMS #558879

Appendix 1: Qualification of MergeSpall

Utility MERGESPALL is qualified under NP 9-1 (Chavez, 2001). It was first qualified for CRA-2004. See Appendix C of Analysis Package for DRSPALL: CRA-2004 Performance Assessment Baseline Calculation (ERMS #540415), authored by Eric Vugrin, 2005.

This document describes MERGESPALL Version 1.01, on the Solaris running SunOS 5.11.

Description and Requirements

MERGESPALL reads a set of SUMMARIZE tables to create an input file for the code CUTTINGS_S. MERGESPALL has the following set of requirements:

1. MERGESPALL reads a SUMMARIZE output file containing the DRSPALL data for multiple times and vectors from the spherical calculations for a single DRSPALL pressure scenario (DPS).
2. MERGESPALL reads a SUMMARIZE output file containing the DRSPALL data for multiple times and vectors from the cylindrical calculations for a single DPS (if necessary).
3. For each vector, MERGESPALL adds the spall volume from the spherical run at the first time that CAVRAD (cavity radius) exceeds REPOSTCK (repository thickness) to the final spall volume from the cylindrical run for the same DPS. If CAVRAD does not exceed REPOSTCK for a vector, MERGESPALL selects the spall volume at the final time from the spherical run and no cylindrical SUMMARIZE file is required.
4. If MERGESPALL does not find a cylindrical spall volume for a vector that requires one, MERGESPALL aborts with an error status.
5. MERGESPALL writes an output text file with a three-line header and the calculated spall volumes for each vector. If the output text file already exists, MERGESPALL appends the spall volume data (no header) to the output file.

Changes to the Code

MERGESPALL was migrated from the current VMS Alpha platform to a Solaris Blade with SunOS 5.11. The following modifications were made to the source code:

- The command line arguments are processed with standard CAMCON_LIB routines.
- The code aborts with an error status if a fatal error is encountered.
- All real variables are now double precision.
- Minor code cleanup, such as using "trim" to print strings.
- Minor cosmetic changes may be visible in the output.

Build Information

Code Name:	MERGESPALL
Version:	1.01
CVS Repository:	$CVSLIB/WIPP_CODES/PA_CODES/MERGESPALL ($CVSLIB is /nfs/data/CVSLIB)
Source code:	Source/MergeSpall.f
Executable:	Build/Solaris/mergespall (in $CVSLIB/WIPP_CODES/PA_CODES/MERGESPALL)
Build Date/Time:	Mon Aug 26 16:39:51 2013
Build Size:	1813184 bytes
Total Size:	1813925 bytes
Executable Tags:	MERGESPALL_101, ver_1_01
Platform:	Oracle SunFire X2270 server / SunOS 5.11 11.0 i86pc i386 i86pc
Host:	tsb.sandia.gov
Compiler:	Sun Fortran 95 8.6 SunOS_i386 2011/11/16
Build log file:	RunControl/Solaris/mergespall_101_Build.rtf

Code Execution and Files

User Interactions with the Software

MERGESPALL expects a number of command line arguments, listed below. The program arguments may either be positional (provided in order) or flagged (preceded by the appropriate flag). Flags always start with "–"; they may be abbreviated. All positional arguments must be listed before any flagged arguments. The only difference between using positional and flagged arguments is that optional arguments may be omitted for flagged arguments.

1. –output The output data file. If the file exists, data will be appended to the file.
2. –log The output log file. If the file exists, log info will be appended to the file.
3. –sph The spherical data file.
4. –cyl The optional cylindrical data file.

Input Files

Two input files are used by MERGESPALL: one contains the DRSPALL data for multiple times and vectors from the spherical calculations for a single DPS, and one contains the DRSPALL data for a single time and multiple vectors from the cylindrical calculations for a single DPS. (If the cylindrical file contains multiple times, only the last time is used.) The first two lines of both files consist of a header, followed by a blank line. The spherical file has five columns: 1) vector

number, 2) time, 3) REPOSTCK value, 4) CAVRAD, and 5) SPLVOL2. REPOSTCK values are constant for all times, but vary by vector. CAVRAD and SPLVOL2 vary by vector and increase with time. The cylindrical file contains three columns: 1) vector, 2) time, and 3) SPLVOL2. Example input files can be found in Test/Input of the CVS directory.

Output Files

MERGESPALL generates a table for the CUTTINGS_S code. The output data file has a three-line header, followed by the calculated spall volumes. If it does not, it writes a three-line header and writes the calculated spall volumes. The three-line header contains: 1) the number of vectors, 2) number of DPSs, and 3) the initial repository pressure used for each DPS. MERGESPALL assumes four pressure scenarios with initial pressures of 10, 12, 14, and 14.8 MPa. The data lines contain three columns: 1) vector, 2) time, and 3) spall volume. Note that if the output data file already exists, the spall data (but no header) will be appended to the file.

MERGESPALL also generates a log file. As with the output data file, if the log file already exists, the log information is appended to the file.

Regression Test

MERGESPALL Version 1.00 was validated on a Compaq ES40 running OpenVMS 7.2-1 for the CRA-2004 Performance Assessment Baseline Calculation (ERMS #540415, Vugrin, 2005). Regression testing against the VMS results was conducted to demonstrate the validity of MERGESPALL Version 1.01 on a Solaris Blade with SunOS 5.11.

MERGESPALL Version 1.01 was tested in the following environment:

Platform:	Oracle Sun X6270 M2 Blade / SunOS 5.11 11.0 i86pc i386 i86pc
Host:	gd.sandia.gov
Test Date:	August 26, 2013

All files related to validation testing are stored in the MERGESPALL CVS Repository under the following test directories.

CVS Repository:	$CVSLIB/WIPP_CODES/PA_CODES/MERGESPALL
Log file:	Test/RunControl/Solaris/mergespall_101_Test.rtf
Input files:	Test/Input (working directory ./Input)
Output files:	Test/Output/Solaris_101/* (working directory MERGESPALL/Test/Output/Solaris_101)
V1.00 Output:	Test/Output/VMS_100 (working directory ./Regr)

The MERGESPALL test suite consists of three test cases. Each test is briefly described below. Each test sub-section contains the command line used to run MERGESPALL for the test. The command line indicates the test input files and test output files. The output file is then differenced with the output file from the MERGESPALL 1.00 VMS test using the Unix diff

command. Floating point values must match to six digits. The output is written in free-format, so differences in the numerical formats are expected.

Test Case #1

The purpose of Test Case #1 is to evaluate the MERGESPALL calculations when CAVRAD does not exceed REPOSTCK for any time or vectors. MERGESPALL should simply return the final SPLVOL2 values for all vectors. No cylindrical input files are needed. Additionally this test case verifies that MERGESPALL formats the output file properly.

Below is the portion of the log file mergespall_101_test.rtf listing the output from the MERGESPALL execution and diff of the output files. The output files show no significant differences.

```
./mergespall MERGESPALL/Test/Output/Solaris_101/mergespall_101_test1.out
        MERGESPALL/Test/Output/Solaris_101/mergespall_101_test1.log
        ./Input/sum_test_p1.tbl CANCEL
  Output file=MERGESPALL/Test/Output/Solaris_101/mergespall_101_test1.out
  Log file=MERGESPALL/Test/Output/Solaris_101/mergespall_101_test1.log
Status for test1 = 0
diff -w MERGESPALL/Test/Output/Solaris_101/mergespall_101_test1.out
        ./Regr/test1.out
        | tee MERGESPALL/Test/Output/Solaris_101/mergespall_101_test1.dif
4,6c4,6
<   51.0 600.0 2.033756
<   74.0 600.0 1.029988
<   97.0 600.0 0.101322
---
>     51.0000000000000        600.000000000000        2.03375600000000
>     74.0000000000000        600.000000000000        1.02998800000000
>     97.0000000000000        600.000000000000       0.101322000000000
```

Test Case #2

The purpose of Test Case #2 is to evaluate the MERGESPALL calculations when CAVRAD exceeds REPOSTCK for multiple vectors. MERGESPALL should identify in which vectors CAVRAD exceeds REPOSTCK and add spall volumes from the cylindrical spall data input file to the spall volumes from the spherical spall data file. Additionally this test case verifies that MERGESPALL appends the spall data to a preexisting file.

Below is the portion of the log file mergespall_101_test.rtf listing the output from the MERGESPALL execution and diff of the output files. The output files show no significant differences. *Note that the test script copies output file from Test Case #1 to the output file for Test Case #2 before the start of the test, to verify that MERGESPALL will append data to a preexisting file.*

```
cp -p MERGESPALL/Test/Output/Solaris_101/mergespall_101_test1.out
        MERGESPALL/Test/Output/Solaris_101/mergespall_101_test2.out
```

```
./mergespall MERGESPALL/Test/Output/Solaris_101/mergespall_101_test2.out
        MERGESPALL/Test/Output/Solaris_101/mergespall_101_test2.log
        ./Input/sum_test_p2.tbl
        ./Input/sum_cyl_test_p2.tbl
  Output file=MERGESPALL/Test/Output/Solaris_101/mergespall_101_test2.out
  Log file=MERGESPALL/Test/Output/Solaris_101/mergespall_101_test2.log
Status for test2 = 0
diff -w MERGESPALL/Test/Output/Solaris_101/mergespall_101_test2.out
        ./Regr/test2.out
        | tee MERGESPALL/Test/Output/Solaris_101/mergespall_101_test2.dif
4,6c4,6
<   51.0 600.0 2.033756
<   74.0 600.0 1.029988
<   97.0 600.0 0.101322
---
>     51.0000000000000         600.000000000000         2.03375600000000
>     74.0000000000000         600.000000000000         1.02998800000000
>     97.0000000000000         600.000000000000        0.101322000000000
8,10c8,10
<   51.0 600.0 14.489024100000002
<   74.0 600.0 3.565143
<   97.0 600.0 7.6753184
---
>     51.0000000000000         600.000000000000         14.4890241000000
>     74.0000000000000         600.000000000000         3.56514300000000
>     97.0000000000000         600.000000000000         7.67531840000000
```

Test Case #3

The purpose of Test Case #3 is to verify that MERGESPALL aborts with an error status when CAVRAD exceeds REPOSTCK for a vector and no cylindrical spall volume is given. For MERGESPALL 1.00, an error message was written to the log file, but the code did not abort.

Below is the portion of the log file mergespall_101_test.rtf listing the output from the MERGESPALL execution and diff of the output files. The code aborts with an error message and an error status of 1, as required. The output files show no significant differences.

```
./mergespall MERGESPALL/Test/Output/Solaris_101/mergespall_101_test3.out
        MERGESPALL/Test/Output/Solaris_101/mergespall_101_test3.log
        ./Input/sum_test_p2.tbl
        ./Input/sum_cyl_test_p2_no97.tbl
  Output file=MERGESPALL/Test/Output/Solaris_101/mergespall_101_test3.out
  Log file=MERGESPALL/Test/Output/Solaris_101/mergespall_101_test3.log
  Time= 360.0 CAVRAD= 0.9430975 Repostck= 0.9394274 7.135302
  ERROR: No cylindrical volume was input
  Spherical Input File=./Input/sum_test_p2.tbl
  Cylindrical Input File=./Input/sum_cyl_test_p2_no97.tbl

  %%% FATAL ERROR - Errors in processing
  %%% Program is aborting %%%
```

```
Status for test3 = 1
diff -w MERGESPALL/Test/Output/Solaris_101/mergespall_101_test3.out
        ./Regr/test3.out
        | tee MERGESPALL/Test/Output/Solaris_101/mergespall_101_test3.dif
4,6c4,6
<   51.0 600.0 14.489024100000002
<   74.0 600.0 3.565143
<   97.0 600.0 7.135302
---
>     51.0000000000000          600.000000000000          14.4890241000000
>     74.0000000000000          600.000000000000          3.56514300000000
>     97.0000000000000          600.000000000000          7.13530200000000
```

Appendix 2: Qualification of SCREEN_NUTS

Utility SCREEN_NUTS is qualified under NP 9-1 (Chavez, 2001). It was originally qualified on VMS as SCREEN (or SCREEN.FOR) Version 1.00.

This document describes SCREEN_NUTS Version 1.01, on the Solaris running SunOS 5.11.

Description

The SCREEN_NUTS utility lists vectors that are "screened-in" for use in the full transport simulations. SCREEN_NUTS is run for each scenario. A vector is automatically screened-in for scenario S1 if it was screened in for any other scenario, regardless of the tracer transport results. This is done because the undisturbed simulation results are needed as initial conditions to compute the consequences of intrusions.

Normally the SCREEN_NUTS utility is run as part of a NUTS calculation. First, the NUTS code is run in "screening mode" to compute the transport of a conservative tracer for BRAGFLO scenarios S1-S5. The ALGEBRACDB code is run on each NUTS output CAMDAT file to calculate the variables needed by SCREEN_NUTS. The SUMMARIZE code is run, once for each scenario, to extract certain variables from the output CAMDAT files for all 100 vectors. The SCREEN_NUTS utility is then used to list vectors that are "screened-in" for use in the full transport simulations. NUTS is then run in "isotope mode" and "intrusion mode" on those vectors only.

SCREEN_NUTS reads variables representing the integration of masses for each vector. The output file lists all vectors that are screened-in for that scenario. A vector is screened-in if the markerbed and/or borehole values exceed an input tolerance. The markerbed value is the sum of SMB38N1C + SMB38S1C + SMB39N1C + SMB39S1C + SMBABN1C + SMBABS1C. The borehole value is the sum of SHUP1C + BHUP1C. For scenario 1, the output file also lists the "union" of the screened-in vectors for all scenarios in a separate list.

Changes to the Code

SCREEN_NUTS was migrated from the current VMS Alpha platform to a Solaris Blade with SunOS 5.11. The following modifications were made to the source code:

- The code was renamed to SCREEN_NUTS because "screen" is a Solaris command.
- The command line arguments have changed.
- The input control file is no longer needed. This file contained: the analysis, the replicate, the number of scenarios, the number of vectors, the scenario numbers, the tolerance value, the input files, and the output files. Most of these are now command line arguments.
- The code aborts with an error status if a fatal error is encountered.
- All real variables are now double precision.
- Minor code cleanup, such as using "trim" to print strings.

- Minor cosmetic changes may be visible in the output.

Build Information

Code Name:	SCREEN_NUTS
Version:	1.01
CVS Repository:	$CVSLIB/WIPP_CODES/PA_CODES/SCREEN_NUTS ($CVSLIB is /nfs/data/CVSLIB)
Source code:	Source/Screen_Nuts.f
Executable:	Build/Solaris/screen_nuts (in $CVSLIB/WIPP_CODES/PA_CODES/SCREEN_NUTS)
Build Date/Time:	Thu Aug 29 13:16:27 2013
Build Size:	1804504 bytes
Total Size:	1805245 bytes
Executable Tags:	SCREEN_NUTS_101, ver_1_01
Platform:	Oracle SunFire X2270 server / SunOS 5.11 11.0 i86pc i386 i86pc
Host:	tsb.sandia.gov
Compiler:	Sun Fortran 95 8.6 SunOS_i386 2011/11/16
Build log file:	RunControl/Solaris/screen_nuts_101_Build.rtf

Code Execution and Files

User Interactions with the Software

SCREEN_NUTS expects a number of command line arguments, listed below. The program arguments may either be positional (provided in order) or flagged (preceded by the appropriate flag). Flags always start with "–"; they may be abbreviated. All positional arguments must be listed before any flagged arguments. The only difference between using positional and flagged arguments is that optional arguments may be omitted for flagged arguments.

5. –sum The file name template for the SUMMARIZE file. The template must contain "^" in place of the scenario number.
6. –output The file name template for the SUMMARIZE file. The template must contain "^" in place of the scenario number.
7. –tolerance The tolerance value. The default for this optional argument is 1D-7. A vector is screened-in if the sum of its markerbed and/or borehole values exceed this tolerance.
8. –scenarios The number of scenarios. The default for this optional argument is 5. The scenarios are numbered 1 to the number of scenarios.

Input Files

A SUMMARIZE input file is required for each scenario. The first three lines are header lines and are ignored. Data lines follow for each vector. Each line has the following columns: 1) vector number, 2) time, 3) SMB38N1C, 4) SMB38S1C, 5) SMBABN1C, 6) SMBABS1C, 7) SMB39N1C, 8) SMB39S1C, 9) BHUP1C, and 10) SHUP1C. These columns represent the values of global variables on the final time step of the output CAMDAT file from the NUTS code. Note that there must be a blank line (or *break line) between the lines for each vector, and only one time step is allowed. SCREEN_NUTS reads all vectors on a file. Example input files can be found in Test/Input of the CVS directory.

Output Files

SCREEN_NUTS generates an output file for each scenario. Each file has a header. (Note that the analysis and replicate no longer appear in the header.) The screened-in vectors for the scenario are listed one per line between lines "NONUNION_BEGIN" and "NONUNION_END". Each line also lists the type of tolerance exceeded (markerbed, or borehole or both) and the value that exceeded the tolerance. The output file for Scenario 1 also lists the union of all screened-in vectors for all scenarios. These vectors are listed one per line, in numerical order with no repetitions, between lines "UNION_BEGIN" and "UNION_END".

Regression Test

SCREEN Version 1.00 was validated on a Compaq ES47 running OpenVMS 8.2 for the CRA-2009 PA (Ismail and Garner, 2008). Regression testing against the VMS results was conducted to demonstrate the validity of SCREEN_NUTS Version 1.01 on a Solaris Blade with SunOS 5.11.

SCREEN_NUTS Version 1.01 was tested in the following environment:

Platform:	Oracle Sun X6270 M2 Blade / SunOS 5.11 11.0 i86pc i386 i86pc
Host:	gd.sandia.gov
Test Date:	August 29, 2013

All files related to validation testing are stored in the SCREEN_NUTS CVS Repository under the following test directories.

CVS Repository:	$CVSLIB/WIPP_CODES/PA_CODES/SCREEN_NUTS
Log file:	Test/RunControl/Solaris/SCREEN_NUTS_101_Test.rtf
Input files:	Test/Input (working directory ./Input)
Output files:	Test/Output/Solaris_101/* (working directory SCREEN_NUTS/Test/Output/Solaris_101)
V1.00 Output:	Test/Output/VMS_100 (working directory ./Regr)

The SCREEN_NUTS test suite consists of a single test case, with two scenarios. Each test is briefly described below. Each test sub-section contains the command line used to run

SCREEN_NUTS for the test. The command line indicates the test input files and test output files. The output file for each scenario is then differenced with the output file from the SCREEN_NUTS 1.00 VMS test using the UNIX diff command. Floating point values must match to six digits. The output is written in free-format, so differences in the numerical formats are expected.

Test Case #1

Test Case #1 inputs SUMMARIZE files for two scenarios (screen_test1_s^.tbl), each with data for 12 vectors. Two output files (screen_101_test1_s^.out) are generated and compared with the output from SCREEN 1.00.

Below is the portion of the log file screen_nuts_101_test.rtf listing the SCREEN_NUTS command line and the diff of the output files for each scenario. Each output file diff shows that the analysis and replicate lines are missing from the output file headers, the input file name is in lower case with a directory, and the tolerance is in a different format. These changes are acceptable. The output files show no significant differences.

```
./screen_nuts ./Input/screen_test1_s^.tbl
        SCREEN_NUTS/Test/Output/Solaris_101/screen_101_test1_s^.out
        -tol 1d-10 -scen 2

  Tolerance:  1.0E-10
  Scenarios:  2

  Scenario  2 : ./Input/screen_test1_s2.tbl

      SCREEN_NUTS/Test/Output/Solaris_101/screen_101_test1_s2.out
  Scenario  1 : ./Input/screen_test1_s1.tbl

      SCREEN_NUTS/Test/Output/Solaris_101/screen_101_test1_s1.out
 STOP: SCREEN Normal Completion
Status for test1 = 0
diff -w SCREEN_NUTS/Test/Output/Solaris_101/screen_101_test1_s1.out
        ./Regr/screen_test1_s1.out
        | tee SCREEN_NUTS/Test/Output/Solaris_101/screen_101_test1_s1.dif
 1c1,3
 <  data source: ./Input/screen_test1_s1.tbl
 ---
 >  analysis: CRA1BC
 >  replicate:            1
 >  data source: SCREEN_TEST1_S1.TBL
 3c5
 <  nuts tolerance:  1.0E-10
 ---
 >  nuts tolerance:   1.000000000000000E-010
diff -w SCREEN_NUTS/Test/Output/Solaris_101/screen_101_test1_s2.out
        ./Regr/screen_test1_s2.out
        | tee SCREEN_NUTS/Test/Output/Solaris_101/screen_101_test1_s2.dif
 1c1,3
```

```
<  data source: ./Input/screen_test1_s2.tbl
---
>  analysis: CRA1BC
>  replicate:            1
>  data source: SCREEN_TEST1_S2.TBL
3c5
<  nuts tolerance:  1.0E-10
---
>  nuts tolerance:   1.000000000000000E-010
```

Appendix 3: Run Control

The PABC09 and CRA14 analyses run on Solaris were run using a set of Python scripts. The lowest-level components are contained in the rc.py module, found in $CVSLIB/bin. This module provides the routines for managing the export and import of files into CVS, verifying the presence of all input files and the creation of all output files, parsing and interpreting the command line arguments, and a few other miscellaneous management functions. The script for running each code was named for that code, e.g. LHS.py, and it imported a module named for the code but with "lib" appended, e.g. LHSlib.py. The "lib" module contains the definition of a class named for the code and some auxillary functions. The class defines methods (functions) for declaring the names and locations of the input and output files, for running various steps required to run the code, e.g. running GENMESH and MATSET, running POSTLHS and other pre-execution codes, running the code itself, running algebra and summarize steps, and checking in the output files. The main script (e.g. LHS.py) creates the code object instance and contains calls to these methods. Another Python script, Run.py, was used to execute one or more of the code scripts. Finally a set of Bourne shell scripts was used to set up the directories, including links to previous analyses that contained files being reused, and to call Run.py. For example, to run the CRA14BL analysis the script RunCRA14BL.sh was executed:

```
#!/bin/sh
# Codes =  "LHS","EPAUNI","BRAGFLO","PANEL","NUTS","DRSPALL",\
#               "CUTTINGS_S","BRAGFLO_DBR","SECOTP2D","CCDFGF"
#set up links to PABC09
mkdir /home/run_mast/GD/Analyses/CRA14BL
cd /home/run_mast/GD/Analyses/CRA14BL
ln -s ../PABC09/DRSPALL ./DRSPALL
ln -s ../PABC09/SECOTP2D ./SECOTP2D
cd /home/run_mast/GD
#now run the codes
./Run.py True CRA14BL -r=1-3 -rdb=PA_Results LHS
if ! ./Run.py True CRA14BL -r=1 EPAUNI
then
    exit
fi
#run BRAGFLO for all three reps to support CRA14BV
if ! ./Run.py True CRA14BL -r=1-3 -b=1 -rdb=PA_Results BRAGFLO
then
    exit
fi
./Run.py True CRA14BL -r=1 -b=1 -rdb=PA_Results PANEL NUTS \
           CUTTINGS_S BRAGFLO_DBR CCDFGF
```

The Run.py script was:

```
#!/usr/bin/python
import sys
import os
import string
from rc import *

#Usage: Run.py <True|False> Analysis -rdb ResultsDB|TEST [-r=Reps \
#        [-b=Brine_volumes [...]]] CodeName:Version [CodeName:Version ...]
#      or
```

```
#      Run.py <True|False> Analysis -rdb TEST [-r=Reps \
#         [-b=Brine_volumes [...]]]  ALL:Version
#e.g. Run.py True -r=1 -b=1 LHS:TEST PANEL:TEST
#
#The optional arguments (those denoted using []) must start with a -. These are passed
#to the script or scripts to be run. See rc.py for a listing of the default values.

#The first argument controls whether all previous results are deleted prior to
#starting a run. False would be passed to rerun only portions of a run
#where the output files to be recreated have been removed.

#Note that using "TEST" as the version will suppress logging retrievals in the
#    ParamDB  database. If the version is omitted or is not TEST then version
#    will be extracted from the metadata for the executable file.

def main():

    ranges={}
    arg_list(ranges)
    ANALYSIS_NAME=sys.argv[2]
    if ANALYSIS_NAME.strip()=="":       #ABORT! Bad path names will be constructed
            os.exit()                   #in the rm commands below

    Backup="/nfs/data/CVSLIB_Backup"
    ReposRoot="/nfs/data/CVSLIB/WIPP_ANALYSES/"+ANALYSIS_NAME+""
    run_all="True"

    #list the codes, their run order and the repositories they update
    Dependencies={
        "LHS":[1,["PRELHS","LHS"]],
        "EPAUNI":[2,["EPAUNI"]],
        "BRAGFLO":[3,["ALGEBRACDB","BRAGFLO","GENMESH","ICSET","MATSET",
                      "POSTLHS","PREBRAG","POSTBRAG"]],
        "PANEL":[4,["ALGEBRACDB","GENMESH","LHS","MATSET","PANEL","SUMMARIZE"]],
        "NUTS":[5,["ALGEBRACDB","MATSET","NUTS","SCREEN","SUMMARIZE"]],
        "DRSPALL":[6,[]],
        "CUTTINGS_S":[7,["GENMESH","MATSET","POSTLHS","CUTTINGS_S"]],
        "BRAGFLO_DBR":[8,["ALGEBRACDB","BRAGFLO","GENMESH","ICSET","MATSET",
                      "POSTLHS","PREBRAG","RELATE","POSTBRAG","SUMMARIZE"]],
        "SECOTP2D":[9,["ALGEBRACDB","PRESECOTP2D","RELATE","SUMMARIZE",\
                      "GENMESH","MATSET","SECOTP2D"]],
        "CCDFGF":[10,["PRECCDFGF","GENMESH","MATSET","CCDFGF"]]
    }
    #Get the first non-"-" argument
    code_start=2
    found_codes = False
    for a in sys.argv[3:]:
        code_start+=1
        if a[0]!="-":
            found_codes=True
            break

    if not found_codes:
        print "Too few arguments. You must list one or more codes, or 'ALL'"
        sys.exit(1)
    #get the RUNALL state
    run_all=sys.argv[1]
    if sys.argv[code_start]=="ALL":
        #dictionary keys may not be in the correct order, so list the right
        #order here
        codes=["LHS","EPAUNI","BRAGFLO","PANEL","NUTS","DRSPALL",\
               "CUTTINGS_S","BRAGFLO_DBR","SECOTP2D","CCDFGF"]
    else:
```

```
        codes=sys.argv[code_start:]
    for code_ver in codes:
        if code_ver.strip()=="":        #ABORT. Path names constructed below for rm
            print "ABORTING: Code name is missing!"
            os._exit(1)
        if code_ver.find(":")==-1:
            code = code_ver
            version = "GET"
        else:
            code, version = code_ver.split(":")
        #recover to previous stage?
        if False:       #turn this off for now
            stage=Dependencies[code][0]
            if os.path.exists(Backup+"/Stage"+str(stage)):      #recover prior state
                for rep in Dependencies[code][1]:
                    print "rm -rf "+ReposRoot+"/"+rep
                    os.system("rm -rf "+ReposRoot+"/"+rep)
                    print "cp -rp "+Backup+"/Stage"+str(stage)+"/"+\
                        rep+" "+ReposRoot
                    os.system("cp -rp "+Backup+"/Stage"+str(stage)+"/"+\
                        rep+" "+ReposRoot)
            else:                   #save prior state
                os.system("mkdir "+Backup+"/Stage"+str(stage))
                for rep in Dependencies[code][1]:
                    print "cp -rp "+ReposRoot+"/"+rep+" "+Backup+"/Stage"+str(stage)
                    os.system("cp -rp "+ReposRoot+"/"+rep+"
"+Backup+"/Stage"+str(stage))
        #now run the code
        #clean up the SGE log files
        os.system("rm -rf /home/run_mast/SGE_logs; mkdir /home/run_mast/SGE_logs")
        #Remove work area files
        if run_all:
            print "rm -rf /home/$USER/GD/Analyses/"+ANALYSIS_NAME+"/"+code+"/Run"+\
               code+"/*"
            os.system("rm -rf
/home/$USER/GD/Analyses/"+ANALYSIS_NAME+"/"+code+"/Run"+\
               code+"/*")
            print "rm -rf /home/$USER/GD/Analyses/"+ANALYSIS_NAME+"/"+code+\
                "/Run"+code+""
            os.system("rm -rf /home/$USER/GD/Analyses/"+ANALYSIS_NAME+"/"+code+\
                "/Run"+code+"")
            print "mkdir -p /home/$USER/GD/Analyses/"+ANALYSIS_NAME+"/"+code+\
                "/Run"+code+""
            os.system("mkdir -p /home/$USER/GD/Analyses/"+ANALYSIS_NAME+"/"+code+\
                "/Run"+code+"")
            #remove existing run control scripts
            print "rm -f
/home/$USER/GD/Analyses/"+ANALYSIS_NAME+"/"+code+"/"+code+".py"
            os.system("rm -f
/home/$USER/GD/Analyses/"+ANALYSIS_NAME+"/"+code+"/"+code+".py")
            print "rm -f /home/$USER/GD/Analyses/"+ANALYSIS_NAME+"/"+code+\
                "/"+code+"lib.py"
            os.system("rm -f /home/$USER/GD/Analyses/"+ANALYSIS_NAME+"/"+code+\
                "/"+code+"lib.py")
            print "cd /home/$USER/GD/Analyses/"+ANALYSIS_NAME+"/"+code+"; "+\
                "cvs -d $CVSLIB/WIPP_ANALYSES/"+ANALYSIS_NAME+"/"+code+\
                " export -r HEAD -d . RunControl/"+code+".py;"+\
                "cvs -d $CVSLIB/WIPP_ANALYSES/"+ANALYSIS_NAME+"/"+code+\
                " export -r HEAD -d . RunControl/"+code+"lib.py;"+\
                "cp $CVSLIB/bin/rc.py ."
            os.system("cd /home/$USER/GD/Analyses/"+ANALYSIS_NAME+"/"+code+"; "+\
                "cvs -d $CVSLIB/WIPP_ANALYSES/"+ANALYSIS_NAME+"/"+code+\
                " export -r HEAD -d . RunControl/"+code+".py;"+\
```

```
                "cvs -d $CVSLIB/WIPP_ANALYSES/"+ANALYSIS_NAME+"/"+code+\
                " export -r HEAD -d . RunControl/"+code+"lib.py;"+\
                "cp $CVSLIB/bin/rc.py .")

            print '$CVSLIB/bin/RunProgress '+code+' '+ANALYSIS_NAME+' '+\
                 '0 0:0:0 "Startup" /home/$USER/GD/Analyses/'+ANALYSIS_NAME+'/'+code+\
                 '/Run'+code+'/'+code+'.log  "`date '+"'"+'+%y:%m:%d
%H:%M:%S'+"'"+'`"'
            os.system('$CVSLIB/bin/RunProgress '+code+' '+ANALYSIS_NAME+' '+\
                 '0 0:0:0 "Startup" /home/$USER/GD/Analyses/'+ANALYSIS_NAME+'/'+code+\
                 '/Run'+code+'/'+code+'.log  "`date '+"'"+'+%y:%m:%d
%H:%M:%S'+"'"+'`"')

        cmd = "#!/bin/sh\n"
        if code=="LHS":
            cmd = cmd +"export CompCodeName=PRELHS; export RunCodeName=LHS; "+\
                'export CompCodeVersion="DUMVERS" \n'
        else:
            cmd = cmd +"export CompCodeName="+code+"; export RunCodeName="+code+"; "+\
                'export CompCodeVersion="DUMVERS" \n'

        cmd = cmd + "cd /home/$USER/GD/Analyses/"+ANALYSIS_NAME+"/"+code+"/Run"+\
              code+"\n "+ "../"+code+".py "+\
              "-VER="+version+" "+\
              "-CVSROOT=$CVSLIB/WIPP_ANALYSES/"+ANALYSIS_NAME+"/"+code+" "+\
              "-WD=/home/$USER/GD/Analyses/"+ANALYSIS_NAME+"/"+code+"/Run"+code+" "+\
              "-AT="+ANALYSIS_NAME+" "+\
              "-RUNALL="+run_all+" "+\
              "-DEFER=True "+\
              "-CSH=/home/$USER/GD/Run.py "
        for a in sys.argv:
            if a[0]=="-":
              cmd=cmd+" "+a
        cmd=cmd+"\n"
        print cmd
        scrpt = open("RunScript.csh","w")
        scrpt.write(cmd)
        scrpt.close
        os.chmod("RunScript.csh",0700)

        rval=exec_cmd(cmd)

        if rval<>0:
            print code," failed. Execution of Run.py terminated."
            exec_cmd('$CVSLIB/bin/RunProgress '+code+' '+ANALYSIS_NAME+' - -
"Failed"')
            os._exit(1)
main()
```

An example of a code script is that for CUTTINGS_S:

```
#!/opt/epd-7.2-2/bin/python
#Script to run GENMESH, MATSET, POSTLHS, and CUTTINGS_S
#Command: CUTTINGS_S.py Analysis [-r=<reps>] [-s=<scens>] [-v=<vectors>]
#         [-V=True|False] [-UT=<update tag>] [-AT=<analysis tag>]
#         [-WD=<working directory>]
#         [-OR=<output repository>] [-CVSROOT=<default repository>]
#    Where Analysis  = the analysis name, used as the analysis tag and
#                      repository name
#          -r        = the range (e.g. 1-3) or list (e.g. 1,2) of
#                      replicates. Default=1-3
```

```
#           -s        = the range (e.g. 1-6) or list (e.g. 1,2,6) of
#                       scenarios. Default=1-6
#           -v        = the range (e.g. 1-100) or list (e.g. 1,2,6) of
#                       vectors. Default=1-100
#           -V        = False to suppress verbose output. Default = True
#           -AC       = True to allow commit of files to repository.
#                        Default = False
#           -UT       = Update tag. Default=""
#           -AT       = Analysis tag. Default=""
#           -WD       = Working directory. Default="."
#           -OR       = Output repository tag. Default=""
#           -CVSROOT  = CVSROOT, the default repository. Default=""
#           -FSL      = log for file names used or created
#           -CPATH    = path to repositories for WIPP PA codes
#           -P        = platorm name, e.g. Solaris
#Example: BF1.py PABC09 -r=1-3 -s=1 -v=1,28,29 -AC=True -AT=PABC09 \
#              -UT=PABC09 -CVSROOT=$CVSLIB/WIPP_ANALYSES/PABC09 \
#              -OR=$CVSLIB/WIPP_ANALYSES/PABC09 -WD=. -V=True > & BF1.log

from CUTTINGS_Slib import *

#Create a CUTTINGS_S object. Alternative input and code libraries could be
#passed using dictionaries that give the alternative repository paths, e.g.
#  AlternativeLibraries={"ALGEBRACDB":"/nfs/data/CVSLIB/WIPP_ANALYSES/CRA1"}
#  AlternativeCodeLibraries={"ALGEBRACDB":"/home/tbkirch/CODE/ALGEBRACDB"}
#  CUSP=CUTTINGS_S(AltLibs=AlternativeLibraries, \
#       AltCodeLibs=AlternativeCodeLibraries)

AlternativeLibraries={"DRSPALL":"/nfs/data/CVSLIB/WIPP_ANALYSES/PABC09"}
CUSP = CUTTINGS_S(AltLibs=AlternativeLibraries)

#Get the code executable files
#Alternative code versions can be specified by passing a dictionary to
#GetCodes, e.g.
#  AltVersions={"cuttings_s":"BRAGFLO_603"}
#  CUSP.GetCodes(AltVersions)

#Get the codes
CUSP.GetCodes()

#Get the input files
CUSP.GetInputs()

CUSP.GenmeshMatset()
CUSP.POSTLHS()
CUSP.CUTTINGS_S()
CUSP.CheckIn()
CUSP.CloseFiles()
```

The Python class CUTTINGS_S referenced in the CUTTINGS_S.py script is defined in the CUTTINGS_Slib.py script:

```
#!/opt/epd-7.2-2/bin/python

from rc import *
from string import Template
import glob
import time
import stat

ANALYSIS_NAME=""
WORKING_DIRECTORY=""
```

```
TDIR = "/working/"

#=============================================================================
class CUTTINGS_S:
    def __init__(self,AltLibs={},AltCodeLibs={}):
        global ANALYSIS_NAME
        global WORKING_DIRECTORY

        init_print()

        #get the ranges for the loops by replicate, scenario and vector
        #    and other control settings
        self.ranges={}
        arg_list(self.ranges)

        ANALYSIS_NAME=self.ranges["-AT"]
        self.AR = self.ranges["-R"]
        self.CV = self.ranges["-VER"]
        self.DB=self.ranges["-db"]      #Parameter database
        self.RDB=self.ranges["-rdb"]    #Results database

        self.TMP = TEMP #change to CREATE to preserve all temporary files

        #Set globals for module rc
        set_verbose(self.ranges["-V"])
        set_update_tag(self.ranges["-UT"])
        set_allow_commit(self.ranges["-AC"])
        set_cvslib("/nfs/data/CVSLIB")
        set_fetch_tag("")
        set_working_directory(self.ranges["-WD"])
        set_cvs_output_repos(self.ranges["-OR"])
        set_analysis_tag(self.ranges["-AT"])
        CVSROOT = set_cvsroot(self.ranges["-CVSROOT"])
        set_fsl(self.ranges["-FSL"])
        WORKING_DIRECTORY=self.ranges["-WD"]
        self.code_path=self.ranges["-CPATH"]
        self.platform=self.ranges["-P"]

        init_vars()

        #list the date
        print_comment("DATE:"+time.ctime())
        LogTime("CUTTINGS_S",ANALYSIS_NAME,0,"at Start","Running")

        #define the kinds of error that will stop execution
        stop_on(["ERROR","COPY_CONFLICT"])

        r_path,d=os.path.split(CVSROOT) #get the path to the repositories
        self.libs=["GENMESH","MATSET","CUTTINGS_S","BRAGFLO","POSTLHS",\
                   "DRSPALL"]
        self.lib=libPath(r_path,self.libs,AltLibs) #build list of paths to
                                                   #libraries
        self.codes=["GENMESH","MATSET","POSTLHS","CUTTINGS_S"]
        self.clib = libPath(self.code_path,self.codes,AltCodeLibs)

        #Create an export collection as an instance of InputFiles
        self.Exports=InputFiles()

        #define the output files
        self.Outputs=OutputFiles()

        #Create a directories for management of the slave scripts
        mkdir("Scripts")
```

```
        return

    def GetCodes(self,VersionDict={}):
        #Define the codes to be exported
        global ANALYSIS_NAME

        #Files are exported to the default working directory
        Codes=InputFiles()

        #Export the executables
        for c in self.codes:
            if c.lower() in VersionDict.keys():
                v=VersionDict[c.lower()]
            else:
                v=""
            Codes.Add("Build/"+self.platform+"/"+c.lower(),\
                repos=self.clib[c], ver=v, executable=True)

        #Check for existence of all code files before getting the files
        #Verify() MUST BE CALLED for everything to work properly
        if (not Codes.Verify()):
            print_err("Missing code file: Run aborted.")
            abort()

        #Now export the files
        LogTime("CUTTINGS_S",ANALYSIS_NAME,1,"Exporting codes",\
                "Exporting codes")
        Codes.Export()

        print_comment("Get the metadata for the executables:")
        for c in self.codes:
            log_id(c.lower())

        return

    def GetInputs(self):
        #Define the input files to be exported
        #Files are exported to ./Inputs or the default working directory
        global ANALYSIS_NAME

        InFiles={}      #dictionary of input files by code
        InFiles["GENMESH"]=["gm_cusp_${A}.inp"]
        InFiles["MATSET"]=["ms_cusp_${A}.inp"]
        InFiles["CUTTINGS_S"] = ["cusp_${A}.inp"]

        #Export the input files into a directory
        for c in self.libs[:-3]:
            files=InFiles[c]
            for f in files:
                self.Exports.Add("Input/"+f, repos=self.lib[c],toDir="./Inputs")

        cwd = os.getcwd()

        #Add the DRSPALL tables
        for r in self.ranges["-r"]:
            f="Output/mspall_drs_${A}_r"+str(r)+".out"
            self.Exports.Add(f, repos=self.lib["DRSPALL"], toDir="./Inputs", \
                get_from=cwd+"/../../DRSPALL/RunDRSPALL")

        #Add the Bragflo files
        first=True
        nFilesrsv = len(self.ranges["-r"])*5*len(self.ranges["-v"])-1
```

```
        simListrsv =  "    ... and "+str(nFilesrsv)+" similar files for " +\
                    str(len(self.ranges["-r"]))+" replicates, 5 scenarios and "\
                    +str(len(self.ranges["-v"]))+ " vectors"
        for r in self.ranges["-r"]:
            for s in range(1,6):
                 for v in range(1,101):      #self.ranges["-v"]:
                    arsv="${A}_r"+str(r)+"_s"+str(s)+"_v"+ \
                        ("000"+str(v))[-3:]
                    f = "Output/bf3_"+arsv+".cdb"
                    self.Exports.Add(f, repos=self.lib["BRAGFLO"], \
                        toDir="./Inputs", get_from=cwd+\
                        "/../../BRAGFLO/RunBRAGFLO",log=first,logmsg=simListrsv)
                    if first:
                        first=False
                        simlistrsv=""

        #Check for existence of all input files before getting the files
        #Verify() MUST BE CALLED for everything to work properly
        if (not self.Exports.Verify()):
            print_err("Missing input file: Run aborted.")
            abort()

        #Now export the files
        LogTime("CUTTINGS_S",ANALYSIS_NAME,1,"Exporting files","Exporting files")
        self.Exports.Export()

        return

    def GenmeshMatset(self):
        #run Genmesh and MATSET
        global ANALYSIS_NAME

        LogTime("CUTTINGS_S",ANALYSIS_NAME,1,"Genmesh and Matset",\
                "Genmesh and Matset")

        f="Output/gm_cusp_"+ANALYSIS_NAME+".cdb"
        Cg = self.Outputs.Add(f, action=CHECKIN,repos=self.lib["GENMESH"],\
           comment= "CDB transfer file")
        f="Output/ms_cusp_"+ANALYSIS_NAME+".cdb"
        Cm = self.Outputs.Add(f, repos=self.lib["MATSET"], action=CHECKIN,\
            comment= "CDB transfer file")
        f="Output/gm_cusp_"+ANALYSIS_NAME+".xdbg"
        Og = self.Outputs.Add(f, action=self.TMP,repos=self.lib["GENMESH"],\
            comment= "Debug file (not saved)")
        f="Output/ms_cusp_"+ANALYSIS_NAME+".xdbg"
        Om = self.Outputs.Add(f, action=self.TMP,repos=self.lib["MATSET"],\
             comment= "Debug file (not saved)")

        #Run genmesh and matset. The command line file names are generated using
        #    the analysis tag
        usr = self.Exports.FName("Input/gm_cusp_${A}.inp")
        cmd="./genmesh -user "+usr+" -output "+ \
            Cg.createPath+" -debug "+Og.createPath
        rc = exec_cmd(cmd)
        Og.ClearTemp(rc)

        usr = self.Exports.FName("Input/ms_cusp_${A}.inp")
        cmd="./matset -input GENMESH/Output/gm_cusp_"+ANALYSIS_NAME+".cdb"
        cmd=cmd+" -user "+usr+" -output "+ \
            Cm.createPath+" -debug "+Om.createPath
        cmd=cmd+" -analysis "+ANALYSIS_NAME+" -revision "+self.AR+ \
            " -code CUTTINGS_S -version "+self.CV
        rc = exec_cmd(cmd,of=Cm.createPath)
```

```
        Om.ClearTemp(rc)
        return

    def POSTLHS(self):
        #Run postlhs
        global ANALYSIS_NAME

        RUN_ALL = run_all()

        LogTime("CUTTINGS_S",ANALYSIS_NAME,25,"Starting PostLHS",\
                "Running PostLHS")
        #Define outputs
        first=True
        for rep in self.ranges["-r"]:
            fn="Output/lhs3_cusp_"+ANALYSIS_NAME+"_r"+str(rep)+".dbg"
            self.Outputs.Add(fn,action=self.TMP,repos=self.lib["POSTLHS"],\
                log=first, logmsg="")
            for v in self.ranges["-v"]:
                vstr = ("000"+str(v))[-3:]
                fn="Output/lhs3_cusp_"+ANALYSIS_NAME+"_r"+str(rep)+"_v"+vstr+\
                   ".cdb"
                self.Outputs.Add(fn,action=CHECKIN,repos=self.lib["POSTLHS"],\
                    log=first, logmsg="")
            first=False

        #Run the postlhs jobs.
        first=True
        nFilesr = len(self.ranges["-r"])-1
        for r in self.ranges["-r"]:
            scrpt = write_postlhs_script(r,first, self.AR, self.RDB)
            if first:
               print "    ... and ",nFilesr, \
                     " similar scripts created and queued."
               first=False
            queue_command("./"+scrpt)

        wait_until_gone("*.sh","CUTTINGS_S",ANALYSIS_NAME,2,3,"PostLHS")
        return

    def CUTTINGS_S(self):
        global ANALYSIS_NAME

        RUN_ALL = run_all()

        LogTime("CUTTINGS_S",ANALYSIS_NAME,27,"Starting Cuttings_s",\
                "Running Cuttings_s")

        num = len(self.ranges["-r"])*len(self.ranges["-v"])
        numsvt = len(self.ranges["-r"])*len(self.ranges["-v"]*3*26)
        nr=len(self.ranges["-r"])
        n=0
        n2=0
        first=True
        firstsvt=True
        simListrv =  "    ... and declare "+str(num-1)+" similar files for " + \
                    str(len(self.ranges["-r"]))+" replicates and "+ \
                    str(len(self.ranges["-v"]))+ " vectors"
        simListrvst =  "    ... and declare "+str(numsvt-1)+\
                    " similar files for "+ \
                    str(len(self.ranges["-r"]))+" replicates,"+ \
                    str(len(self.ranges["-v"]))+ " vectors," \
                    "3 locations and 26 times"
        self.times=[[100,350,1000,3000,5000,10000],
```

```
        [550,750,2000,4000,10000],
        [1200,1400,3000,5000,10000],
        [550,750,2000,4000,10000],
        [1200,1400,3000,5000,10000]]

#CUTTINGS_S files
for rep in self.ranges["-r"]:
    self.Outputs.Add("Output/cusp_"+ANALYSIS_NAME+"_r"+str(rep)+".tbl",\
       action=CHECKIN,repos=self.lib["CUTTINGS_S"],log=first,logmsg="")
    n+=1
    for v in self.ranges["-v"]:
       for s in range(1,6):
         for loc in ["U","M","L"]:
           for t in self.times[s-1]:
               self.Outputs.Add("Output/cusp_"+ANALYSIS_NAME+"_r"+\
                   str(rep)+"_s"+str(s)+"_t"+("00000"+ str(t))[-5:]+\
                   "_"+loc+"_v"+("000"+ str(v))[-3:]+".cdb", \
                   action=CHECKIN, repos=self.lib["CUTTINGS_S"],\
                   log=firstsvt,logmsg=simListrvst)
               if firstsvt:
                   firstsvt=False
                   simListrvst=""
               n+=1

       if first:
           first=False
           simListrv=""
       prog = 1+float(n)/float(numsvt+num*3)*25
       LogTime("CUTTINGS_S",ANALYSIS_NAME,prog,"after Outputs defined", \
           str(n)+" of "+str(nr*3+numsvt) + \
           " Outputs are defined",False)

#Create the input file for CUTTINGS_S, which lists the I/O files
for r in self.ranges["-r"]:
    bf3_anal=AnalysisName(self.lib["BRAGFLO"])
    msp_anal=AnalysisName(self.lib["DRSPALL"])
    cusp_anal=AnalysisName(self.lib["CUTTINGS_S"])
    f = write_cusp_input(r,bf3_anal,msp_anal,cusp_anal)
    self.Outputs.Add("Output/"+f, action=CHECKIN,repos=\
        self.lib["CUTTINGS_S"])

#Create the status log for the cuttings_s runs. This file will list the
#success or failure of the cuttings_s slave scripts.
if os.path.exists("./status.log"):  #remove status log for cuttings_s
    os.remove("./status.log")
nJobs=len(self.ranges["-r"])*5*len(self.ranges["-v"])*3
status = open("status.log","w")
status.write(str(nJobs)+" cases to be run\n")
status.close()

#Run the Cuttings_s jobs.
first = True
nFilesr = len(self.ranges["-r"])
for r in self.ranges["-r"]:
    fn = "./CUTTINGS_S/Output/cusp_"+ANALYSIS_NAME+"_r"+str(r)+".tbl"
    if RUN_ALL or (not RUN_ALL and not os.path.exists(fn)):
        scrpt = write_cuttings_script(r,first)
        args=" ./Inputs/bf1_"+ANALYSIS_NAME+"_s"+str(s)+".inp"
        if first:
           print "    ... and ",nFilesr-1, \
                 " similar scripts created and queued."
           first=False
        queue_command("./"+scrpt+args)
```

```
        wait_until_gone("*.sh","CUTTINGS_S",ANALYSIS_NAME,27,52,"CUTTINGS_S")
        return

    def CheckIn(self):
        #Checkin all output files
        global ANALYSIS_NAME

        #checkin the output files
        stop_on(["COPY_CONFLICT"])
        self.Outputs.Checkin("CUTTINGS_S",ANALYSIS_NAME,52,100)
        LogTime("CUTTINGS_S",ANALYSIS_NAME,52,"Check-in", "Check-in outputs")
        return

    def CloseFiles(self):
        #Close the log files

        print_comment("DATE:"+time.ctime())
        LogTime("CUTTINGS_S",ANALYSIS_NAME,100,"Check In","Completed")
        close_all()
        return

def write_postlhs_script(r,prnt,AR,RDBase):     #write the slave script
    global ANALYSIS_NAME
    global WORKING_DIRECTORY
    global TDIR

    #define the slave script
    scr="#$$ -S /bin/sh\n\
cd ${WORK}\n\
if ${WORK}/postlhs \\\n\
    -input ${WORK}/MATSET/Output/ms_cusp_${ANAL}.cdb \\\n\
    -output ${WORK}/POSTLHS/Output/lhs3_cusp_${ANAL}_r${rep}_v001.cdb \\\n\
    -debug ${TMP}lhs3_cusp_${ANAL}_r${rep}.dbg \\\n\
    -analysis ${ANAL} \\\n\
    -revision ${REV} \\\n\
    -replicate ${rep} \\\n\
    -database ${RDB} \n\
then \n\
    mv  ${WORK}/postlhs_r${rep}.sh \\\n        ${WORK}/Scripts\n\
    rm ${TMP}lhs3_cusp_${ANAL}_r${rep}.dbg \n\
else \n\
    cp ${TMP}lhs3_cusp_${ANAL}_r${rep}.dbg
${WORK}/POSTLHS/Output/lhs3_cusp_${ANAL}_r${rep}.dbg \n\
    rm ${TMP}lhs3_cusp_${ANAL}_r${rep}.dbg \n\
fi \n"
    s=Template(scr)
    wrk=WORKING_DIRECTORY
    txt = s.substitute(WORK=wrk, ANAL=ANALYSIS_NAME, rep=str(r), REV=AR, \
          RDB=RDBase, TMP=TDIR)
    #write the slave script
    sname="postlhs_r"+str(r)+".sh"
    scrpt=open(sname,"w")
    scrpt.write(txt)
    scrpt.close()
    os.chmod(sname,0700)
    if prnt:
        print "================================================="
        print "Script queued: ",sname
        print "================================================="
        txt="~    "+txt.replace("\n","\n~    ") #add ~ format char to start of all
lines
        print txt
```

```
        print "======================================================"
    return sname

def write_cusp_input(r,bf3_anal,msp_anal,cusp_anal):
    global ANALYSIS_NAME
    fname="cusp_"+ANALYSIS_NAME+"_master_r"+str(r)+".inp"
    inp=open("./CUTTINGS_S/Output/"+fname,"w")
    txt="5 100 3\n\
 1 6 100 350 1000 3000 5000 10000\n\
 2 5 550 750 2000 4000 10000\n\
 4 5 550 750 2000 4000 10000\n\
 3 5 1200 1400 3000 5000 10000\n\
 5 5 1200 1400 3000 5000 10000\n\
L\n\
M\n\
U\n\
./Inputs/cusp_"+cusp_anal+".inp\n\
./POSTLHS/Output/lhs3_cusp_${ANAL}_r${rep}_v^.cdb\n\
./Inputs/bf3_"+bf3_anal+"_r${rep}_s%_v^.cdb\n\
./Inputs/mspall_drs_"+msp_anal+"_r${rep}.out\n\
./CUTTINGS_S/Output/cusp_${ANAL}_r${rep}.tbl\n\
./CUTTINGS_S/Output/cusp_${ANAL}_r${rep}_s%_t!!!!!_(_v^.cdb\n\
!-------------------------------------------------\n\
! CUTTINGS_S Master Control File                  \n\
!-------------------------------------------------\n\
!\n\
"
    T=Template(txt)
    wrk=WORKING_DIRECTORY
    txt = T.substitute(WORK=wrk, ANAL=ANALYSIS_NAME, rep=str(r))
    inp.write(txt)
    inp.close
    return fname

def write_cuttings_script(r,prnt):      #write the slave script
    global ANALYSIS_NAME
    global WORKING_DIRECTORY
    global TDIR

    scr="#$$ -S /bin/sh\n\
cd ${WORK}\n\
if ./cuttings_s -user ./CUTTINGS_S/Output/cusp_${ANAL}_master_r${rep}.inp \\\n\
     -debug ${TMP}cusp_${ANAL}_r${rep}.xdbg \n\
then \n\
    echo CUTTINGS_S succeeded \n\
    mv script_r${rep}.sh ./Scripts \n\
else\n\
    return_code=$$? \n\
    echo $$return_code ${rep} script_r${rep}.sh $$1 cuttings_s >> status.log \n\
    echo CUTTINGS Failed. \n\
    cp ${TMP}cusp_${ANAL}_r${rep}.xdbg ./CUTTINGS_S/Output/cusp_${ANAL}_r${rep}.xdbg
\n\
    rm ${TMP}cusp_${ANAL}_r${rep}.xdbg \n\
fi\n"

    T=Template(scr)
    wrk=WORKING_DIRECTORY
    txt = T.substitute(WORK=wrk, ANAL=ANALYSIS_NAME, rep=str(r), TMP=TDIR)
    #write the slave script
    sname="script_r"+str(r)+".sh"
    scrpt=open(sname,"w")
    scrpt.write(txt)
    scrpt.close()
```

```
    os.chmod(sname,0700)
    if prnt:
        print "================================================="
        print "Script queued: ",sname
        print "================================================="
        txt="~    "+txt.replace("\n","\n~    ") #add ~ format char to start of all
lines
        print txt
        print "================================================="
    return sname
```

Code scripts and their associates "lib" scripts are stored in the RunControl module of the CVS repository for the code under the analysis where it was used, e.g. $CVSLIB/WIPP_ANALYSES/CRA14BL/CUTTING_S.

The input files and pertinent output files used or created during the execution of the PABC09 and CRA14 analyses on Solaris are listed below. These tables were created by the scripts with one exception. After completing the analyses a decision was made to no longer store the CDB files in the repository because 1) in most cases the time it takes to store or recover the files is many times longer than the time it takes to create them and 2) the files will be preserved in the original locations until deemed no longer necessary.

Run Control Tables

The PABC09 and CRA14 analyses produced approximately half a million output files excluding transfer files that are produced then discarded. Previously, on VMS, the CDB (Camdat database files) were preserved in the repository, since many were re-used in subsequent analyses. On Solaris, which has greater computation speeds, the CDB files take longer to check into and out of the CVS repositories than is takes to create them, so these files will no longer be stored in the repository but will be preserved in the location where they were created.

The CVS repositories for an analyses are placed in a directory named for the analysis under the directory $CVSLIB/WIPP_ANALYSES, e.g. $CVSLIB/WIPP_ANALYSES/PABC09. Each code use in an analysis will have a repository in that analysis directory and the repository will store the input files, output files and any scripts needed to run the code. The strategy for organizing files is somewhat different that that employed on VMS. On Solaris the input files for a utility code, such as SUMMARIZE, will be found within the Input module of the code repository and these input files are those needed for all the models runs where the code is employed. Likewise, all output files from the utility will be placed on the Output module even though they were generated when running different models. Under VMS such input and output files were generally stored in a library associated with the model that was run and where they were used or created.

The directory structure where codes are run has as its head the directory /home/run_mast/GD/Analyses. Each analysis has a directory under this head, and each model used in the analyses has a directory within the analysis directory, e.g. /home/run_mast/Analyses/PABC09/BRAGFLO. The scripts for that code are placed in the model directory but the code execution takes place in a subdirectory named for the model with "Run" prepended, e.g. RunBRAGFLO. Input files are placed within the Inputs directory of the

"Run" directory. The directory structure for the analyses run under QA is dictated to some degree by the requirements that files being checked into CVS must be placed in "sandbox" directories. Sandbox directories are checked out of CVS and contain a special CVS directory that contains the information used to track and to versions of files. The sandbox directories for each code were created within the "Run" directory where the execution of the code took place. Thus the output files for BRAGFLO are placed in /home/run_mast/Analyses/PABC09/BRAGFLO/RunBRAGFLO/BRAGFLO/Output, while those from the SUMMARIZE code for the BRAGFLO results are placed in /home/run_mast/Analyses/PABC09/BRAGFLO/RunBRAGFLO/SUMMARIZE/Output. Although a SUMMARIZE sandbox may exist under more than one code it represents a single repository, so all output files for SUMMARIZE or other intermediate codes are placed in the Output module for that code. In the tables below the $PATH symbol is used in the Path column to indicate the path of the sandbox directory where output files are written and the $REP symbol is used to represent the path to the CVS repository for those files that are stored in a repository. The path component in the File column represents the subdirectory of the sandbox where files not checked into the repository are stored, or the module where files are stored under CVS.

Analysis PABC09

BRAGFLO

Table 3. The run script files used were:

File	Path	Comment
RunControl/BRAGFLO.py	$REP/BRAGFLO	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/BRAGFLO	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 4. The input files used were:

File	Path	Comment
Input/alg1_bf_PABC09.inp	$REP/ALGEBRACDB	Input file
Input/alg2_bf_PABC09.inp	$REP/ALGEBRACDB	Input file
Input/bf1_PABC09_s*n*.inp	$REP/PREBRAG	Input file
Input/bf1_PABC09_s*n*_mod1.inp	$REP/PREBRAG	Input file
Input/bf1_PABC09_s*n*_mod2.inp	$REP/PREBRAG	Input file
Input/bf2_PABC09_closure.dat	$REP/BRAGFLO	Input file
Input/gm_bf_PABC09.inp	$REP/GENMESH	Input file
Input/ic_bf_PABC09.inp	$REP/ICSET	Input file
Input/ms_bf_PABC09.inp	$REP/MATSET	Input file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

n is 1-6

Table 5.The CVS repositories used were:

CVS Repositories
ALGEBRACDB
BRAGFLO
GENMESH
ICSET
MATSET
POSTBRAG
POSTLHS
PREBRAG

Table 6. The log files used were:

File	Path	Comment
RunControl/BRAGFLO.log	$REP/BRAGFLO	log file
RunControl/BRAGFLO.rtf	$REP/BRAGFLO	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 7.The output files produced were:

File	Path	Comment
Output/alg1_bf_PABC09_r*i*_v*vvv*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg2_bf_PABC09_r*i*_s*n*_v*vvv*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/bf2_PABC09_r*i*_s*n*_v*vvv*.inp	$REP/PREBRAG	BRAGFLO input file
Output/bf2_PABC09_r*i*_s*n*_v*vvv*.log	$REP/BRAGFLO	Logfile
Output/bf2_PABC09_r*i*_s*n*_v*vvv*.sum	$REP/BRAGFLO	Summary file
Output/bf3_PABC09_r*i*_s*n*_v*vvv*.cdb	$PATH/BRAGFLO	CDB transfer file
Output/gm_bf_PABC09.cdb	$PATH/GENMESH	CDB transfer file
Output/ic_bf_PABC09_r*i*_v*vvv*.cdb	$PATH/ICSET	CDB transfer file
Output/lhs3_bf_PABC09_r*i*_v*vvv*.cdb	$PATH/POSTLHS	CDB transfer file
Output/ms_bf_PABC09.cdb	$PATH/MATSET	CDB transfer file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09
$PATH = /home/run_mast/GD/Analyses/PABC09/BRAGFLO/RunBRAGFLO
i is 1-3
n is 1-6
vvv is 001-100

Table 8. The executable files used were:

File	Path	Comment
Build/Solaris/algebracdb (Ver:2.36)	$REP/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Solaris/bragflo (Ver:6.03)	$REP/BRAGFLO	Computes brine and gas flow in the repository
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/icset (Ver:2.23)	$REP/ICSET	Assigns initial conditions to the CAMDAT grid elements
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks

File	Path	Comment
Build/Solaris/postbrag (Ver:4.02)	$REP/POSTBRAG	Post-processes data for bragflo
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/prebrag (Ver:8.03)	$REP/PREBRAG	Pre-processes data for bragflo

Where:

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

BRAGFLO_DBR

Table 9 .The run script files used were:

File	Path	Comment
RunControl/BRAGFLO_DBR.py	$REP/BRAGFLO_DBR	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/BRAGFLO_DBR	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 10 .The input files used were:

File	Path	Comment
Input/alg1_dbr_PABC09.inp	$REP/ALGEBRACDB	
Input/alg2_dbr_PABC09_*sn*.inp	$REP/ALGEBRACDB	
Input/alg3_dbr_PABC09.inp	$REP/ALGEBRACDB	
Input/bf1_dbr_PABC09_L.inp	$REP/PREBRAG	
Input/bf1_dbr_PABC09_M.inp	$REP/PREBRAG	
Input/bf1_dbr_PABC09_U.inp	$REP/PREBRAG	
Output/bf3_PABC09_r*i*_s*m*_v*vvv*.cdb	$PATH/BRAGFLO/RunBRAGFLO/BRAGFLO	
Output/cusp_PABC09_r*i*_s*n*_t*tttt*_L_v*vvv*.cdb	$PATH/CUTTINGS_S/RunCUTTINGS_S/CUTTINGS_S	
Output/cusp_PABC09_r*i*_s*n*_t*tttt*_M_v*vvv*.cdb	$PATH/CUTTINGS_S/RunCUTTINGS_S/CUTTINGS_S	
Output/cusp_PABC09_r*i*_s*n*_t*tttt*_U_v*vvv*.cdb	$PATH/CUTTINGS_S/RunCUTTINGS_S/CUTTINGS_S	
Input/gm_dbr_PABC09.inp	$REP/GENMESH	
Input/ic_dbr_PABC09_*sn*.inp	$REP/ICSET	
Input/ms_dbr_PABC09.inp	$REP/MATSET	
Input/rel1_dbr_PABC09.inp	$REP/RELATE	
Input/rel2_dbr_PABC09_*sn*.inp	$REP/RELATE	
Input/sum_dbr.inp	$REP/SUMMARIZE	

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09
$PATH = /home/run_mast/GD/Analyses/PABC09/
i is 1-3
n is 1-5
m is 1-6
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

Table 11 .The CVS repositories used were:

CVS Repositories
ALGEBRACDB
BRAGFLO
BRAGFLO_DBR
CUTTINGS_S
GENMESH
ICSET
MATSET
POSTBRAG
POSTLHS
PREBRAG
RELATE
SUMMARIZE

Table 12 .The log files used were:

File	Path	Comment
RunControl/BRAGFLO_DBR.log	$REP/BRAGFLO_DBR	log file
RunControl/BRAGFLO_DBR.rtf	$REP/BRAGFLO_DBR	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09/RunBRAGFLO_DBR

Table 13 .The output files produced were:

File	Path	Comment
Output/alg1_dbr_PABC09_r*j*_s*o*_t*tttt*_v*vvv*.cdb	$PATH/ALGEBRACDB	
Output/alg2_dbr_PABC09_r*j*_s*o*_t*tttt*_v*vvv*.cdb	$PATH/ALGEBRACDB	
Output/alg3_dbr_PABC09_r*j*_s*o*_t*tttt*_L_v*vvv*.cdb	$PATH/ALGEBRACDB	
Output/alg3_dbr_PABC09_r*j*_s*o*_t*tttt*_M_v*vvv*.cdb	$PATH/ALGEBRACDB	
Output/alg3_dbr_PABC09_r*j*_s*o*_t*tttt*_U_v*vvv*.cdb	$PATH/ALGEBRACDB	
Output/bf2_dbr_PABC09_r*j*_s*o*_t*tttt*_L_v*vvv*.inp	$REP/BRAGFLO_DBR	
Output/bf2_dbr_PABC09_r*j*_s*o*_t*tttt*_M_v*vvv*.inp	$REP/BRAGFLO_DBR	
Output/bf2_dbr_PABC09_r*j*_s*o*_t*tttt*_U_v*vvv*.inp	$REP/BRAGFLO_DBR	
Output/bf3_dbr_PABC09_r*j*_s*o*_t*tttt*_L_v*vvv*.cdb	$PATH/BRAGFLO_DBR	
Output/bf3_dbr_PABC09_r*j*_s*o*_t*tttt*_M_v*vvv*.cdb	$PATH/BRAGFLO_DBR	
Output/bf3_dbr_PABC09_r*j*_s*o*_t*tttt*_U_v*vvv*.cdb	$PATH/BRAGFLO_DBR	
Output/gm_dbr_PABC09.cdb	$PATH/GENMESH	
Output/ic_dbr_PABC09_r*j*_s*o*_t*tttt*_v*vvv*.cdb	$PATH/ICSET	
Output/ms_dbr_PABC09.cdb	$PATH/MATSET	
Output/rel1_dbr_PABC09_r*j*_s*o*_t*tttt*_v*vvv*.cdb	$PATH/RELATE	
Output/rel2_dbr_PABC09_r*j*_s*o*_t*tttt*_v*vvv*.cdb	$PATH/RELATE	
Output/sum_dbr_PABC09_r*j*_s*o*_t*tttt*_L.tbl	$REP/SUMMARIZE	
Output/sum_dbr_PABC09_r*j*_s*o*_t*tttt*_M.tbl	$REP/SUMMARIZE	
Output/sum_dbr_PABC09_r*j*_s*o*_t*tttt*_U.tbl	$REP/SUMMARIZE	

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09
$PATH = /home/run_mast/GD/Analyses/PABC09/BRAGFLO_DBR/RunBRAGFLO_DBR
j is 1-3
o is 1-5
tttt 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

Table 14 .The executable files used were:

File	Path	Comment
Build/Solaris/algebracdb (Ver:2.36)	$REP/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Solaris/bragflo (Ver:6.03)	$REP/BRAGFLO	Computes brine and gas flow in the repository
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/icset (Ver:2.23)	$REP/ICSET	Assigns initial conditions to the CAMDAT grid elements
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postbrag (Ver:4.02)	$REP/POSTBRAG	Post-processes data for bragflo
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/prebrag (Ver:8.03)	$REP/PREBRAG	Pre-processes data for bragflo
Build/Solaris/relate (Ver:1.45)	$REP/RELATE	Transfers CAMDAT data to another CAMDAT file
Build/Solaris/summarize (Ver:3.02)	$REP/SUMMARIZE	Writes tables of data from many CAMDAT files

Where:

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

CCDFGF

Table 15 .The run script files used were:

File	Path	Comment
RunControl/CCDFGF.py	$REP/CCDFGF	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/CCDFGF	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 16 .The input files used were:

File	Path	Comment
Input/ccgf_PABC09_control_*ri*.inp	$REP/CCDFGF	Input file
Output/cusp_PABC09_*ri*.tbl	$REP/CUTTINGS_S	Release table file
Output/epu_PABC09_ch.dat	$REP/EPAUNI	Release table file
Output/epu_PABC09_rh.dat	$REP/EPAUNI	Release table file
Input/gm_ccgf_PABC09.inp	$REP/GENMESH	Input file
Input/intrusiontimes.in	$REP/PRECCDFGF	Input file
Input/ms_ccgf_PABC09.inp	$REP/MATSET	Input file
Output/sum_dbr_PABC09_*ri*_*so*_*tvvvvv*_L.tbl	$REP/SUMMARIZE	Release table file
Output/sum_dbr_PABC09_*ri*_*so*_*tvvvvv*_M.tbl	$REP/SUMMARIZE	Release table file
Output/sum_dbr_PABC09_*ri*_*so*_*tvvvvv*_U.tbl	$REP/SUMMARIZE	Release table file
Output/sum_nut_PABC09_*ri*_*so*_*tuuuuu*.tbl	$REP/SUMMARIZE	Release table file
Output/sum_panel_con_PABC09_*ri*_*sn*.tbl	$REP/SUMMARIZE	Release table file
Output/sum_panel_int_PABC09_*ri*_*sp*_*ttttt*.tbl	$REP/SUMMARIZE	Release table file
Output/sum_panel_st_PABC09_*ri*_*sn*.tbl	$REP/SUMMARIZE	Release table file
Output/sum_st2d_PABC09_*ri*_mf.tbl	$REP/SUMMARIZE	Release table file
Output/sum_st2d_PABC09_*ri*_mp.tbl	$REP/SUMMARIZE	Release table file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09
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 00100 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

Table 17 .The CVS repositories used were:

CVS Repositories
CCDFGF
CUTTINGS_S
EPAUNI
GENMESH
MATSET
POSTLHS
PRECCDFGF
SUMMARIZE

Table 18 .The log files used were:

File	Path	Comment
RunControl/CCDFGF.log	$REP/CCDFGF	log file
RunControl/CCDFGF.rtf	$REP/CCDFGF	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 19 .The output files produced were:

File	Path	Comment
Output/ccgf_PABC09_reltab_*ri*.dat	$REP/PRECCDFGF	CCDFGF Results
Output/ccgf_PABC09_*ri*.out	$REP/CCDFGF	CCDFGF Results
Output/gm_ccgf_PABC09.cdb	$PATH/GENMESH	CDB transfer file
Output/lhs3_ccgf_PABC09_*ri*_*vvvv*.cdb	$PATH/POSTLHS	LHS file
Output/ms_ccgf_PABC09.cdb	$PATH/MATSET	CDB transfer file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09
$PATH = /home/run_mast/GD/Analyses/PABC09/CCDFGF/RunCCDFGF
i is 1-3
vvv is 001-100

Table 20 .The executable files used were:

File	Path	Comment
Build/Solaris/ccdfgf (Ver:5.03)	$REP/CCDFGF	Constructs complimentary cumulative distribution functions for radionuclide releases
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/preccdfgf (Ver:1.06)	$REP/PRECCDFGF	Pre-processes data for ccdfgf

Where:

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

CUTTINGS

Table 21 .The run script files used were:

File	Path	Comment
RunControl/CUTTINGS_S.py	$REP/CUTTINGS_S	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/CUTTINGS_S	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 22 .The input files used were:

File	Path	Comment
Output/bf3_PABC09_*ri*_*sn*_*vvvv*.cdb	$PATH/BRAGFLO	
Input/cusp_PABC09.inp	$REP/CUTTINGS_S	
Input/gm_cusp_PABC09.inp	$REP/GENMESH	
Input/ms_cusp_PABC09.inp	$REP/MATSET	
Output/mspall_drs_PABC09_*ri*.out	$REP/DRSPALL	

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09
$PATH = /home/run_mast/GD/Analyses/PABC09/BRAGFLO/RunBRAGFLO
i is 1-3
n is 1-5
vvv is 001-100

Table 23 .The CVS repositories used were:

CVS Repositories
BRAGFLO
CUTTINGS_S
DRSPALL
GENMESH
MATSET
POSTLHS

Table 24 .The log files used were:

File	Path	Comment
RunControl/CUTTINGS_S.log	$REP/CUTTINGS_S	log file
RunControl/CUTTINGS_S.rtf	$REP/CUTTINGS_S	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 25 .The output files produced were:

File	Path	Comment
Output/cusp_PABC09_master_*ri*.inp	$REP/CUTTINGS_S	
Output/cusp_PABC09_*ri*.tbl	$REP/CUTTINGS_S	
Output/cusp_PABC09_*ri*_*sn*_*ttttt*_L_*vvvv*.cdb	$PATH/CUTTINGS_S	
Output/cusp_PABC09_*ri*_*sn*_*ttttt*_M_*vvvv*.cdb	$PATH/CUTTINGS_S	
Output/cusp_PABC09_*ri*_*sn*_*ttttt*_U_*vvvv*.cdb	$PATH/CUTTINGS_S	
Output/gm_cusp_PABC09.cdb	$PATH/GENMESH	CDB transfer file
Output/lhs3_cusp_PABC09_*ri*_*vvvv*.cdb	$PATH/POSTLHS	
Output/ms_cusp_PABC09.cdb	$PATH/MATSET	CDB transfer file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09
$PATH = /home/run_mast/GD/Analyses/PABC09/CUTTINGS_S/RunCUTTINGS_S
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

Table 26 .The executable files used were:

File	Path	Comment
Build/Solaris/cuttings_s (Ver:6.03)	$REP/CUTTINGS_S	Computes cuttings/spall generated by drilling
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements

Where:

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

DRSPALL

Table 27. The run script files used were:

File	Path	Comment
RunControl/DRSPALL.py	$REP/DRSPALL	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/DRSPALL	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 28. The input files used were:

File	Path	Comment
Input/drs_PABC09_mxt_p2.inp	$REP/DRSPALL	Input file
Input/drs_PABC09_mxt_p3.inp	$REP/DRSPALL	Input file
Input/drs_PABC09_mxt_p4.inp	$REP/DRSPALL	Input file
Input/drs_PABC09_p1.inp	$REP/DRSPALL	Input file
Input/drs_PABC09_p2.inp	$REP/DRSPALL	Input file
Input/drs_PABC09_p3.inp	$REP/DRSPALL	Input file
Input/drs_PABC09_p4.inp	$REP/DRSPALL	Input file
Input/gm_drs_PABC09.inp	$REP/GENMESH	Input file
Input/ms_drs_PABC09.inp	$REP/MATSET	Input file
Input/sum_drs_PABC09_sphere_p1.inp	$REP/SUMMARIZE	Input file
Input/sum_drs_PABC09_sphere_p2.inp	$REP/SUMMARIZE	Input file
Input/sum_drs_PABC09_sphere_p3.inp	$REP/SUMMARIZE	Input file
Input/sum_drs_PABC09_sphere_p4.inp	$REP/SUMMARIZE	Input file

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 29. The CVS repositories used were:

CVS Repositories
DRSPALL
GENMESH
MATSET
MERGESPALL
POSTLHS
SUMMARIZE

Table 30. The log files used were:

File	Path	Comment
RunControl/DRSPALL.log	$REP/DRSPALL	log file
RunControl/DRSPALL.rtf	$REP/DRSPALL	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 31.The output files produced were:

File	Path	Comment
Output/drs_PABC09_cyl_*ri*_p3_v*yyy*.cdb	$PATH/DRSPALL	CDB transfer file
Output/drs_PABC09_cyl_*ri*_p4_v*xxx*.cdb	$PATH/DRSPALL	CDB transfer file
Output/drs_PABC09_*ri*_p1_v*vvv*.cdb	$PATH/DRSPALL	CDB transfer file
Output/drs_PABC09_*ri*_p2_v*vvv*.cdb	$PATH/DRSPALL	CDB transfer file
Output/drs_PABC09_*ri*_p3_v*vvv*.cdb	$PATH/DRSPALL	CDB transfer file
Output/drs_PABC09_*ri*_p4_v*vvv*.cdb	$PATH/DRSPALL	CDB transfer file
Output/drs_PABC09_cyl_r1_p3_v*www*.cdb	$PATH/DRSPALL	CDB transfer file
Output/drs_PABC09_cyl_r1_p4_v*xxx*.cdb	$PATH/DRSPALL	CDB transfer file
Output/drs_PABC09_cyl_r2_p3_v041.cdb	$PATH/DRSPALL	CDB transfer file
Output/drs_PABC09_cyl_r2_p4_v041.cdb	$PATH/DRSPALL	CDB transfer file
Output/drs_PACB09_cyl_r3_p3_v*yyy*.cdb	$PATH/DRSPALL	CDB transfer file
Output/drs_PABC09_cyl_r3_p4_v*zzz*.cdb	$PATH/DRSPALL	CDB transfer file
Output/gm_drs_PABC09.cdb	$PATH/GENMESH	CDB transfer file
Output/lhs3_drs_PABC09_*ri*_v*vvv*.cdb	$PATH/POSTLHS	CDB file
Output/ms_drs_PABC09.cdb	$PATH/MATSET	CDB transfer file

File	Path	Comment
Output/ms_drs_PABC09.xdbg	$REP/MATSET	Debug file
Output/mspall_drs_PABC09_*ri*.out	$REP/DRSPALL	MergeSpall output file
Output/sum_drs_PABC09_cyl_r*i*_p3.tbl	$REP/SUMMARIZE	Table file
Output/sum_drs_PABC09_cyl_r*i*_p4.tbl	$REP/SUMMARIZE	Table file
Output/sum_drs_PABC09_sphere_r*i*_p1.tbl	$REP/SUMMARIZE	Table file
Output/sum_drs_PABC09_sphere_r*i*_p2.tbl	$REP/SUMMARIZE	Table file
Output/sum_drs_PABC09_sphere_r*i*_p3.tbl	$REP/SUMMARIZE	Table file
Output/sum_drs_PABC09_sphere_r*i*_p4.tbl	$REP/SUMMARIZE	Table file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09
$PATH = /home/run_mast/GD/Analyses/PABC09/BRAGFLO_DBR/RunBRAGFLO_DRS
i is 1-3
www is 032, 036
vvv is 001-100
xxx is 032, 036, 042, 059
yyy is 025, 036
zzz is 001, 025, 036

Table 32. The executable files used were:

File	Path	Comment
Build/Solaris/drspall (Ver:1.21)	$REP/DRSPALL	Computes volume of waste from drilling
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/mergespall (Ver:1.01)	$REP/MERGESPALL	Executable file
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/summarize (Ver:3.02)	$REP/SUMMARIZE	Writes tables of data from many CAMDAT files

Where:

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

EPAUNI

Table 33. The run script files used were:

File	Path	Comment
RunControl/EPAUNI.py	$REP/EPAUNI	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/EPAUNI	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 34. The input files used were:

File	Path	Comment
Input/epu_PABC09_ch.inp	$REP/EPAUNI	
Input/epu_PABC09_ch_misc.inp	$REP/EPAUNI	
Input/epu_PABC09_rh.inp	$REP/EPAUNI	
Input/epu_PABC09_rh_misc.inp	$REP/EPAUNI	

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 35. The CVS repository used was:

File
EPAUNI

Table 36. The log files used were:

File	Path	Comment
RunControl/EPAUNI.log	$REP/EPAUNI	log file
RunControl/EPAUNI.rtf	$REP/EPAUNI	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 37. The output files produced were:

File	Path	Comment
Output/epu_PABC09_ch.dat	$REP/EPAUNI	Radionuclide inventory
Output/epu_PABC09_ch.dia	$REP/EPAUNI	Diagnostic file
Output/epu_PABC09_ch.out	$REP/EPAUNI	supplemental output file
Output/epu_PABC09_ch.out2	$REP/EPAUNI	supplemental output file
Output/epu_PABC09_ch_activity.dia	$REP/EPAUNI	diagnostic file
Output/epu_PABC09_rh.dat	$REP/EPAUNI	Radionuclide inventory
Output/epu_PABC09_rh.dia	$REP/EPAUNI	Diagnostic file
Output/epu_PABC09_rh.out	$REP/EPAUNI	supplemental output file
Output/epu_PABC09_rh.out2	$REP/EPAUNI	supplemental output file
Output/epu_PABC09_rh_activity.dia	$REP/EPAUNI	diagnostic file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 38. The executable file used was:

File	Path	Comment
Build/Solaris/epauni (Ver:1.16)	$REP/EPAUNI	Computes decay of radionuclide components in inventory

Where:

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

LHS

Table 39.The run script files used were:

File	Path	Comment
RunControl/LHS.py	$REP/LHS	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/LHS	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 40. The input file used was:

File	Path	Comment
Input/lhs1_PABC09_r*i*_con.inp	$REP/PRELHS	PRELHS input file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

i is 1-3

Table 41. The CVS repositories used were:

CVS Repositories
LHS
PRELHS

Table 42. The log files used were:

File	Path	Comment
RunControl/LHS.log	$REP/LHS	log file
RunControl/LHS.rtf	$REP/LHS	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 43. The output files produced were:

File	Path	Comment
Output/lhs1_PABC09_*ri*_con.dbg	$REP/PRELHS	PRELHS debug file
Output/lhs1_PABC09_*ri*_con.trn	$REP/PRELHS	PRELHS transfer file
Output/lhs2_PABC09_*ri*_con.dbg	$REP/LHS	LHS debug file
Output/lhs2_PABC09_*ri*_con.trn	$REP/LHS	LHS transfer file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

i is 1-3

Table 44. The executable files used were:

File	Path	Comment
Build/Solaris/lhs (Ver:2.43)	$REP/LHS	Code to sample uncertain parameters
Build/Solaris/prelhs (Ver:2.41)	$REP/PRELHS	Pre-processes data for lhs

Where:

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

NUTS

Table 45 .The run script files used were:

File	Path	Comment
RunControl/NUTS.py	$REP/NUTS	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/NUTS	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 46 .The input files used were:

File	Path	Comment
Input/alg_nut_iso_PABC09.inp	$REP/ALGEBRACDB	Input file
Input/alg_nut_scn_PABC09.inp	$REP/ALGEBRACDB	Input file
Output/bf2_PABC09_*ri*_*sn*_*vvvv*.inp	$REP/PREBRAG	Input file
Output/bf3_PABC09_*ri*_*sn*_*vvvv*.cdb	$PATH/BRAGFLO/RunBRAGFLO/BRAGFLO	CDB transfer file
Input/ms_nut_PABC09.inp	$REP/MATSET	Input file
Input/nut_int_PABC09_*so*_*ttttt*.inp	$REP/NUTS	Input file
Input/nut_iso_PABC09_*sn*.inp	$REP/NUTS	Input file
Input/nut_scn_PABC09_*sn*.inp	$REP/NUTS	Input file
Output/panel_con_PABC09_*ri*_*sm*_*vvvv*.cdb	$PATH/PANEL/RunPANEL/PANEL	CDB transfer file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09
$PATH = /home/run_mast/GD/Analyses/PABC09/RunNUTS
i is 1-3
m is 1-6
n is 1-5
o is 2-5
tttt is 00100 for S2, S4
01000, 03000, 05000, 07000, 09000 for S3, S5
vvv is 001-100

Table 47 .The CVS repositories used were:

CVS Repositories
ALGEBRACDB
BRAGFLO
MATSET
NUTS
PANEL
PREBRAG
SCREEN_NUTS
SUMMARIZE

Table 48 .The log files used were:

File	Path	Comment
RunControl/NUTS.log	$REP/NUTS	log file
RunControl/NUTS.rtf	$REP/NUTS	Formatted log file (Word file)

Where:
$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 49 .The output files produced were:

File	Path	Comment
Output/alg_nut_int_PABC09_r*i*_s*o*_t*uuuuu*_V*VVV*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg_nut_iso_PABC09_r*i*_s*n*_V*VVV*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg_nut_scn_PABC09_r*i*_s*n*_v*vvv*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/ms_nut_PABC09_r*i*_s*n*_V*VVV*.cdb	$PATH/MATSET	CDB transfer file
Output/nut_int_PABC09_r*i*_s*o*_t*uuuuu*_V*VVV*.cdb	$PATH/NUTS	CDB transfer file
Output/nut_iso_PABC09_r*i*_s*n*_V*VVV*.cdb		NOT SAVED:CDB transfer file
Output/nut_scn_PABC09_r*i*_s*n*_v*vvv*.cdb	$PATH/NUTS	CDB transfer file
Output/screen_nut_scn_PABC09_r*i*_EDIT.inp	$REP/SCREEN_NUTS	Input file
Output/screen_nut_scn_PABC09_r*i*_s*n*.out	$REP/SCREEN_NUTS	Output file
Output/sum_nut_PABC09_r*i*_s*n*_t*tttt*.tbl	$PATH/SUMMARIZE	NOT SAVED:Table file
Output/sum_nut_scn_PABC09_r*i*_s*n*.tbl	$REP/SUMMARIZE	Table file

Where:
$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09
$PATH = /home/run_mast/GD/Analyses/PABC09/NUTS/RunNUTS
i is 1-3
n is 1-5
o is 2-5
tttt is 00100 for S1
00100, 00350 for S2, S4

01000, 03000, 05000, 07000, 09000 for S3, S5
uuuuu is 00100 for S2, S4
03000, 05000, 07000, 09000 for S3, S5
vvv is 001-100
VVV are the screened-in vectors listed in Table 50.

Table 50 .The screened-in vectors were:

Replicate	Scenario	Vectors
1	1	2,3,5,6,7,8,9,10,12,13,14,16,17,19,20,22,23,24,25,26,27,28,29,30,34,35,36,38,41,43, 45,46,47,48,49,50,51,52,53,54,55,58,59,60,61,62,63,64,66,67,69,70,71,72,73,74,76, 78,79,80,82, 83,84,86,88,89,90,92,93,94,98
1	2	2,3,5,6,7,8,9,10,12,13,14,16,17,19,20,22,23,24,25,26,27,28,29,30,34,35,36,38,41,43, 45,46,47,48,49,50,51,52,53,54,55,58,59,60,61,62,63,64,66,67,69,70,71,72,73,74,76, 78,79,80,82, 83,84,86,88,89,90,92,93,94,98
1	3	2,3,7,8,9,10,12,13,14,16,17,20,22,23,24,25,27,28,29,30,34,35,36,41,43,45,46,47,49, 50,52,54,55,58,59,60,61,62,63,66,67,70,71,72,76,78,79,80,82,83,84,86,89,90,93,94, 98
1	4	7,9,16,17,27,30,36,45,50,53,55,67,76,78,82,83,98
1	5	7,9,16,17,27,30,36,45,50,53,55,67,76,78,82,83,98
2	1	2,3,4,6,8,9,10,11,12,14,16,17,18,19,20,21,24,25,26,27,28,29,30,31,32,33,34,35,36,37 ,38,39,40,41,43,44,45,48,49,50,51,52,53,54,55,56,59,61,62,63,64,65,66,67,68,69,71, 72,74,75, 77,79,80,81,83,84,87,89,90,91,92,95,96,98,99,100
2	2	2,3,4,6,8,9,10,11,12,14,16,17,18,19,20,21,24,25,26,27,28,29,30,31,32,33,34,35,36,37 ,38,39,40,41,43,44,45,48,49,50,51,52,53,54,55,56,59,61,62,63,64,65,66,67,68,69,71, 72,74,75, 77,79,80,81,83,84,87,89,90,91,92,95,96,98,99,100
2	3	3,4,6,8,9,12,14,16,17,18,20,21,24,25,26,28,29,30,31,32,33,34,35,36,37,38,39,40,41, 44,48,50,51,52,53,54,55,59,63,65,66,67,68,71,72,74,75,77,79,80,84,87,89,90,92,95, 96,98,99
2	4	4,17,21,24,28,30,34,36,40,53,55,63,68,79,90,92,95,98
2	5	4,17,21,24,28,30,34,40,53,55,63,68,79,90,92,95,98
3	1	2,3,4,7,10,11,13,14,15,17,18,19,21,22,24,25,26,27,28,29,30,32,33,34,35,37,38,39,40, 41,42,43,44,45,46,47,49,50,52,53,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71, 73,74,76,77,78,79,84,85,86,88,89,90,91,93,94,95,96,97,98,99,100
3	2	2,3,4,7,10,11,13,14,15,17,18,19,21,22,24,25,26,27,28,29,30,32,33,34,35,37,38,39,40, 41,42,43,44, 45,46,47,49,50,52,53,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70, 71,73,74,76,77,78,79,84,85,86,88,89,90,91,93,94,95,96,97,98,99,100
3	3	2,10,11,14,15,18,21,24,25,26,27,28,29,30,32,33,34,35,37,38,39,40,42,43,44,45,46, 47,49,50,53,56,58,59,60,61,63,64,65,66,67,68,69,73,74,77,78,79,84,85,86,88,89,91, 93,94,95,96,97,98,99
3	4	30,35,37,42,44,47,49,53,59,66,69,77,79,86,91,93,96
3	5	30,35,37,42,44,47,49,53,59,66,69,77,79,86,91,93,96

Table 51 .The executable files used were:

File	Path	Comment
Build/Solaris/algebracdb (Ver:2.36)	$REP/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/nuts (Ver:2.06)	$REP/NUTS	Nuclide Transport system model
Build/Solaris/screen_nuts (Ver:1.01)	$REP/SCREEN_NUTS	

Build/Solaris/summarize (Ver:3.02)	$REP/SUMMARIZE	Writes tables of data from many CAMDAT files

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

PANEL

Table 52. The run script files used were:

File	Path	Comment
RunControl/PANEL.py	$REP/PANEL	Python run control script
RunControl//home/run_mast /GD/Run.py	$REP/PANEL	Main control script

Where:
$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 53. The input files used were:

File	Path	Comment
Input/alg1_panel_PABC09.inp	$REP/ALGEBRACDB	Input file
Output/alg2_bf_PABC09_r*i*_s*n*_v*vv*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Input/alg2_panel_PABC09.inp	$REP/ALGEBRACDB	Input file
Input/alg3_panel_PABC09.inp	$REP/ALGEBRACDB	Input file
Input/gm_panel_PABC09.inp	$REP/GENMESH	Input file
Input/ms_panel_PABC09.inp	$REP/MATSET	Input file
Input/sum_panel_con.inp	$REP/SUMMARIZE	Input file
Input/sum_panel_int.inp	$REP/SUMMARIZE	Input file
Input/sum_panel_st.inp	$REP/SUMMARIZE	Input file

Where:
$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09
$PATH = /home/run_mast/GD/Analyses/PABC09/BRAGFLO/RunBRAGFLO
i is 1-3
n is 1-6
vvv is 001-100

Table 54 .The CVS repositories used were:

CVS Repositories
ALGEBRACDB
GENMESH
MATSET
PANEL
POSTLHS
SUMMARIZE

Table 55 .The log files used were:

File	Path	Comment
RunControl/PANEL.log	$REP/PANEL	log file
RunControl/PANEL.rtf	$REP/PANEL	Formatted log file (Word file)

Where:
$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 56 .The output files produced were:

File	Path	Comment
Output/alg1_panel_PABC09.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg2_panel_PABC09.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg3_panel_PABC09_rj_vwww.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/gm_panel_PABC09.cdb	$PATH/GENMESH	CDB transfer file
Output/lhs3_panel_PABC09_rj_vwww.cdb	$PATH/POSTLHS	CDB transfer file
Output/ms_panel_PABC09.cdb	$PATH/MATSET	CDB transfer file
Output/panel_con_PABC09_rj_sk_vwww.cdb	$PATH/PANEL	CDB transfer file
Output/panel_decay_PABC09_r1_s1_v001.cdb	$PATH/PANEL	CDB transfer file
Output/panel_int_PABC09_rj_so_ttttt_vwww.cdb	$PATH/PANEL	CDB transfer file
Output/sum_panel_con_PABC09_rj_sp.tbl	$REP/SUMMARIZE	Table file
Output/sum_panel_int_PABC09_rj_so_ttttt.tbl	$REP/SUMMARIZE	Table file
Output/sum_panel_st_PABC09_rj_sp.tbl	$REP/SUMMARIZE	Table file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/PABC09
$PATH = /home/run_mast/GD/Analyses/PABC09/PANEL/RunPANEL
j is 1-3
k is 1-6
o is 6
p is 1, 2
q is 1-5
ttttt is 00100, 00350, 01000, 02000, 04000, 06000, 09000
www is 001-100

Table 57 .The executable files used were:

File	Path	Comment
Build/Solaris/algebracdb (Ver:2.36)	$REP/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/panel (Ver:4.04)	$REP/PANEL	Computes release concentrations of nuclides from repository
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/summarize (Ver:3.02)	$REP/SUMMARIZE	Writes tables of data from many CAMDAT files

Where:

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

SECOTP2D

Table 58 .The run script files used were:

File	Repository	Comment
RunControl//home/run_mast/GD/Run.py	$REP/SECOTP2D	Main control script
RunControl/SECOTP2D.py	$REP/SECOTP2D	Python run control script

Where:

$REP=/nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 59 .The input files used were:

File	Repository	Comment
Input/alg_st2d_PABC09.inp	$REP/ALGEBRACDB	Input file
Output/R*i*/$*f*/$*k*/dtrk.dbg	$REP/MODFLOW	Modflow 'budget' file
Output/R*i*/$*f*/$*k*/dtrkmf_stdout	$REP/MODFLOW	Modflow 'budget' file
Input/gm_st2d_PABC09.inp	$REP/GENMESH	Input file
Input/grope_st2d_PABC09.inp	$REP/GROPECDB	Input file
Input/keepers	$REP/MODFLOW	Modflow 'keepers' file
Output/R*i*/$*f*/$*k*/mf2k_stdout	$REP/MODFLOW	Modflow 'budget' file
Output/R*i*/$*f*/$*k*/mf2k_stdout.50	$REP/MODFLOW	Modflow 'budget' file
Output/R*i*/$*f*/$*k*/modeled_flow.bud	$REP/MODFLOW	Modflow 'budget' file
Output/R*i*/$*f*/$*k*/modeled_flow_50.bud	$REP/MODFLOW	Modflow 'budget' file
Output/R*i*/$*f*/$*k*/modeled_head.bin	$REP/MODFLOW	Modflow 'budget' file
Output/R*i*/$*f*/$*k*/modeled_head.lst	$REP/MODFLOW	Modflow 'budget' file
Output/R*i*/$*f*/$*k*/modeled_head_50.bin	$REP/MODFLOW	Modflow 'budget' file
Output/R*i*/$*f*/$*k*/modeled_head_50.lst	$REP/MODFLOW	Modflow 'budget' file
Output/R*i*/$*f*/$*k*/modeled_K_field.mod.50	$REP/MODFLOW	Modflow 'budget' file
Output/R*i*/$*f*/$*k*/modeled_mined_K_field.mod	$REP/MODFLOW	Modflow 'budget' file
Output/R*i*/$*f*/$*k*/modeled_mined_K_field.mod.dbg	$REP/MODFLOW	Modflow 'budget' file
Input/ms_st2d_PABC09.inp	$REP/MATSET	Input file
Output/ R*i*/$*f*/$*k*/ReadMe	$REP/MODFLOW	Modflow 'budget' file
Input/rel_st2d_PABC09.inp	$REP/RELATE	Input file
Input/st2d1_PABC09.inp	$REP/PRESECOTP2D	Input file
Input/st2d1_PABC09_mod.inp	$REP/PRESECOTP2D	Input file
Input/sum1_st2d_PABC09.inp	$REP/SUMMARIZE	Input file
Input/sum_st2d_PABC09.inp	$REP/SUMMARIZE	Input file

Where:

$REP=/nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

i is 1-3

$f is full, partial

$k is r001, r002, r004, r006, r007, r009, r010, r012, r013, r017, r024, r027, r028, r029, r032, r034, r037, r038, r040, r041, r045, r051, r052, r053, r054, r055, r058, r059, r060, r061, r064, r070, r073, r074, r076, r078, r082, r083, r084, r090,r092, r095, r097, r098, r102, r104, r137, r142, r191, r203,r207, r256, r260, r273, r276, r279, r298, r327, r328, r361, r431, r440, r465, r486, r489, r506, r508, r511, r515, r522,r568, r571, r631, r634, r640, r652, r655, r657, r664, r669,r694, r707, r727, r752, r791, r806, r808, r809, r814, r823, r861, r883, r902, r910, r921, r922, r940, r981, r982, r984

Table 60 .The CVS repositories used were:

CVS Repositories
ALGEBRACDB
GENMESH
GROPECDB
MATSET
MODFLOW
POSTLHS
POSTSECOTP2D
PRESECOTP2D
RELATE
SECOTP2D
SUMMARIZE

Table 61 .The log files used were:

File	Repository	Comment
RunControl/SECOTP2D.log	$REP/SECOTP2D	log file
RunControl/SECOTP2D.rtf	$REP/SECOTP2D	Formatted log file (Word file)

Where:

$REP=/nfs/data/CVSLIB/WIPP_ANALYSES/PABC09

Table 62 .The output files produced were:

File	Repository	Comment
Output/alg_st2d_PABC09_r*i*_v*vvv*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/gm_st2d_PABC09.cdb	$PATH/GENMESH	CDB transfer file
Output/lhs3_st2d_PABC09_r*i*_v*vvv*.cdb	$ PATH /POSTLHS	CDB transfer file
Output/ms_st2d_PABC09.cdb	$ PATH /MATSET	CDB transfer file
Output/rel_st2d_PABC09_r*i*_v*vvv*.cdb	$ PATH /RELATE	CDB transfer file
Output/st2d2_PABC09_r*i*_mf_v*vvv*.xbin		NOT SAVED:
Output/st2d2_PABC09_r*i*_mp_v*vvv*.xbin		NOT SAVED:
Output/st2d3_PABC09_r*i*_mf_v*vvv*.cdb	$ PATH /POSTSECOTP2D	CDB transfer file
Output/st2d3_PABC09_r*i*_mp_v*vvv*.cdb	$ PATH /POSTSECOTP2D	CDB transfer file
Output/sum1_st2d_PABC09_r*i*.tbl	$REP/SUMMARIZE	Table file
Output/sum_st2d_PABC09_r*i*_mf.tbl	$REP/SUMMARIZE	Table file
Output/sum_st2d_PABC09_r*i*_mp.tbl	$REP/SUMMARIZE	Table file

Where:

$REP=/nfs/data/CVSLIB/WIPP_ANALYSES/PABC09
$PATH=/home/run_mast/GD/Analyses/PABC09/SECOTP2D/RunSECOTP2D
i is 1-3
v is 001-100

Table 63 .The executable files used were:

File	Repository	Comment
Build/Solaris/algebracdb (Ver:2.36)	$REP/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/gropecdb (Ver:2.13)	$REP/GROPECDB	Displays CAMDAT data in text form
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/postsecotp2d (Ver:1.05)	$REP/POSTSECOTP2D	Post-processes data for secotp2d
Build/Solaris/presecotp2d (Ver:1.23)	$REP/PRESECOTP2D	Pre-processes data for secotp2d
Build/Solaris/relate (Ver:1.45)	$REP/RELATE	Transfers CAMDAT data to another CAMDAT file
Build/Solaris/secotp2d (Ver:1.43)	$REP/SECOTP2D	SECO 2D transport code
Build/Solaris/summarize (Ver:3.02)	$REP/SUMMARIZE	Writes tables of data from many CAMDAT files

Where:

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

Analysis CRA14BL

BRAGFLO

Table 64 .The run script files used were:

File	Path	Comment
RunControl/BRAGFLO.py	$REP/BRAGFLO	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/BRAGFLO	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

Table 65 .The input files used were:

File	Path	Comment
Input/alg1_bf_CRA14BL.inp	$REP/ALGEBRACDB	Input file
Input/alg2_bf_CRA14BL.inp	$REP/ALGEBRACDB	Input file
Input/bf1_CRA14BL_s*n*.inp	$REP/PREBRAG	Input file
Input/bf1_CRA14BL_s*n*_mod1.inp	$REP/PREBRAG	Input file
Input/bf1_CRA14BL_s*n*_mod2.inp	$REP/PREBRAG	Input file
Input/bf2_CRA14BL_closure.dat	$REP/BRAGFLO	Input file
Input/gm_bf_CRA14BL.inp	$REP/GENMESH	Input file
Input/ic_bf_CRA14BL.inp	$REP/ICSET	Input file
Input/ms_bf_CRA14BL.inp	$REP/MATSET	Input file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

n is 1-6

Table 66 .The CVS repositories used were:

CVS Repositories
ALGEBRACDB
BRAGFLO
GENMESH
ICSET
MATSET
POSTBRAG
POSTLHS
PREBRAG

Table 67 .The log files used were:

File	Path	Comment
RunControl/BRAGFLO.log	$REP/BRAGFLO	log file
RunControl/BRAGFLO.rtf	$REP/BRAGFLO	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

Table 68 .The output files produced were:

File	Path	Comment
Output/alg1_bf_CRA14BL_r*i*_v*vvv*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg2_bf_CRA14BL_r*i*_s*n*_v*vvv*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/bf2_CRA14BL_r*i*_s*n*_v*vvv*.inp	$REP/PREBRAG	BRAGFLO input file
Output/bf2_CRA14BL_r*i*_s*n*_v*vvv*.log	$REP/BRAGFLO	Logfile
Output/bf2_CRA14BL_r*i*_s*n*_v*vvv*.sum	$REP/BRAGFLO	Summary file
Output/bf3_CRA14BL_r*i*_s*n*_v*vvv*.cdb	$PATH/BRAGFLO	CDB transfer file
Output/gm_bf_CRA14BL.cdb	$PATH/GENMESH	CDB transfer file
Output/ic_bf_CRA14BL_r*i*_v*vvv*.cdb	$PATH/ICSET	CDB transfer file
Output/lhs3_bf_CRA14BL_r*i*_v*vvv*.cdb	$PATH/POSTLHS	CDB transfer file
Output/ms_bf_CRA14BL.cdb	$PATH/MATSET	CDB transfer file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL
$PATH = /home/run_mast/GD/Analyses/CRA14BL/BRAGFLO/RunBRAGFLO
i is 1-3
n is 1-6
vvv is 001-100

Table 69 .The executable files used were:

File	Path	Comment
Build/Solaris/algebracdb (Ver:2.36)	$REP/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Solaris/bragflo (Ver:6.03)	$REP/BRAGFLO	Computes brine and gas flow in the repository
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/icset (Ver:2.23)	$REP/ICSET	Assigns initial conditions to the CAMDAT grid elements
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postbrag (Ver:4.02)	$REP/POSTBRAG	Post-processes data for bragflo
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/prebrag (Ver:8.03)	$REP/PREBRAG	Pre-processes data for bragflo

Where:

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

BRAGFLO_DBR

Table 70 .The run script files used were:

File	Path	Comment
RunControl/BRAGFLO_DBR.py	$REP/BRAGFLO_DBR	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/BRAGFLO_DBR	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

Table 71 .The input files used were:

File	Path	Comment
Input/alg1_dbr_CRA14BL.inp	$REP/ALGEBRACDB	
Input/alg2_dbr_CRA14BL_*so*.inp	$REP/ALGEBRACDB	
Input/alg3_dbr_CRA14BL_L.inp	$REP/ALGEBRACDB	
Input/alg3_dbr_CRA14BL_M.inp	$REP/ALGEBRACDB	
Input/alg3_dbr_CRA14BL_U.inp	$REP/ALGEBRACDB	
Input/bf1_dbr_CRA14BL_L.inp	$REP/PREBRAG	
Input/bf1_dbr_CRA14BL_M.inp	$REP/PREBRAG	
Input/bf1_dbr_CRA14BL_*sn*_100_L.inp	$REP/PREBRAG	
Input/bf1_dbr_CRA14BL_*sn*_100_M.inp	$REP/PREBRAG	
Input/bf1_dbr_CRA14BL_*sn*_100_U.inp	$REP/PREBRAG	
Input/bf1_dbr_CRA14BL_U.inp	$REP/PREBRAG	
Output/bf3_CRA14BL_r*i*_*so*_*vvvv*.cdb	$PATH/BRAGFLO/RunBRAGFLO /BRAGFLO	
Output/cusp_CRA14BL_r*i*_*so*_*ttttt*_L_*vvvv*.cdb	$PATH/CUTTINGS_S/RunCUTTINGS_S/CUTTINGS_S	
Output/cusp_CRA14BL_r*i*_*so*_*ttttt*_M_*vvvv*.cdb	$PATH/CUTTINGS_S/RunCUTTINGS_S/CUTTINGS_S	
Output/cusp_CRA14BL_r*i*_*so*_*ttttt*_U_*vvvv*.cdb	$PATH/CUTTINGS_S/RunCUTTINGS_S/CUTTINGS_S	
Input/gm_dbr_CRA14BL.inp	$REP/GENMESH	
Input/ic_dbr_CRA14BL_*so*.inp	$REP/ICSET	
Input/ms_dbr_CRA14BL.inp	$REP/MATSET	
Input/rel1_dbr_CRA14BL.inp	$REP/RELATE	
Input/rel2_dbr_CRA14BL_*so*.inp	$REP/RELATE	
Input/sum_dbr.inp	$REP/SUMMARIZE	

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL
$PATH = /home/run_mast/GD/Analyses/CRA14BL
i is 1
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

Table 72 .The CVS repositories used were:

CVS Repositories
ALGEBRACDB
BRAGFLO
BRAGFLO_DBR
CUTTINGS_S
GENMESH
ICSET
MATSET
POSTBRAG
POSTLHS
PREBRAG
RELATE
SUMMARIZE

Table 73 .The log files used were:

File	Path	Comment
RunControl/BRAGFLO_DBR.log	$REP/BRAGFLO_DBR	log file
RunControl/BRAGFLO_DBR.rtf	$REP/BRAGFLO_DBR	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

Table 74 .The output files produced were:

File	Path	Comment
Output/alg1_dbr_CRA14BL_ri_so_twwwww_vvvv.cdb	$PATH/ALGEBRACDB	
Output/alg2_dbr_CRA14BL_ri_so_twwwww_vvvv.cdb	$PATH/ALGEBRACDB	
Output/alg3_dbr_CRA14BL_ri_so_twwwww_L_vvvv.cdb	$PATH/ALGEBRACDB	
Output/alg3_dbr_CRA14BL_ri_so_twwwww_M_vvvv.cdb	$PATH/ALGEBRACDB	
Output/alg3_dbr_CRA14BL_ri_so_twwwww_U_vvvv.cdb	$PATH/ALGEBRACDB	
Output/bf2_dbr_CRA14BL_ri_so_twwwww_L_vvvv.inp	$REP/BRAGFLO_DBR	
Output/bf2_dbr_CRA14BL_ri_so_twwwww_M_vvvv.inp	$REP/BRAGFLO_DBR	
Output/bf2_dbr_CRA14BL_ri_so_twwwww_U_vvvv.inp	$REP/BRAGFLO_DBR	
Output/bf3_dbr_CRA14BL_ri_so_twwwww_L_vvvv.cdb	$PATH/BRAGFLO_DBR	
Output/bf3_dbr_CRA14BL_ri_so_twwwww_M_vvvv.cdb	$PATH/BRAGFLO_DBR	
Output/bf3_dbr_CRA14BL_ri_so_twwwww_U_vvvv.cdb	$PATH/BRAGFLO_DBR	
Output/gm_dbr_CRA14BL.cdb	$PATH/GENMESH	
Output/ic_dbr_CRA14BL_ri_so_twwwww_vvvv.cdb	$PATH/ICSET	
Output/ms_dbr_CRA14BL.cdb	$PATH/MATSET	
Output/rel1_dbr_CRA14BL_ri_so_twwwww_vvvv.cdb	$PATH/RELATE	
Output/rel2_dbr_CRA14BL_ri_so_twwwww_vvvv.cdb	$PATH/RELATE	
Output/sum_dbr_CRA14BL_ri_so_twwwww_L.tbl	$REP/SUMMARIZE	
Output/sum_dbr_CRA14BL_ri_so_twwwww_M.tbl	$REP/SUMMARIZE	
Output/sum_dbr_CRA14BL_ri_so_twwwww_U.tbl	$REP/SUMMARIZE	

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL
$PATH = /home/run_mast/GD/Analyses/CRA14BL/BRAGFLO_DBR/RunBRAGFLO_DBR
i is 1
o is 1-5
wwwww 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

Table 75 .The executable files used were:

File	Path	Comment
Build/Solaris/algebracdb (Ver:2.36)	$REP/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Solaris/bragflo (Ver:6.03)	$REP/BRAGFLO	Computes brine and gas flow in the repository
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/icset (Ver:2.23)	$REP/ICSET	Assigns initial conditions to the CAMDAT grid elements
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postbrag	$REP/POSTBRAG	Post-processes data for bragflo

File	Path	Comment
(Ver:4.02)		
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/prebrag (Ver:8.03)	$REP/PREBRAG	Pre-processes data for bragflo
Build/Solaris/relate (Ver:1.45)	$REP/RELATE	Transfers CAMDAT data to another CAMDAT file
Build/Solaris/summarize (Ver:3.02)	$REP/SUMMARIZE	Writes tables of data from many CAMDAT files

Where:

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

CCDFGF

Table 76 .The run script files used were:

File	Path	Comment
RunControl/CCDFGF.py	$REP/CCDFGF	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/CCDFGF	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

Table 77 .The input files used were:

File	Path	Comment
Input/ccgf_CRA14BL_control_*ri*.inp	$REP/CRA14BL/CCDFGF	Input file
Output/cusp_CRA14BL_*ri*.tbl	$REP/CRA14BL/CUTTINGS_S	Release table file
Output/epu_CRA14BL_ch.dat	$REP/CRA14BL/EPAUNI	Release table file
Output/epu_CRA14BL_rh.dat	$REP/CRA14BL/EPAUNI	Release table file
Input/gm_ccgf_CRA14BL.inp	$REP/CRA14BL/GENMESH	Input file
Input/intrusiontimes.in	$REP/CRA14BL/PRECCDFGF	Input file
Input/ms_ccgf_CRA14BL.inp	$REP/CRA14BL/MATSET	Input file
Output/sum_dbr_CRA14BL_*ri*_so_tvvvvv_L.tbl	$REP/CRA14BL/SUMMARIZE	Release table file
Output/sum_dbr_CRA14BL_*ri*_so_tvvvvv_M.tbl	$REP/CRA14BL/SUMMARIZE	Release table file
Output/sum_dbr_CRA14BL_*ri*_so_tvvvvv_U.tbl	$REP/CRA14BL/SUMMARIZE	Release table file
Output/sum_nut_CRA14BL_*ri*_so_t*uuuuu*.tbl	$REP/CRA14BL/SUMMARIZE	Release table file
Output/sum_panel_con_CRA14BL_b1_*ri*_s*n*.tbl	$REP/CRA14BL/SUMMARIZE	Release table file
Output/sum_panel_int_CRA14BL_b1_*ri*_s*p*_t*tttt*.tbl	$REP/CRA14BL/SUMMARIZE	Release table file
Output/sum_panel_st_CRA14BL_b1_*ri*_s*n*.tbl	$REP/CRA14BL/SUMMARIZE	Release table file
Output/sum_st2d_PABC09_*ri*_mf.tbl	$REP/PABC09/SUMMARIZE	Release table file
Output/sum_st2d_PABC09_*ri*_mp.tbl	$REP/PABC09/SUMMARIZE	Release table file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/

i is 1

n is 1-2

o is 1-5

p is 6

tttt is 00100, 00350, 01000, 02000, 04000, 06000, 09000

uuuuu is 00100 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

Table 78 .The CVS repositories used were:

CVS Repositories
CCDFGF
CUTTINGS_S
EPAUNI
GENMESH
MATSET
POSTLHS
PRECCDFGF
SUMMARIZE

Table 79 .The log files used were:

File	Path	Comment
RunControl/CCDFGF.log	$REP/CCDFGF	log file
RunControl/CCDFGF.rtf	$REP/CCDFGF	Formatted log file (Word file)

Where:
$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

Table 80 .The output files produced were:

File	Path	Comment
Output/ccgf_CRA14BL_reltab_*ri*.dat	$REP/PRECCDFGF	CCDFGF Reults
Output/ccgf_CRA14BL_*ri*.out	$REP/CCDFGF	CCDFGF Reults
Output/gm_ccgf_CRA14BL.cdb	$PATH/GENMESH	CDB transfer file
Output/lhs3_ccgf_CRA14BL_*ri*_*vvvv*.cdb	$PATH/POSTLHS	LHS file
Output/ms_ccgf_CRA14BL.cdb	$PATH/MATSET	CDB transfer file

Where:
$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL
$PATH = /home/run_mast/GD/Analyses/CRA14BL/CCDFGF/RunCCDFGF
i is 1
vvv is 001-100

Table 81 .The executable files used were:

File	Path	Comment
Build/Solaris/ccdfgf (Ver:6.01)	$REP/CCDFGF	Constructs complimentary cumulative distribution functions for radionuclide releases
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/preccdfgf (Ver:2.01)	$REP/PRECCDFGF	Pre-processes data for ccdfgf

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

CUTTINGS_S

Table 82 .The run script files used were:

File	Path	Comment
RunControl/CUTTINGS_S.py	$REP/CUTTINGS_S	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/CUTTINGS_S	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

Table 83 .The input files used were:

File	Path	Comment
Output/bf3_CRA14BL_r*i*_s*n*_vvvv.cdb	$PATH/CRA14BL/BRAGFLO/RunBRAGFLO/BRAGFLO	
Input/cusp_CRA14BL.inp	$REP/CRA14BL/CUTTINGS_S	
Input/gm_cusp_CRA14BL.inp	$REP/CRA14BL/GENMESH	
Input/ms_cusp_CRA14BL.inp	$REP/CRA14BL/MATSET	
Output/mspall_drs_PABC09_r*i*.out	$REP/PABC09/DRSPALL	

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES
$PATH = /home/run_mast/GD/Analyses
i is 1
n is 1-5
vvv is 001-100

Table 84 .The CVS repositories used were:

CVS Repositories
BRAGFLO
CUTTINGS_S
DRSPALL
GENMESH
MATSET
POSTLHS

Table 85 .The log files used were:

File	Path	Comment
RunControl/CUTTINGS_S.log	$REP/CUTTINGS_S	log file
RunControl/CUTTINGS_S.rtf	$REP/CUTTINGS_S	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

Table 86 .The output files produced were:

File	Path	Comment
Output/cusp_CRA14BL_master_*ri*.inp	$REP/CUTTINGS_S	
Output/cusp_CRA14BL_*ri*.tbl	$REP/CUTTINGS_S	
Output/cusp_CRA14BL_*ri*_*sn*_*ttttt*_L_*vvvv*.cdb	$PATH/CUTTINGS_S	
Output/cusp_CRA14BL_*ri*_*sn*_*ttttt*_M_*vvvv*.cdb	$PATH/CUTTINGS_S	
Output/cusp_CRA14BL_*ri*_*sn*_*ttttt*_U_*vvvv*.cdb	$PATH/CUTTINGS_S	
Output/gm_cusp_CRA14BL.cdb	$PATH/GENMESH	CDB transfer file
Output/lhs3_cusp_CRA14BL_*ri*_*vvvv*.cdb	$PATH/POSTLHS	
Output/ms_cusp_CRA14BL.cdb	$PATH/MATSET	CDB transfer file

Where:
$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL
$PATH = /home/run_mast/GD/Analyses/CRA14BL/CUTTINGS_S/RunCUTTINGS_S
i is 1
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

Table 87 .The executable files used were:

File	Path	Comment
Build/Solaris/cuttings_s (Ver:6.03)	$REP/CUTTINGS_S	Computes cuttings/spall generated by drilling
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements

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

EPAUNI

Table 88 .The run script files used were:

File	Path	Comment
RunControl/EPAUNI.py	$REP/EPAUNI	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/EPAUNI	Main control script

Where:
$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

Table 89 .The input files used were:

File	Path	Comment
Input/epu_CRA14BL_ch.inp	$REP/EPAUNI	
Input/epu_CRA14BL_ch_misc.inp	$REP/EPAUNI	
Input/epu_CRA14BL_rh.inp	$REP/EPAUNI	
Input/epu_CRA14BL_rh_misc.inp	$REP/EPAUNI	

Where:
$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

Table 90 .The CVS repository used was:

File
EPAUNI

Table 91 .The log files used were:

File	Path	Comment
RunControl/EPAUNI.log	$REP/EPAUNI	log file
RunControl/EPAUNI.rtf	$REP/EPAUNI	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

Table 92 .The output files produced were:

File	Path	Comment
Output/epu_CRA14BL_ch.dat	$REP/EPAUNI	Radionuclide inventory
Output/epu_CRA14BL_ch.dia	$REP/EPAUNI	Diagnostic file
Output/epu_CRA14BL_ch.out	$REP/EPAUNI	supplemental output file
Output/epu_CRA14BL_ch.out2	$REP/EPAUNI	supplemental output file
Output/epu_CRA14BL_ch_activity.dia	$REP/EPAUNI	diagnostic file
Output/epu_CRA14BL_rh.dat	$REP/EPAUNI	Radionuclide inventory
Output/epu_CRA14BL_rh.dia	$REP/EPAUNI	Diagnostic file
Output/epu_CRA14BL_rh.out	$REP/EPAUNI	supplemental output file
Output/epu_CRA14BL_rh.out2	$REP/EPAUNI	supplemental output file
Output/epu_CRA14BL_rh_activity.dia	$REP/EPAUNI	diagnostic file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

Table 93 .The executable file used was:

File	Path	Comment
Build/Solaris/epauni (Ver:1.16)	$REP/EPAUNI	Computes decay of radionuclide components in inventory

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

LHS

Table 94 .The run script files used were:

File	Path	Comment
RunControl/LHS.py	$REP/LHS	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/LHS	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

Table 95 .The input file used was:

File	Path	Comment
Input/lhs1_CRA14BL_ri_con.inp	$REP/PRELHS	PRELHS input file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

i is 1

Table 96 .The CVS repositories used were:

CVS Repositories
LHS
PRELHS

Table 97 .The log files used were:

File	Path	Comment
RunControl/LHS.log	$REP/LHS	log file
RunControl/LHS.rtf	$REP/LHS	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

Table 98 .The output files produced were:

File	Path	Comment
Output/lhs1_CRA14BL_r*i*_con.dbg	$REP/PRELHS	PRELHS debug file
Output/lhs1_CRA14BL_r*i*_con.trn	$REP/PRELHS	PRELHS transfer file
Output/lhs2_CRA14BL_r*i*_con.dbg	$REP/LHS	LHS debug file
Output/lhs2_CRA14BL_r*i*_con.trn	$REP/LHS	LHS transfer file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

i is 1

Table 99 .The executable files used were:

File	Path	Comment
Build/Solaris/lhs (Ver:2.43)	$REP/LHS	Code to sample uncertain parameters
Build/Solaris/prelhs (Ver:2.41)	$REP/PRELHS	Pre-processes data for lhs

Where:

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

NUTS

Table 100 .The run script files used were:

File	Path	Comment
RunControl/NUTS.py	$REP/NUTS	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/NUTS	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

Table 101 .The input files used were:

File	Path	Comment
Input/alg_nut_iso_CRA14BL.inp	$REP/ALGEBRACDB	Input file
Input/alg_nut_scn_CRA14BL.inp	$REP/ALGEBRACDB	Input file
Output/bf2_CRA14BL_r*i*_s*n*_v*vvv*.inp	$REP/PREBRAG	Input file
Output/bf3_CRA14BL_r*i*_s*n*_v*vvv*.cdb	$PATH/BRAGFLO/RunBRAGFLO/BRAGFLO	CDB transfer file
Input/ms_nut_CRA14BL.inp	$REP/MATSET	Input file
Input/nut_int_CRA14BL_s*o*_t*tttt*.inp	$REP/NUTS	Input file
Input/nut_iso_CRA14BL_s*n*.inp	$REP/NUTS	Input file

File	Path	Comment
Input/nut_scn_CRA14BL_*sn*.inp	$REP/NUTS	Input file
Output/panel_con_CRA14BL_b1_*ri*_*sm*_*vvvv*.cdb	$PATH/PANEL/RunPANEL/PANEL	CDB transfer file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL
$PATH = /home/run_mast/GD/Analyses/CRA14BL
i is 1
m is 1-6
n is 1-5
o is 2-5
ttttt is 00100 for S2, S4
03000, 05000, 07000, 09000 for S3, S5
vvv is 001-100

Table 102 .The CVS repositories used were:

CVS Repositories
ALGEBRACDB
BRAGFLO
MATSET
NUTS
PANEL
PREBRAG
SCREEN_NUTS
SUMMARIZE

Table 103 .The log files used were:

File	Path	Comment
RunControl/NUTS.log	$REP/NUTS	log file
RunControl/NUTS.rtf	$REP/NUTS	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

Table 104 .The output files produced were:

File	Path	Comment
Output/alg_nut_int_CRA14BL_*ri*_*so*_*tuuuuu*_*VVVV*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg_nut_iso_CRA14BL_*ri*_*sn*_*VVVV*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg_nut_scn_CRA14BL_*ri*_*sn*_*vvvv*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/ms_nut_CRA14BL_*ri*_*sn*_*VVVV*.cdb	$PATH/MATSET	CDB transfer file
Output/nut_int_CRA14BL_*ri*_*so*_*tuuuuu*_*VVVV*.cdb	$PATH/NUTS	CDB transfer file
Output/nut_iso_CRA14BL_*ri*_*sn*_*VVVV*.cdb		NOT SAVED:CDB transfer file
Output/nut_scn_CRA14BL_*ri*_*sn*_*vvvv*.cdb	$PATH/NUTS	CDB transfer file
Output/screen_nut_scn_CRA14BL_*ri*_EDIT.inp	$REP/SCREEN_NUTS	Input file
Output/screen_nut_scn_CRA14BL_*ri*_*sn*.out	$REP/SCREEN_NUTS	Output file
Output/sum_nut_CRA14BL_*ri*_*sn*_*ttttt*.tbl		NOT SAVED:Table file
Output/sum_nut_scn_CRA14BL_*ri*_*sn*.tbl	$REP/SUMMARIZE	Table file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL
$PATH = /home/run_mast/GD/Analyses/CRA14BL/NUTS/RunNUTS
i is 1
n is 1-5
o is 2-5
ttttt is 00100 for S1
00100, 00350 for S2, S4
01000, 03000, 05000, 07000, 09000 for S3, S5
uuuuu is 00100 for S2, S4
03000, 05000, 07000, 09000 for S3, S5

vvv is 001-100
VVV are the screened-in vectors listed in Table 105.

Table 105 .The screened-in vectors were:

Replicate	Scenario	Vectors
1	1	2,3,5,6,7,8,9,10,12,13,14,16,17,19,20,22,23,24,25,26,27,28,29,30,34,35,36,38,39,41,43,45,46,47,48,49,50,51,52,53,54,55,58,59,60,61,62,63,64,66,67,69,70,71,72,74,76,78,79,80,82,83,84,86,88,89,90,91,92,93,94,98
1	2	2,3,5,6,7,8,9,10,12,13,14,16,17,19,20,22,23,24,25,26,27,28,29,30,34,35,36,38,39,41,43,45,46,47,48,49,50,51,52,54,55,58,59,60,61,62,63,64,66,67,69,70,71,72,74,76,78,79,80,82,83,84,86,88,89,90,91,92,93,94,98
1	3	2,3,7,8,9,10,12,13,14,16,17,20,22,23,24,25,27,28,29,30,34,35,36,38,41,43,45,46,47,49,50,51,52,54,55,58,59,60,61,62,63,66,67,69,70,72,76,78,79,80,82,83,86,89,90,93,98
1	4	2,7,9,12,16,17,27,28,30,36,45,50,55,67,76,78,79,83,93,98
1	5	7,9,12,16,17,27,28,30,36,45,50,55,67,76,78,83,93,98

Table 106 .The executable files used were:

File	Path	Comment
Build/Solaris/algebracdb (Ver:2.36)	$REP/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/nuts (Ver:2.06)	$REP/NUTS	Nuclide Transport system model
Build/Solaris/screen_nuts (Ver:1.01)	$REP/SCREEN_NUTS	
Build/Solaris/summarize (Ver:3.02)	$REP/SUMMARIZE	Writes tables of data from many CAMDAT files

Where:

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

PANEL

Table 107 .The run script files used were:

File	Path	Comment
RunControl/PANEL.py	$REP/PANEL	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/PANEL	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

Table 108 .The input files used were:

File	Path	Comment
Input/alg1_panel_CRA14BL.inp	$REP/ALGEBRACDB	Input file
Output/alg2_bf_CRA14BL_r*i*_s*n*_v*vvv*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Input/alg2_panel_CRA14BL_b1.inp	$REP/ALGEBRACDB	Input file
Input/alg3_panel_CRA14BL_b1.inp	$REP/ALGEBRACDB	Input file
Input/gm_panel_CRA14BL.inp	$REP/GENMESH	Input file
Input/ms_panel_CRA14BL.inp	$REP/MATSET	Input file
Input/sum_panel_con.inp	$REP/SUMMARIZE	Input file
Input/sum_panel_int.inp	$REP/SUMMARIZE	Input file
Input/sum_panel_st.inp	$REP/SUMMARIZE	Input file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL
$PATH = /home/run_mast/GD/Analyses/CRA14BL/BRAGFLO/RunBRAGFLO
i is 1
n is 1-6
vvv is 001-100

Table 109 .The CVS repositories used were:

CVS Repositories
ALGEBRACDB
GENMESH
MATSET
PANEL
POSTLHS
SUMMARIZE

Table 110 .The log files used were:

File	Path	Comment
RunControl/PANEL.log	$REP/PANEL	log file
RunControl/PANEL.rtf	$REP/PANEL	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL

Table 111 .The output files produced were:

File	Path	Comment
Output/alg1_panel_CRA14BL.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg2_panel_CRA14BL_b1.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg3_panel_CRA14BL_b1_r*i*_v*wwww*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/gm_panel_CRA14BL.cdb	$PATH/GENMESH	CDB transfer file
Output/lhs3_panel_CRA14BL_b1_r*i*.xdbg		NOT SAVED:Debug file
Output/lhs3_panel_CRA14BL_b1_r*i*_v*wwww*.cdb	$PATH/POSTLHS	CDB transfer file
Output/ms_panel_CRA14BL.cdb	$PATH/MATSET	CDB transfer file
Output/panel_con_CRA14BL_b1_r*i*_s*k*_v*wwww*.cdb	$PATH/PANEL	CDB transfer file
Output/panel_decay_CRA14BL_r*i*_s*n*_v*vvv*.cdb	$PATH/PANEL	CDB transfer file
Output/panel_int_CRA14BL_b1_r*i*_s*q*_t*ttttt*_v*wwww*.cdb	$PATH/PANEL	CDB transfer file
Output/sum_panel_con_CRA14BL_b1_r*i*_s*o*.tbl	$REP/SUMMARIZE	Table file
Output/sum_panel_int_CRA14BL_b1_r*i*_s*q*_t*ttttt*.tbl	$REP/SUMMARIZE	Table file

File	Path	Comment
Output/sum_panel_st_CRA14BL_b1_*ri*_*so*.tbl	$REP/SUMMARIZE	Table file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BL
$PATH = /home/run_mast/GD/Analyses/CRA14BL/PANEL/RunPANEL
i is 1
k is 1-6
n is 1
o is 1-2
q is 6
ttttt is 00100, 00350, 01000, 02000, 04000, 06000, 09000
vvv is 001
www is 001-100

Table 112 .The executable files used were:

File	Path	Comment
Build/Solaris/algebracdb (Ver:2.36)	$REP/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/panel (Ver:4.04)	$REP/PANEL	Computes release concentrations of nuclides from repository
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/summarize (Ver:3.02)	$REP/SUMMARIZE	Writes tables of data from many CAMDAT files

Where:

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

Analysis CRA14TP

CCDFGF

Table 113 .The run script files used were:

File	Path	Comment
RunControl/CCDFGF.py	$REP/CCDFGF	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/CCDFGF	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14TP

Table 114 .The input files used were:

File	Path	Comment
Input/ccgf_CRA14TP_control_r*i*.inp	$REP/CRA14TP/CCDFGF	Input file
Output/cusp_CRA14TP_r*i*.tbl	$REP/CRA14TP/CUTTINGS_S	Release table file
Output/epu_CRA14BL_ch.dat	$REP/CRA14BL/EPAUNI	Release table file
Output/epu_CRA14BL_rh.dat	$REP /CRA14BL/EPAUNI	Release table file
Input/gm_ccgf_CRA14TP.inp	$REP/CRA14TP/GENMESH	Input file
Input/intrusiontimes.in	$REP/CRA14TP/PRECCDFGF	Input file
Input/ms_ccgf_CRA14TP.inp	$REP/CRA14TP/MATSET	Input file
Output/sum_dbr_CRA14BL_r*i*_so_t*vvvvv*_L.tbl	$REP/CRA14BL/SUMMARIZE	Release table file
Output/sum_dbr_CRA14BL_r*i*_so_t*vvvvv*_M.tbl	$REP/CRA14BL/SUMMARIZE	Release table file
Output/sum_dbr_CRA14BL_r*i*_so_t*vvvvv*_U.tbl	$REP/CRA14BL/SUMMARIZE	Release table file
Output/sum_nut_CRA14BL_r*i*_so_t*uuuuu*.tbl	$REP/CRA14BL/SUMMARIZE	Release table file
Output/sum_panel_con_CRA14BL_b1_r*i*_s*n*.tbl	$REP/CRA14BL/SUMMARIZE	Release table file
Output/sum_panel_int_CRA14BL_b1_r*i*_s*p*_t*tttt*.tbl	$REP/CRA14BL/SUMMARIZE	Release table file
Output/sum_panel_st_CRA14BL_b1_r*i*_s*n*.tbl	$REP/CRA14BL/SUMMARIZE	Release table file
Output/sum_st2d_PABC09_r*i*_mf.tbl	$REP/PABC09/SUMMARIZE	Release table file
Output/sum_st2d_PABC09_r*i*_mp.tbl	$REP/PABC09/SUMMARIZE	Release table file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES
i is 1
n is 1-2
o is 1-5
p is 6
ttttt is 00100, 00350, 01000,02000,04000,
06000, 09000
uuuuu is 00100 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

Table 115 .The CVS repositories used were:

CVS Repositories
CCDFGF
CUTTINGS_S
EPAUNI
GENMESH
MATSET

POSTLHS
PRECCDFGF
SUMMARIZE

Table 116 .The log files used were:

File	Path	Comment
RunControl/CCDFGF.log	$REP/CCDFGF	log file
RunControl/CCDFGF.rtf	$REP/CCDFGF	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14TP

Table 117 .The output files produced were:

File	Path	Comment
Output/ccgf_CRA14TP_reltab_*ri*.dat	$REP/PRECCDFGF	CCDFGF Reults
Output/ccgf_CRA14TP_*ri*.out	$REP/CCDFGF	CCDFGF Reults
Output/gm_ccgf_CRA14TP.cdb	$PATH/GENMESH	CDB transfer file
Output/lhs3_ccgf_CRA14TP_*ri*_*vvvv*.cdb	$PATH/POSTLHS	LHS file
Output/ms_ccgf_CRA14TP.cdb	$PATH/MATSET	CDB transfer file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14TP
$PATH = /home/run_mast/GD/Analyses/CRA14TP/CCDFGF/RunCCDFGF
i is 1
vvv is 001-100

Table 118 .The executable files used were:

File	Path	Comment
Build/Solaris/ccdfgf (Ver:6.01)	$REP/CCDFGF	Constructs complimentary cumulative distribution functions for radionuclide releases
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/preccdfgf (Ver:2.01)	$REP/PRECCDFGF	Pre-processes data for ccdfgf

Where:

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

CUTTING_S

Table 119 .The run script files used were:

File	Path	Comment
RunControl/CUTTINGS_S.py	$REP/CUTTINGS_S	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/CUTTINGS_S	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14TP

Table 120 .The input files used were:

File	Path	Comment
Output/bf3_CRA14BL_r*i*_s*n*_v*vvv*.cdb	$PATH/BRAGFLO	
Input/cusp_CRA14TP.inp	$REP/CRA14TP/CUTTINGS_S	
Input/gm_cusp_CRA14TP.inp	$REP/CRA14TP/GENMESH	
Input/ms_cusp_CRA14TP.inp	$REP/CRA14TP/MATSET	
Output/mspall_drs_PABC09_r*i*.out	$REP/PABC09/DRSPALL	

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES
$PATH = /home/run_mast/GD/Analyses/CRA14BL/BRAGFLO/RunBRAGFLO
i is 1
n is 1-5
vvv is 001-100

Table 121 .The CVS repositories used were:

CVS Repositories
BRAGFLO
CUTTINGS_S
DRSPALL
GENMESH
MATSET
POSTLHS

Table 122 .The log files used were:

File	Path	Comment
RunControl/CUTTINGS_S.log	$REP/CUTTINGS_S	log file
RunControl/CUTTINGS_S.rtf	$REP/CUTTINGS_S	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14TP

Table 123 .The output files produced were:

File	Path	Comment
Output/cusp_CRA14TP_master_r*i*.inp	$REP/CUTTINGS_S	
Output/cusp_CRA14TP_r*i*.tbl	$REP/CUTTINGS_S	
Output/cusp_CRA14TP_r*i*_s*n*_t*tttt*_L_v*vvv*.cdb	$PATH/CUTTINGS_S	
Output/cusp_CRA14TP_r*i*_s*n*_t*tttt*_M_v*vvv*.cdb	$PATH/CUTTINGS_S	
Output/cusp_CRA14TP_r*i*_s*n*_t*tttt*_U_v*vvv*.cdb	$PATH/CUTTINGS_S	
Output/gm_cusp_CRA14TP.cdb	$PATH/GENMESH	CDB transfer file
Output/lhs3_cusp_CRA14TP_r*i*_v*vvv*.cdb	$PATH/POSTLHS	
Output/ms_cusp_CRA14TP.cdb	$PATH/MATSET	CDB transfer file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14TP
$PATH = /home/run_mast/GD/Analyses/CRA14TP/CUTTINGS_S/RunCUTTINGS_S
i is 1
n is 1-5
tttt 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

Table 124 .The executable files used were:

File	Path	Comment
Build/Solaris/cuttings_s (Ver:6.03)	$REP/CUTTINGS_S	Computes cuttings/spall generated by drilling
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements

Where:

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

Analysis CRA14BV

CCDFGF

Table 125 .The run script files used were:

File	Path	Comment
RunControl/CCDFGF.py	$REP/CCDFGF	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/CCDFGF	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BV

Table 126 .The input files used were:

File	Path	Comment
Input/ccgf_CRA14BV_control_*ri*.inp	$REP/CRA14BV/CCDFGF	Input file
Output/cusp_CRA14TP_*ri*.tbl	$PATH/CRA14TP/CUTTINGS_S/ RunCUTTINGS_S/CUTTINGS_S	Release table file
Output/epu_CRA14BL_ch.dat	$PATH/CRA14BL/EPAUNI/ RunEPAUNI/EPAUNI	Release table file
Output/epu_CRA14BL_rh.dat	$PATH/CRA14BL/EPAUNI/ RunEPAUNI/EPAUNI	Release table file
Input/gm_ccgf_CRA14BV.inp	$REP/CRA14BV/GENMESH	Input file
Input/intrusiontimes.in	$REP/CRA14BV/PRECCDFGF	Input file
Input/ms_ccgf_CRA14BV.inp	$REP/CRA14BV/MATSET	Input file
Output/sum_dbr_CRA14BL_*ri*_*so*_*tvvvvv*_L.tbl	$REP/CRA14BL/SUMMARIZE	Release table file
Output/sum_dbr_CRA14BL_*ri*_*so*_*tvvvvv*_M.tbl	$REP/CRA14BL/SUMMARIZE	Release table file
Output/sum_dbr_CRA14BL_*ri*_*so*_*tvvvvv*_U.tbl	$REP/CRA14BL/SUMMARIZE	Release table file
Output/sum_nut_CRA14BL_*ri*_*so*_*tuuuuu*.tbl	$REP/CRA14BL/SUMMARIZE	Release table file
Output/sum_panel_con_CRA14BV_b1_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_con_CRA14BV_b2_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_con_CRA14BV_b3_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_con_CRA14BV_b4_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_con_CRA14BV_b5_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_int_CRA14BV_b1_*ri*_*sp*_*ttttt*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_st_CRA14BV_b1_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_st_CRA14BV_b2_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_st_CRA14BV_b3_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_st_CRA14BV_b4_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_st_CRA14BV_b5_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_st2d_PABC09_*ri*_mf.tbl	$REP/PABC09/SUMMARIZE	Release table file
Output/sum_st2d_PABC09_*ri*_mp.tbl	$REP/PABC09/SUMMARIZE	Release table file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/
$PATH = /home/run_mast/GD/Analyses
i is 1
n is 1-2
o is 1-5
p is 6
ttttt is 00100, 00350, 01000, 02000, 04000,
06000, 09000
uuuuu is 00100 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

Table 127 .The CVS repositories used were:

CVS Repositories
CCDFGF
CUTTINGS_S
EPAUNI
GENMESH
MATSET
POSTLHS
PRECCDFGF
SUMMARIZE

Table 128 .The log files used were:

File	Path	Comment
RunControl/CCDFGF.log	$REP/CCDFGF	log file
RunControl/CCDFGF.rtf	$REP/CCDFGF	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BV

Table 129 .The output files produced were:

File	Path	Comment
Output/ccgf_CRA14BV_reltab_*ri*.dat	$REP/PRECCDFGF	CCDFGF Results
Output/ccgf_CRA14BV_*ri*.out	$REP/CCDFGF	CCDFGF Results
Output/gm_ccgf_CRA14BV.cdb	$PATH/GENMESH	CDB transfer file
Output/lhs3_ccgf_CRA14BV_*ri*_*vvvv*.cdb	$PATH/POSTLHS	LHS file
Output/ms_ccgf_CRA14BV.cdb	$PATH/MATSET	CDB transfer file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BV
$PATH = /home/run_mast/GD/Analyses/CRA14BV/CCDFGF/RunCCDFGF
i is 1
vvv is 001-100

Table 130 .The executable files used were:

File	Path	Comment
Build/Solaris/ccdfgf (Ver:6.01)	$REP/CCDFGF	Constructs complimentary cumulative distribution functions for radionuclide releases
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/preccdfgf (Ver:2.01)	$REP/PRECCDFGF	Pre-processes data for ccdfgf

Where:

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

PANEL

Table 131 .The run script files used were:

File	Path	Comment
RunControl/PANEL.py	$REP/PANEL	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/PANEL	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BV

Table 132 .The input files used were:

File	Path	Comment
Input/alg1_panel_CRA14BV.inp	$REP/ALGEBRACDB	Input file
Output/alg2_bf_CRA14BL_r*i*_s*n*_v*vvv*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Input/alg2_panel_CRA14BV_b1.inp	$REP/ALGEBRACDB	Input file
Input/alg2_panel_CRA14BV_b2.inp	$REP/ALGEBRACDB	Input file
Input/alg2_panel_CRA14BV_b3.inp	$REP/ALGEBRACDB	Input file
Input/alg2_panel_CRA14BV_b4.inp	$REP/ALGEBRACDB	Input file
Input/alg2_panel_CRA14BV_b5.inp	$REP/ALGEBRACDB	Input file
Input/alg3_panel_CRA14BV_b1.inp	$REP/ALGEBRACDB	Input file
Input/alg3_panel_CRA14BV_b2.inp	$REP/ALGEBRACDB	Input file
Input/alg3_panel_CRA14BV_b3.inp	$REP/ALGEBRACDB	Input file
Input/alg3_panel_CRA14BV_b4.inp	$REP/ALGEBRACDB	Input file
Input/alg3_panel_CRA14BV_b5.inp	$REP/ALGEBRACDB	Input file
Input/gm_panel_CRA14BV.inp	$REP/GENMESH	Input file
Input/ms_panel_CRA14BV.inp	$REP/MATSET	Input file
Input/sum_panel_con.inp	$REP/SUMMARIZE	Input file
Input/sum_panel_int.inp	$REP/SUMMARIZE	Input file
Input/sum_panel_st.inp	$REP/SUMMARIZE	Input file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BV
$PATH = /home/run_mast/GD/Analyses/CRA14BL/BRAGFLO/RunBRAGFLO
i is 1-3
n is 1-6
vvv is 001-100

Table 133 .The CVS repositories used were:

CVS Repositories
ALGEBRACDB
GENMESH
MATSET
PANEL
POSTLHS
SUMMARIZE

Table 134 .The log files used were:

File	Path	Comment
RunControl/PANEL.log	$REP/PANEL	log file
RunControl/PANEL.rtf	$REP/PANEL	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BV

Table 135 .The output files produced were:

File	Path	Comment
Output/alg1_panel_CRA14BV.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg2_panel_CRA14BV_b1.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg2_panel_CRA14BV_b2.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg2_panel_CRA14BV_b3.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg2_panel_CRA14BV_b4.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg2_panel_CRA14BV_b5.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg3_panel_CRA14BV_b1_*rj_vwww*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg3_panel_CRA14BV_b2_*rj_vwww*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg3_panel_CRA14BV_b3_*rj_vwww*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg3_panel_CRA14BV_b4_*rj_vwww*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg3_panel_CRA14BV_b5_*rj_vwww*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/gm_panel_CRA14BV.cdb	$PATH/GENMESH	CDB transfer file
Output/lhs3_panel_CRA14BV_b1_*rj*.xdbg		NOT SAVED:Debug file
Output/lhs3_panel_CRA14BV_b1_*rj_vwww*.cdb	$PATH/POSTLHS	CDB transfer file
Output/lhs3_panel_CRA14BV_b2_*rj*.xdbg		NOT SAVED:Debug file
Output/lhs3_panel_CRA14BV_b2_*rj_vwww*.cdb	$PATH/POSTLHS	CDB transfer file
Output/lhs3_panel_CRA14BV_b3_*rj*.xdbg		NOT SAVED:Debug file
Output/lhs3_panel_CRA14BV_b3_*rj_vwww*.cdb	$PATH/POSTLHS	CDB transfer file
Output/lhs3_panel_CRA14BV_b4_*rj*.xdbg		NOT SAVED:Debug file
Output/lhs3_panel_CRA14BV_b4_*rj_vwww*.cdb	$PATH/POSTLHS	CDB transfer file
Output/lhs3_panel_CRA14BV_b5_*rj*.xdbg		NOT SAVED:Debug file
Output/lhs3_panel_CRA14BV_b5_*rj_vwww*.cdb	$PATH/POSTLHS	CDB transfer file
Output/ms_panel_CRA14BV.cdb	$PATH/MATSET	CDB transfer file
Output/panel_con_CRA14BV_b1_*rj_sp_vwww*.cdb	$PATH/PANEL	CDB transfer file
Output/panel_con_CRA14BV_b2_*rj_sp_vwww*.cdb	$PATH/PANEL	CDB transfer file
Output/panel_con_CRA14BV_b3_*rj_sp_vwww*.cdb	$PATH/PANEL	CDB transfer file
Output/panel_con_CRA14BV_b4_*rj_sp_vwww*.cdb	$PATH/PANEL	CDB transfer file
Output/panel_con_CRA14BV_b5_*rj_sp_vwww*.cdb	$PATH/PANEL	CDB transfer file
Output/panel_decay_CRA14BV_*ri_sn_vvvv*.cdb	$PATH/PANEL	CDB transfer file
Output/panel_int_CRA14BV_b1_*rj_sq_ttttt_vwww*.cdb	$PATH/PANEL	CDB transfer file
Output/panel_int_CRA14BV_b2_*rj_sq_ttttt_vwww*.cdb	$PATH/PANEL	CDB transfer file
Output/panel_int_CRA14BV_b3_*rj_sq_ttttt_vwww*.cdb	$PATH/PANEL	CDB transfer file
Output/panel_int_CRA14BV_b4_*rj_sq_ttttt_vwww*.cdb	$PATH/PANEL	CDB transfer file
Output/panel_int_CRA14BV_b5_*rj_sq_ttttt_vwww*.cdb	$PATH/PANEL	CDB transfer file
Output/sum_panel_con_CRA14BV_b1_*rj_so*.tbl	$REP/SUMMARIZE	Table file
Output/sum_panel_con_CRA14BV_b2_*rj_so*.tbl	$REP/SUMMARIZE	Table file
Output/sum_panel_con_CRA14BV_b3_*rj_so*.tbl	$REP/SUMMARIZE	Table file
Output/sum_panel_con_CRA14BV_b4_*rj_so*.tbl	$REP/SUMMARIZE	Table file
Output/sum_panel_con_CRA14BV_b5_*rj_so*.tbl	$REP/SUMMARIZE	Table file
Output/sum_panel_int_CRA14BV_b1_*rj_sq_ttttt*.tbl	$REP/SUMMARIZE	Table file
Output/sum_panel_int_CRA14BV_b2_*rj_sq_ttttt*.tbl	$REP/SUMMARIZE	Table file
Output/sum_panel_int_CRA14BV_b3_*rj_sq_ttttt*.tbl	$REP/SUMMARIZE	Table file
Output/sum_panel_int_CRA14BV_b4_*rj_sq_ttttt*.tbl	$REP/SUMMARIZE	Table file
Output/sum_panel_int_CRA14BV_b5_*rj_sq_ttttt*.tbl	$REP/SUMMARIZE	Table file
Output/sum_panel_st_CRA14BV_b1_*rj_so*.tbl	$REP/SUMMARIZE	Table file
Output/sum_panel_st_CRA14BV_b2_*rj_so*.tbl	$REP/SUMMARIZE	Table file

File	Path	Comment
Output/sum_panel_st_CRA14BV_b3_r*j*_s*o*.tbl	$REP/SUMMARIZE	Table file
Output/sum_panel_st_CRA14BV_b4_r*j*_s*o*.tbl	$REP/SUMMARIZE	Table file
Output/sum_panel_st_CRA14BV_b5_r*j*_s*o*.tbl	$REP/SUMMARIZE	Table file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14BV
$PATH = /home/run_mast/GD/Analyses/CRA14BV/PANEL/RunPANEL
i is 1
j is 1-3
n is 1
o is 1-2
p is 1-6
q is 6
ttttt is 00100, 00350, 01000, 02000, 04000, 06000, 09000
vvv is 001
www is 001-100

Table 136 .The executable files used were:

File	Path	Comment
Build/Solaris/algebracdb (Ver:2.36)	$REP/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/panel (Ver:4.04)	$REP/PANEL	Computes release concentrations of nuclides from repository
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/summarize (Ver:3.02)	$REP/SUMMARIZE	Writes tables of data from many CAMDAT files

Where:

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

Analysis CRA14-0

BRAGFLO

Table 137 .The run script files used were:

File	Path	Comment
RunControl/BRAGFLO.py	$REP/BRAGFLO	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/BRAGFLO	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14

Table 138 .The input files used were:

File	Path	Comment
Input/alg1_bf_CRA14.inp	$REP/ALGEBRACDB	Input file
Input/alg2_bf_CRA14.inp	$REP/ALGEBRACDB	Input file
Input/bf1_CRA14_s*n*.inp	$REP/PREBRAG	Input file
Input/bf1_CRA14_s*n*_mod1.inp	$REP/PREBRAG	Input file
Input/bf1_CRA14_s*n*_mod2.inp	$REP/PREBRAG	Input file
Input/bf2_CRA14_closure.dat	$REP/BRAGFLO	Input file
Input/gm_bf_CRA14.inp	$REP/GENMESH	Input file
Input/ic_bf_CRA14.inp	$REP/ICSET	Input file
Input/ms_bf_CRA14.inp	$REP/MATSET	Input file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14

n is 1-6

Table 139 .The CVS repositories used were:

CVS Repositories
ALGEBRACDB
BRAGFLO
GENMESH
ICSET
MATSET
POSTBRAG
POSTLHS
PREBRAG

Table 140 .The log files used were:

File	Path	Comment
RunControl/BRAGFLO.log	$REP/BRAGFLO	log file
RunControl/BRAGFLO.rtf	$REP/BRAGFLO	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14

Table 141 .The output files produced were:

File	Path	Comment
Output/alg1_bf_CRA14_r*i*_v*vvv*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg2_bf_CRA14_r*i*_s*n*_v*vvv*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/bf2_CRA14_r*i*_s*n*_v*vvv*.inp	$REP/PREBRAG	BRAGFLO input file
Output/bf2_CRA14_r*i*_s*n*_v*vvv*.log	$REP/BRAGFLO	Logfile
Output/bf2_CRA14_r*i*_s*n*_v*vvv*.sum	$REP/BRAGFLO	Summary file
Output/bf3_CRA14_r*i*_s*n*_v*vvv*.cdb	$PATH/BRAGFLO	CDB transfer file
Output/gm_bf_CRA14.cdb	$PATH/GENMESH	CDB transfer file
Output/ic_bf_CRA14_r*i*_v*vvv*.cdb	$PATH/ICSET	CDB transfer file
Output/lhs3_bf_CRA14_r*i*_v*vvv*.cdb	$PATH/POSTLHS	CDB transfer file
Output/ms_bf_CRA14.cdb	$PATH/MATSET	CDB transfer file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14
$PATH = /home/run_mast/GD/Analyses/CRA14/BRAGFLO/RunBRAGFLO
i is 1-3
n is 1-6
vvv is 001-100

Table 142 .The executable files used were:

File	Path	Comment
Build/Solaris/algebracdb (Ver:2.36)	$REP/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Solaris/bragflo (Ver:6.03)	$REP/BRAGFLO	Computes brine and gas flow in the repository
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/icset (Ver:2.23)	$REP/ICSET	Assigns initial conditions to the CAMDAT grid elements
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postbrag (Ver:4.02)	$REP/POSTBRAG	Post-processes data for bragflo
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/prebrag (Ver:8.03)	$REP/PREBRAG	Pre-processes data for bragflo

Where:

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

BRAGFLO_DBR

Table 143 .The run script files used were:

File	Path	Comment
RunControl/BRAGFLO_DBR.py	$REP/BRAGFLO_DBR	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/BRAGFLO_DBR	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14

Table 144 .The input files used were:

File	Path	Comment

Input/alg1_dbr_CRA14.inp	$REP/ALGEBRACDB	
Input/alg2_dbr_CRA14_*so*.inp	$REP/ALGEBRACDB	
Input/alg3_dbr_CRA14_L.inp	$REP/ALGEBRACDB	
Input/alg3_dbr_CRA14_M.inp	$REP/ALGEBRACDB	
Input/alg3_dbr_CRA14_U.inp	$REP/ALGEBRACDB	
Input/bf1_dbr_CRA14_L.inp	$REP/PREBRAG	
Input/bf1_dbr_CRA14_M.inp	$REP/PREBRAG	
Input/bf1_dbr_CRA14_*sn*_100_L.inp	$REP/PREBRAG	
Input/bf1_dbr_CRA14_*sn*_100_M.inp	$REP/PREBRAG	
Input/bf1_dbr_CRA14_*sn*_100_U.inp	$REP/PREBRAG	
Input/bf1_dbr_CRA14_U.inp	$REP/PREBRAG	
Output/bf3_CRA14_*ri*_*so*_*vvvv*.cdb	$PATH/BRAGFLO/RunBRAGFLO/BRAGFLO	
Output/cusp_CRA14_*ri*_*so*_*ttttt*_L_*vvvv*.cdb	$PATH/CUTTINGS_S/RunCUTTINGS_S/CUTTINGS_S	
Output/cusp_CRA14_*ri*_*so*_*ttttt*_M_*vvvv*.cdb	$PATH/CUTTINGS_S/RunCUTTINGS_S/CUTTINGS_S	
Output/cusp_CRA14_*ri*_*so*_*ttttt*_U_*vvvv*.cdb	$PATH/CUTTINGS_S/RunCUTTINGS_S/CUTTINGS_S	
Input/gm_dbr_CRA14.inp	$REP/GENMESH	
Input/ic_dbr_CRA14_*so*.inp	$REP/ICSET	
Input/ms_dbr_CRA14.inp	$REP/MATSET	
Input/rel1_dbr_CRA14.inp	$REP/RELATE	
Input/rel2_dbr_CRA14_*so*.inp	$REP/RELATE	
Input/sum_dbr.inp	$REP/SUMMARIZE	

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14
$PATH = /home/run_mast/GD/Analyses/CRA14
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

Table 145 .The CVS repositories used were:

CVS Repositories
ALGEBRACDB
BRAGFLO
BRAGFLO_DBR
CUTTINGS_S
GENMESH
ICSET
MATSET
POSTBRAG
POSTLHS
PREBRAG
RELATE
SUMMARIZE

Table 146 .The log files used were:

File	Path	Comment
RunControl/BRAGFLO_DBR.log	$REP/BRAGFLO_DBR	log file
RunControl/BRAGFLO_DBR.rtf	$REP/BRAGFLO_DBR	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14

Table 147 .The output files produced were:

File	Path	Comment
Output/alg1_dbr_CRA14_r*j*_s*o*_t*uuuuu*_v*vvv*.cdb	$PATH/ALGEBRACDB	
Output/alg2_dbr_CRA14_r*j*_s*o*_t*uuuuu*_v*vvv*.cdb	$PATH/ALGEBRACDB	
Output/alg3_dbr_CRA14_r*j*_s*o*_t*uuuuu*_L_v*vvv*.cdb	$PATH/ALGEBRACDB	
Output/alg3_dbr_CRA14_r*j*_s*o*_t*uuuuu*_M_v*vvv*.cdb	$PATH/ALGEBRACDB	
Output/alg3_dbr_CRA14_r*j*_s*o*_t*uuuuu*_U_v*vvv*.cdb	$PATH/ALGEBRACDB	
Output/bf2_dbr_CRA14_r*j*_s*o*_t*uuuuu*_L_v*vvv*.inp	$REP/BRAGFLO_DBR	
Output/bf2_dbr_CRA14_r*j*_s*o*_t*uuuuu*_M_v*vvv*.inp	$REP/BRAGFLO_DBR	
Output/bf2_dbr_CRA14_r*j*_s*o*_t*uuuuu*_U_v*vvv*.inp	$REP/BRAGFLO_DBR	
Output/bf3_dbr_CRA14_r*j*_s*o*_t*uuuuu*_L_v*vvv*.cdb	$PATH/BRAGFLO_DBR	
Output/bf3_dbr_CRA14_r*j*_s*o*_t*uuuuu*_M_v*vvv*.cdb	$PATH/BRAGFLO_DBR	
Output/bf3_dbr_CRA14_r*j*_s*o*_t*uuuuu*_U_v*vvv*.cdb	$PATH/BRAGFLO_DBR	
Output/gm_dbr_CRA14.cdb	$PATH/GENMESH	
Output/ic_dbr_CRA14_r*j*_s*o*_t*uuuuu*_v*vvv*.cdb	$PATH/ICSET	
Output/ms_dbr_CRA14.cdb	$PATH/MATSET	
Output/rel1_dbr_CRA14_r*j*_s*o*_t*uuuuu*_v*vvv*.cdb	$PATH/RELATE	
Output/rel2_dbr_CRA14_r*j*_s*o*_t*uuuuu*_v*vvv*.cdb	$PATH/RELATE	
Output/sum_dbr_CRA14_r*j*_s*o*_t*uuuuu*_L.tbl	$REP/SUMMARIZE	
Output/sum_dbr_CRA14_r*j*_s*o*t*uuuuu*_M.tbl	$REP/SUMMARIZE	
Output/sum_dbr_CRA14_r*j*_s*o*_t*uuuuu*_U.tbl	$REP/SUMMARIZE	

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14
$PATH = /home/run_mast/GD/Analyses/CRA14/BRAGFLO_DBR/
RunBRAGFLO_DBR
j is 1-3
o is 1-5
uuuuu 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

Table 148 .The executable files used were:

File	Path	Comment
Build/Solaris/algebracdb (Ver:2.36)	$REP/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Solaris/bragflo (Ver:6.03)	$REP/BRAGFLO	Computes brine and gas flow in the repository
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/icset (Ver:2.23)	$REP/ICSET	Assigns initial conditions to the CAMDAT grid elements
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postbrag	$REP/POSTBRAG	Post-processes data for bragflo

File	Path	Comment
(Ver:4.02)		
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/prebrag (Ver:8.03)	$REP/PREBRAG	Pre-processes data for bragflo
Build/Solaris/relate (Ver:1.45)	$REP/RELATE	Transfers CAMDAT data to another CAMDAT file
Build/Solaris/summarize (Ver:3.02)	$REP/SUMMARIZE	Writes tables of data from many CAMDAT files

Where:

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

CCDFGF

Table 149 .The run script files used were:

File	Path	Comment
RunControl/CCDFGF.py	$REP/CCDFGF	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/CCDFGF	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14

Table 150 .The input files used were:

File	Path	Comment
Input/ccgf_CRA14_control_*ri*.inp	$REP/CRA14/CCDFGF	Input file
Output/cusp_CRA14_*ri*.tbl	$REP/CRA14/CUTTINGS_S	Release table file
Output/epu_CRA14BL_ch.dat	$REP/CRA14BL/EPAUNI	Release table file
Output/epu_CRA14BL_rh.dat	$REP/CRA14BL/EPAUNI	Release table file
Input/gm_ccgf_CRA14.inp	$REP/CRA14/GENMESH	Input file
Input/intrusiontimes.in	$REP/CRA14/PRECCDFGF	Input file
Input/ms_ccgf_CRA14.inp	$REP/CRA14/MATSET	Input file
Output/sum_dbr_CRA14_*ri*_*so*_*tvvvvv*_L.tbl	$REP/CRA14/SUMMARIZE	Release table file
Output/sum_dbr_CRA14_*ri*_*so*_*tvvvvv*_M.tbl	$REP/CRA14/SUMMARIZE	Release table file
Output/sum_dbr_CRA14_*ri*_*so*_*tvvvvv*_U.tbl	$REP/CRA14/SUMMARIZE	Release table file
Output/sum_nut_CRA14_*ri*_*so*_*tuuuuu*.tbl	$REP/CRA14/SUMMARIZE	Release table file
Output/sum_panel_con_CRA14BV_b1_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_con_CRA14BV_b2_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_con_CRA14BV_b3_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_con_CRA14BV_b4_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_con_CRA14BV_b5_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_int_CRA14BV_b1_*ri*_*sp*_*ttttt*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_int_CRA14BV_b2_*ri*_*sp*_*ttttt*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_int_CRA14BV_b3_*ri*_*sp*_*ttttt*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_int_CRA14BV_b4_*ri*_*sp*_*ttttt*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_int_CRA14BV_b5_*ri*_*sp*_*ttttt*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_st_CRA14BV_b1_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_st_CRA14BV_b2_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_st_CRA14BV_b3_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_st_CRA14BV_b4_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_panel_st_CRA14BV_b5_*ri*_*sn*.tbl	$REP/CRA14BV/SUMMARIZE	Release table file
Output/sum_st2d_PABC09_*ri*_mf.tbl	$REP/PABC09/SUMMARIZE	Release table file
Output/sum_st2d_PABC09_*ri*_mp.tbl	$REP/PABC09/SUMMARIZE	Release table file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES
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 00100 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

Table 151 .The CVS repositories used were:

CVS Repositories
CCDFGF
CUTTINGS_S
EPAUNI
GENMESH
MATSET
POSTLHS
PRECCDFGF
SUMMARIZE

Table 152 .The log files used were:

File	Path	Comment
RunControl/CCDFGF.log	$REP/CCDFGF	log file
RunControl/CCDFGF.rtf	$REP/CCDFGF	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14

Table 153 .The output files produced were:

File	Path	Comment
Output/ccgf_CRA14_reltab_*ri*.dat	$REP/PRECCDFGF	CCDFGF Results
Output/ccgf_CRA14_*ri*.out	$REP/CCDFGF	CCDFGF Results
Output/gm_ccgf_CRA14.cdb	$PATH/GENMESH	CDB transfer file
Output/lhs3_ccgf_CRA14_*ri*_*vvvv*.cdb	$PATH/POSTLHS	LHS file
Output/ms_ccgf_CRA14.cdb	$PATH/MATSET	CDB transfer file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14
$PATH = /home/run_mast/GD/Analyses/CRA14/CCDFGF/RunCCDFGF
i is 1-3
v is 001-100

Table 154 .The executable files used were:

File	Path	Comment
Build/Solaris/ccdfgf (Ver:6.01)	$REP/CCDFGF	Constructs complimentary cumulative distribution functions for radionuclide releases
Build/Solaris/genmesh	$REP/GENMESH	Generates the CAMDAT computational grid

(Ver:6.09)		
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements
Build/Solaris/preccdfgf (Ver:2.01)	$REP/PRECCDFGF	Pre-processes data for ccdfgf

Where:

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

CUTTINGS_S

Table 155 .The run script files used were:

File	Path	Comment
RunControl/CUTTINGS_S.py	$REP/CUTTINGS_S	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/CUTTINGS_S	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14

Table 156 .The input files used were:

File	Path	Comment
Output/bf3_CRA14_r*i*_s*n*_v*vvv*.cdb	$PATH/BRAGFLO	
Input/cusp_CRA14.inp	$REP/CRA14/CUTTINGS_S	
Input/gm_cusp_CRA14.inp	$REP/CRA14/GENMESH	
Input/ms_cusp_CRA14.inp	$REP/CRA14/MATSET	
Output/mspall_drs_PABC09_r*i*.out	$REP/PABC09/DRSPALL	

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES
$PATH = /home/run_mast/GD/Analyses/CRA14/BRAGFLO/RunBRAGFLO
i is 1-3
n is 1-5
vvv is 001-100

Table 157 .The CVS repositories used were:

CVS Repositories
BRAGFLO
CUTTINGS_S
DRSPALL
GENMESH
MATSET
POSTLHS

Table 158 .The log files used were:

File	Path	Comment
RunControl/CUTTINGS_S.log	$REP/CUTTINGS_S	log file
RunControl/CUTTINGS_S.rtf	$REP/CUTTINGS_S	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14

Table 159 .The output files produced were:

File	Path	Comment
Output/cusp_CRA14_master_*ri*.inp	$REP/CUTTINGS_S	
Output/cusp_CRA14_*ri*.tbl	$REP/CUTTINGS_S	
Output/cusp_CRA14_*ri*_*sn*_*ttttt*_L_*vvvv*.cdb	$PATH/CUTTINGS_S	
Output/cusp_CRA14_*ri*_*sn*_*ttttt*_M_*vvvv*.cdb	$PATH/CUTTINGS_S	
Output/cusp_CRA14_*ri*_*sn*_*ttttt*_U_*vvvv*.cdb	$PATH/CUTTINGS_S	
Output/gm_cusp_CRA14.cdb	$PATH/GENMESH	CDB transfer file
Output/lhs3_cusp_CRA14_*ri*_*vvvv*.cdb	$PATH/POSTLHS	
Output/ms_cusp_CRA14.cdb	$PATH/MATSET	CDB transfer file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14
$PATH=/home/run_mast/GD/Analyses/CRA14/CUTTINGS_S/RunCUTTINGS_S
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

Table 160 .The executable files used were:

File	Path	Comment
Build/Solaris/cuttings_s (Ver:6.03)	$REP/CUTTINGS_S	Computes cuttings/spall generated by drilling
Build/Solaris/genmesh (Ver:6.09)	$REP/GENMESH	Generates the CAMDAT computational grid
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/postlhs (Ver:4.08)	$REP/POSTLHS	Assigns sampled parameters to the grid blocks and elements

Where:

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

LHS

Table 161 .The run script files used were:

File	Path	Comment
RunControl/LHS.py	$REP/LHS	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/LHS	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14

Table 162 .The input file used was:

File	Path	Comment
Input/lhs1_CRA14_*ri*_con.inp	$REP/PRELHS	PRELHS input file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14
i is 1-3

Table 163 .The CVS repositories used were:

CVS Repositories
LHS
PRELHS

Table 164 .The log files used were:

File	Path	Comment
RunControl/LHS.log	$REP/LHS	log file
RunControl/LHS.rtf	$REP/LHS	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14

Table 165 .The output files produced were:

File	Path	Comment
Output/lhs1_CRA14_r*i*_con.dbg	$REP/PRELHS	PRELHS debug file
Output/lhs1_CRA14_r*i*_con.trn	$REP/PRELHS	PRELHS transfer file
Output/lhs2_CRA14_r*i*_con.dbg	$REP/LHS	LHS debug file
Output/lhs2_CRA14_r*i*_con.trn	$REP/LHS	LHS transfer file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14

i is 1-3

Table 166 .The executable files used were:

File	Path	Comment
Build/Solaris/lhs (Ver:2.43)	$REP/LHS	Code to sample uncertain parameters
Build/Solaris/prelhs (Ver:2.41)	$REP/PRELHS	Pre-processes data for lhs

Where:

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

NUTS

Table 167 .The run script files used were:

File	Path	Comment
RunControl/NUTS.py	$REP/NUTS	Python run control script
RunControl//home/run_mast/GD/Run.py	$REP/NUTS	Main control script

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14

Table 168 .The input files used were:

File	Path	Comment
Input/alg_nut_iso_CRA14.inp	$REP/ALGEBRACDB	Input file
Input/alg_nut_scn_CRA14.inp	$REP/ALGEBRACDB	Input file
Output/bf2_CRA14_r*i*_s*m*_v*vvv*.inp	$REP/PREBRAG	Input file
Output/bf3_CRA14_r*i*_s*m*_v*vvv*.cdb	$PATH/CRA14/BRAGFLO/RunBRAGFLO/BRAGFLO	CDB transfer file
Input/ms_nut_CRA14.inp	$REP/MATSET	Input file
Input/nut_int_CRA14_s*o*_t*ttttt*.inp	$REP/NUTS	Input file

Input/nut_iso_CRA14_*sn*.inp	$REP/NUTS	Input file
Input/nut_scn_CRA14_*sn*.inp	$REP/NUTS	Input file
Output/panel_con_CRA14BV_b1_*ri*_*sm*_*vvvv*.cdb	$PATH/CRA14BV/PANEL/RunPANEL/PANEL	CDB transfer file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14
$PATH = /home/run_mast/GD/Analyses
i is 1-3
m is 1-6
n is 1-5
o is 2-5
tttt is 00100 for S2, S4
03000, 05000, 07000, 09000 for S3, S5
vvv is 001-100

Table 169 .The CVS repositories used were:

CVS Repositories
ALGEBRACDB
BRAGFLO
MATSET
NUTS
PANEL
PREBRAG
SCREEN_NUTS
SUMMARIZE

Table 170 .The log files used were:

File	Path	Comment
RunControl/NUTS.log	$REP/NUTS	log file
RunControl/NUTS.rtf	$REP/NUTS	Formatted log file (Word file)

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14

Table 171 .The output files produced were:

File	Path	Comment
Output/alg_nut_int_CRA14_*ri*_*so*_*tuuuuu*_*VVVV*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg_nut_iso_CRA14_*ri*_*sn*_*VVVV*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/alg_nut_scn_CRA14_*ri*_*sn*_*vvvv*.cdb	$PATH/ALGEBRACDB	CDB transfer file
Output/ms_nut_CRA14_*ri*_*sn*_*VVVV*.cdb	$PATH/MATSET	CDB transfer file
Output/nut_int_CRA14_*ri*_*sn*_*tuuuuu*_*VVVV*.cdb	$PATH/NUTS	CDB transfer file
Output/nut_iso_CRA14_*ri*_*sn*_*VVVV*.cdb		NOT SAVED:CDB transfer file
Output/nut_scn_CRA14_*ri*_*sn*_*vvvv*.cdb	$PATH/NUTS	CDB transfer file
Output/screen_nut_scn_CRA14_*ri*_EDIT.inp	$REP/SCREEN_NUTS	Input file
Output/screen_nut_scn_CRA14_*ri*_*sn*.out	$REP/SCREEN_NUTS	Output file
Output/sum_nut_CRA14_*ri*_*sn*_*ttttt*.tbl		NOT SAVED:Table file
Output/sum_nut_scn_CRA14_*ri*_*sn*.tbl	$REP/SUMMARIZE	Table file

Where:

$REP = /nfs/data/CVSLIB/WIPP_ANALYSES/CRA14
$PATH = /home/run_mast/GD/Analyses/CRA14/NUTS/RunNUTS
i is 1-3
n is 1-5
o is 2-5
ttttt is 00100 for S1
00100, 00350 for S2, S4
01000, 03000, 05000, 07000, 09000 for S3, S5
uuuuu is 00100 for S2, S4
03000, 05000, 07000, 09000 for S3, S5
vvv is 001-100
VVV are the screened-in vectors listed in Table 172.

Table 172 .The screened-in vectors were:

Replicate	Scenario	Vectors
1	1	1,2,3,5,6,7,8,9,10,11,12,13,14,16,17,19,20,22,23,24,25,26,27,28,29,30,31,33,34,35,36,37,38,39,41,43,44,45,46,47,48,49,50,51,52,53,54,55,58,59,60,61,62,63,64,66,67,68,69,70,71,72,74,75,76,77,78,79,80,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,100
1	2	1,2,3,5,6,7,8,9,10,11,12,13,14,16,17,19,20,22,23,24,25,26,27,28,29,30,31,33,34,35,36,37,38,39,41,43,44,45,46,47,48,49,50,51,52,53,54,55,58,59,60,61,62,63,64,66,67,68,69,70,71,72,74,75,76,77,78,79,80,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,100
1	3	1,2,3,5,6,7,8,9,11,12,13,14,16,17,19,20,22,23,24,25,26,27,28,29,30,33,34,35,36,37,38,39,41,43,44,45,46,47,48,49,50,51,52,54,55,58,59,60,61,62,63,64,66,67,69,70,71,72,74,75,76,77,78,79,80,82,83,84,86,88,89,90,92,93,94,95,96,97,98,100
1	4	2,7,9,12,16,17,20,27,28,30,36,45,50,63,66,67,76,78,98
1	5	2,7,9,12,16,17,27,28,30,36,45,50,63,66,67,76,78,98
2	1	1,2,3,4,6,7,8,9,10,11,12,13,14,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,43,44,45,46,47,48,49,50,51,52,53,54,55,56,59,61,62,63,65,66,67,68,69,70,71,72,73,74,75,77,79,80,81,82,83,84,86,87,88,89,90,92,93,94,95,96,98,99,100
2	2	1,2,3,4,6,7,8,9,10,11,12,13,14,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,43,44,45,46,47,48,49,50,51,52,53,54,55,56,59,61,62,63,65,66,67,68,69,70,71,72,73,74,75,77,79,80,81,82,83,84,86,87,88,89,90,92,93,94,95,96,98,99,100
2	3	1,2,3,4,6,8,9,10,11,12,13,14,16,17,18,19,20,21,22,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,43,44,45,46,47,48,49,50,51,52,53,54,55,56,59,61,62,63,65,66,67,68,70,71,72,74,75,77,79,80,81,83,84,86,87,89,90,92,94,95,96,98,99,100
2	4	4,17,21,24,25,28,30,34,36,40,43,53,55,59,63,67,68,79,90,92,95,96,98
2	5	4,17,21,24,25,28,30,34,36,40,43,53,55,59,63,67,68,79,90,92,95,96,98
3	1	2,3,5,6,7,8,9,10,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,77,78,79,81,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100
3	2	2,3,5,6,7,8,9,10,11,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,77,78,79,81,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100
3	3	2,3,5,7,8,10,11,13,14,15,17,18,20,21,22,24,25,26,27,28,29,30,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,49,50,51,52,53,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,73,74,75,77,78,79,81,84,85,86,88,89,90,91,93,94,95,96,97,98,99,100
3	4	30,35,37,40,42,44,47,49,53,59,61,63,66,69,77,86,91,93,96,97
3	5	30,35,40,42,44,47,49,53,59,63,66,69,77,86,93,96

Table 173 .The executable files used were:

File	Path	Comment
Build/Solaris/algebracdb (Ver:2.36)	$REP/ALGEBRACDB	Manipulates CAMDAT data by evaluating algebraic expressions
Build/Solaris/matset (Ver:9.21)	$REP/MATSET	Assigns material properties to CAMDAT grid blocks
Build/Solaris/nuts (Ver:2.06)	$REP/NUTS	Nuclide Transport system model
Build/Solaris/screen_nuts (Ver:1.01)	$REP/SCREEN_NUTS	
Build/Solaris/summarize (Ver:3.02)	$REP/SUMMARIZE	Writes tables of data from many CAMDAT files

Where:

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