EPA Comment (Received during EPA Run Control Inspection for the Week of August 9, 2004):

Investigate how CCDFGF and SUMMARIZE are checked/verified/tested for capturing the correct code CDB data streams.

Original DOE Response

Dunagan (2004) describes the procedure used to verify that SUMMARIZE and CCDFGF have captured the correct code Computational Database (CDB) data streams. The actual processing of CDB data streams is verified in advance because the codes which manipulate CDB data streams have been validated and verified for this purpose in accordance with Sandia National Laboratories (SNL) software quality assurance procedure NP 19-1. The selection of CDB data streams to process is controlled by scripts that are part of the Performance Assessment Run Control System (PA RCS). Verifying that these scripts choose the appropriate data streams is accomplished by comparing the script input files with PA RCS log files.

Addendum to DOE Response

The purpose of this addendum is to report on additional work (Kanney and Kirchner 2005) to verify that data from process model CDB files is being transferred across the SUMMARIZE interface with sufficient accuracy.

During their review of the 2004 Compliance Recertification Application (CRA-2004), EPA requested information regarding how data transfer between process model codes and CCDFGF was verified. DOE responded to this request by describing a procedure for inspecting WIPP PA Run Control log files to verify that the correct files were fetched from CMS libraries and used by SUMMARIZE (Dunagan 2004, Piper 2004). Subsequent to the response, several problems with data transfer across the SUMMARIZE interface were discovered, corrected and reported. The problems encountered fell into two categories:

- Certain releases were miscalculated because information in certain SUMMARIZE output tables was not in the column that PRECCDFGF expected.
- Releases to the Culebra were under-estimated in several cases because SUMMARIZE wrote zeros to the output table when it could not locate an input CDB file.

The impact of the first class of errors on the results and conclusions of the CRA-2004 were determined to be minimal (Vugrin 2004). An informal review of the second type of error indicates that the impact is minimal (a formal review is in progress). Regardless, it was recognized that the SUMMARIZE interface needed to be improved in future PAs.

Both classes of problems outlined can be traced back to errors in the SUMMARIZE input files, not with the SUMMARIZE code itself. Nearly 600 SUMMARIZE input files were used in the CRA-2004. Since each one was prepared manually, the probability of errors was considerable. However, it was determined that improving the SUMMARIZE code was an important part of improving the interface between process model codes and CCDF construction codes. The following steps were followed to improve the SUMMARIZE interface:

- A PRECCDFGF driver was added to the set of output drivers in SUMMARIZE. This driver writes header information at the top of the output table. The header information identifies the data that appears in the rest of the table.

  The PRECCDFGF code was modified to read the header information produced by SUMMARIZE to verify that the data in the input file is in the order required by PRECCDFGF.
- The SUMMARIZE code was modified to incorporate strict checking of the vector list, removing the possibility that vectors are inadvertently skipped due to errors in CDB file specification.

  The WIPP PA Run Control scripts were re-written to automate the generation of the SUMMARIZE input files and eliminate errors introduced during manual editing of the files.

The steps outlined above were implemented (following all applicable software quality assurance procedures) and used in the 2004 Compliance Recertification Performance Assessment Baseline Calculation (CRA-2004 PABC, Kanney and Leigh (2005)).

In order to demonstrate that the improvements made to the SUMMARIZE interface have eliminated the errors discussed above, a verification study has been performed. The procedures and results of the verification of the SUMMARIZE interface in the CRA-2004 PABC are documented in Kanney and Kirchner (2005). This report verifies that data from process model CDB files is now being transferred across the SUMMARIZE interface with sufficient accuracy.

References

Dunagan, S. 2004. Explanation of how SUMMARIZE and CCDFGF are checked/verified/tested for capturing the correct CDB data streams in the WIPP CRA-2004 Performance Assessment. Technical Memorandum ERMS 536767, Sandia National Laboratories, Carlsbad, NM.

Kanney, J. F., and T. B. Kirchner. 2005. Verification of the SUMMARIZE Interface in the CRA-2004 Performance Assessment Baseline Calculation. ERMS 540977, Sandia National Laboratories, Carlsbad, NM.

Kanney, J. F., and C. Leigh. 2005. Analysis Plan for Post CRA PA Baseline Calculation (AP-122). ERMS 539624, Sandia National Laboratories, Carlsbad, NM.

Piper, L. 2004. 4th Response Submittal to EPA. U.S. Department of Energy, Carlsbad, NM.

Vugrin, E. D. 2004. Corrected CRA Figures. Technical Memorandum ERMS 538260, Sandia National Laboratories, Carlsbad, NM.

540977

Sandia National Laboratories
Waste Isolation Pilot Plant

Verification of the SUMMARIZE Interface in the CRA-2004 Performance Assessment Baseline Calculation

Task number
1.4.1.3

Author:	Joseph F. Kanney (6821) Print	Signature	9/16/05 Date
Author:	Thomas B. Kirchner (6821) Print	Thomas B Kirchner Signature	9/16/05 Date
Technical Review:	Eric D. Vugrin (6821) Print	Signature	9/16/05 Date
QA Review:	Mario J. Chavez (6820) Print	Mario Chavez Signature	9/16/05 Date
Management Review:	David S. Kessel (6821) Print	David Kessel Signature	9/16/05 Date

THIS PAGE IS INTENTIONALLY BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

The Waste Isolation Pilot Plant (WIPP) is a deep geologic repository developed by the U.S. Department of Energy (DOE) for the disposal of transuranic (TRU) radioactive waste. Containment of TRU waste at the WIPP is regulated by the U.S. Environmental Protection Agency (EPA) according to the regulations set forth in Title 40 of the Code of Federal Regulations (CFR), Parts 191 and 194. The DOE demonstrates compliance with the containment requirements in the regulations by means of a performance assessment (PA) that estimates releases from the repository for the regulatory period of 10,000 years. WIPP PA calculations were included in DOE's 2004 WIPP Compliance Recertification Application (CRA-2004) (U.S. DOE 2004), which is currently being reviewed by the EPA. To address questions raised by the EPA's review of the CRA-2004 (Cotsworth 2005), a second WIPP PA calculation, referred to as the 2004 Compliance Recertification Application Baseline Performance Assessment, or CRA-2004 PABC, is in progress (Kanney and Leigh 2005). In both PAs, the verification of correct data flow between process model codes and codes which calculate releases has been an issue of interest.

1.1 BACKGROUND

In WIPP PA, the SUMMARIZE code functions as an interface between certain process model codes (BRAGFLO_DBR, PANEL, NUTS, SECOTP2D) and the PRECCDFGF/CCDFGF code suite. SUMMARIZE is used to extract data from process model output files written in the binary CAMDAT database (CDB) format and write the data to text tables that are read by PRECCDFGF. PRECCDFGF reads in a large number of such tables produced by SUMMARIZE, along with information from codes that do not produce CDB files, to produce an input file for CCDFGF (see Figure 1). This input file is referred to as the release table (RELTAB) file. The CCDFGF code constructs the complimentary cumulative distribution function (CCDF) for radionuclide releases to the accessible environment.

During an August 9 2004 technical exchange meeting, EPA requested additional information regarding how data transfer between process model codes and CCDFGF was verified (CRA Response Activity Tracking Number 08/09/04A was assigned to this verbal request). DOE responded to this request by describing a procedure for inspecting WIPP PA Run Control log files to verify that the correct files were fetched from CMS libraries and used by SUMMARIZE (Dunagan 2004, Piper 2004). Subsequent to the response, several problems with data transfer across the SUMMARIZE interface were discovered, corrected and reported. The problems encountered fell into two categories:

- Certain releases were miscalculated because information in certain SUMMARIZE output tables was not in the column that PRECCDFGF expected (Kirchner and Vugrin 2004).
- Releases to the Culebra were under-estimated in several cases because SUMMARIZE wrote zeros to the output table when it could not locate an input CAMDAT database file.

The impact of the first class of errors on the results and conclusions of the CRA-2004 were determined to be minimal (Vugrin 2004). An informal review of the second type of error indicates that the impact is minimal. A formal review is in progress. Regardless, it was recognized that the SUMMARIZE interface needed to be improved in future PAs.

Both classes of problems outlined can be traced back to errors in the SUMMARIZE input files, not with the SUMMARIZE code itself. Nearly 600 SUMMARIZE input files were used in the CRA-2004 PA. Since each one was prepared manually, the probability of errors is considerable. However, it was determined that improving the SUMMARIZE code was an important part of improving the interface between process model codes and CCDF construction codes. The following steps were followed to improve the SUMMARIZE interface:

- The SUMMARIZE code was updated to Version 3.00 (Gilkey 2005a, WIPP PA 2005b, d, f).
    - A PRECCDFGF driver was added to the set of output drivers in SUMMARIZE. This driver writes header information at the top of the output table. The header information identifies the data that appears in the rest of the table.
    - SUMMARIZE was modified to incorporate strict checking of the vector list, removing the possibility that vectors are inadvertently skipped due to errors in CDB file specification.
- The PRECCDFGF code was updated to Version 1.01 (WIPP PA 2005e, a, c). PRECCDFGF was modified to read the header information produced by SUMMARIZE 3.00 to verify that the data in the input file is in the order required by PRECCDFGF.
- The WIPP PA Run Control scripts were re-written to automate the generation of the SUMMARIZE input files and eliminate errors introduced during manual editing of the files.

The steps outlined above were implemented and used in the CRA-2004 PABC. In order to demonstrate that the improvements made to the SUMMARIZE interface eliminated the errors discussed above, a verification study has been performed. The procedures and results of this verification of the SUMMARIZE interface in the CRA-

2004 PABC are documented in this report. The work described in this report was performed as a deviation from Analysis Plan AP-122.

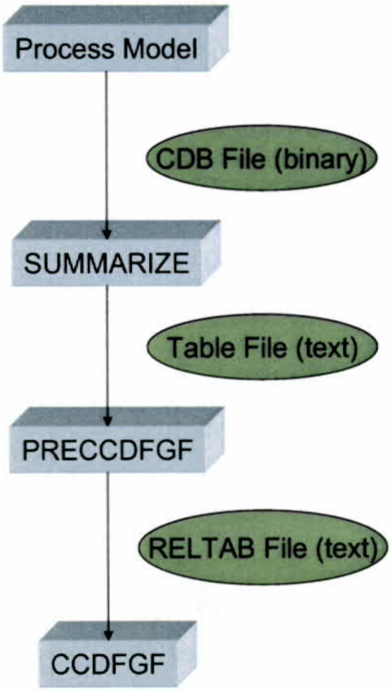


Figure 1. Data flow between process models and CCDFGF

1.2 OBJECTIVE

The objective of the work reported here was to verify that data used to compute estimated releases in the CRA-2004 PABC was correctly transferred across the SUMMARIZE interface between the process model CDB files and the CCDFGF input RELTAB file. Because the objective of this work was to verify data transfer across the SUMMARIZE interface, we focused our attention on data from process models which produce binary output files in the CAMDAT database (CDB) format that are read using SUMMARIZE (i.e., PANEL, NUTS, BRAGFLO, and SECOTP2D). The transfer of information from process models that interface directly with PRECCDFGF (i.e., EPAUNI, and CUTTINGS_S) was not investigated. The correct transfer of this information has been addressed in the validation and verification testing for PRECCDFGF (WIPP PA 2005e).

1.3 ORGANIZATION

The types of information passed from various process models to PRECCDFGF via SUMMARIZE is discussed in Section 2. The methodology used to verify the data transfer is described in Section 3. The results obtained by implementing the methodology are presented in Section 4. Conclusions regarding the adequacy of data transfer across the SUMMARIZE interface are presented in Section 5. APPENDIX A provides the necessary details for reproducing the data extraction and comparison exercise. APPENDIX B displays the source code for the Fortran utility ReadReltab. APPENDIX C presents the validation for the ReadReltab utility. APPENDIX D displays the CheckReltab database and SQL queries used in this work. APPENDIX E discusses the validation of the CheckReltab database.

2 CDB DATA TRANSFERRED ACROSS SUMMARIZE INTERFACE

Process model output CDB files typically contain several types of data: initialization information, quality assurance information, model description data, and analysis results data. A detailed description of the CAMDAT file format is given in the WIPP PA User's Manual for CAMDAT_LIB, Version 1.22 (WIPP PA 1995) and in the Programmer's Manual for CAMCON: Compliance Assessment Methodology Controller (Rechard et al. 1993).

Initialization information and quality assurance information are not transferred across the SUMMARIZE interface, but certain model description data and analysis results data are transferred. Both model description data and analysis results data are accessed via names, which are referred to here as CDB variables. Each variable has an (up to) eight-character identifying name. In some instances model description data in the form of element block properties is extracted by SUMMARIZE. In other instances, analysis results data in the form of global or history variables is extracted.

Element block properties are material parameters that vary only with each material (e.g., rock density, radionuclide partition coefficients) or do not vary at all. Each property has a name for identification. A history variable has a single value for each time step. These values often apply to large regions or the entire analysis, such as total energy or mass of the system. A history variable may also represent a value for a specific node or element. A global variable has a single value for each time step that includes the values of nodal and element variables. The main difference between history variables and global variables is how often the variables are output on the CDB file.

The following sections describe the CDB variables whose values are transferred across the SUMMARIZE interface. The discussion is organized according to the type of information transferred and process model code that produced the CDB file.

2.1 RADIONUCLIDE MOBILIZATION

Radionuclides are mobilized in WIPP brines as dissolved species and as species sorbed to colloids. These mobilization processes are modeled using the PANEL code. WIPP PA treats four colloid types: humic, actinide intrinsic, microbial, and mineral fragment. WIPP PA further assumes that humic colloids transport as do dissolved radionuclides, and that other colloid types do not transport through the Culebra. Thus, the fraction of mobilized radionuclide present in each colloid type is needed to calculate transport through the Culebra.

2.1.1 PANEL_CON

In WIPP PA, mobilized radionuclide concentrations (dissolved and sorbed to colloids, in Curies/m^3) for ^{241}Am, ^{239}Pu, ^{234}U, and ^{230}TH are used to calculate releases. Mobilized radionuclide concentrations in WIPP brines are calculated by the code PANEL run in "concentration" mode (referred to here as PANEL_CON) and stored in the output CDB file as history variables (CLEAM241, CLEPU239, CLEU234, CLETH230). In addition, the total radionuclide concentration (history variable CNETOTAL) is also passed across the SUMMARIZE interface.

PANEL_CON results for two scenarios are used: S1 corresponds to mobilization in Salado brine while scenario S2 corresponds to Castile brine. Because these two scenarios cover the range of possible brines, mobilized radionuclide concentration are not calculated for other scenarios. Instead, for scenarios S4 and S5 (E2 intrusions) mobilized radionuclide concentrations are set equal to those for scenario S1 (E0 intrusion). For scenarios S3 and S6 (E1 and E1E2 intrusions, respectively), mobilized radionuclide concentrations are set equal to those for scenario S2 (E1 intrusion)

PANEL_CON is run for a full set of 100 vectors for each scenario. Thus, a single replicate will have 200 PANEL_CON CDB files.

2.1.2 PANEL_ST

Colloidal mobilization fractions are calculated by the PANEL code and are stored in the PANEL_CON CDB files. However, this information is extracted from the CDB files by SUMMARIZE in a procedure separate from the extracting the mobilized concentrations described above. This colloidal source term extraction process is referred to here as PANEL_ST. The colloidal mobilization fractions do not vary in time, and so are stored in the PANEL_CON CDB files as element block properties. Element blocks 32, 40, 44, and 43 pertain to radionuclides ^{241}Am, ^{239}Pu, ^{234}U, and ^{230}TH, respectively. For each of these element blocks, the microbial, intrinsic, and mineral colloid fractions are stored as element block properties FRCMIC, FRCINT, and FRCMIN, respectively.

As with PANEL_CON, 100 vectors with two scenarios (S1 and S2) for a total of 200 CDB files are processed by SUMMARIZE. Colloidal mobilization fractions for scenarios S4-S6 are assigned in the same manner describe in Section 2.1.1.

2.2 *TRANSPORT TO CULEBRA*

Radionuclide transport to the Culebra for undisturbed conditions, single E1 intrusions, and single E2 intrusions (BRAGFLO scenarios S1-S5) is calculated using the NUTS code. The combined E1E2 intrusion case (BRAGFLO scenario S6) is calculated using the code PANEL run in "intrusion mode" (PANEL_INT).

2.2.1 NUTS

The NUTS simulation calculates radionuclide transport from the time of intrusion to the end of the 10,000-year regulatory period for the four radionuclides ^{241}Am, ^{239}Pu, ^{234}U, and ^{230}TH. Subsequent post-processing of the NUTS output CDB file with the ALGEBRACDB code produces a CDB file containing the radioactivity (in Curies, dissolved or sorbed to colloids) released to the Culebra during each time interval stored in the global variables A00AM241, A00PU239, A00U234, and A00TH230. In addition, the total activity in all marker beds at the land withdrawal boundary is stored in the global variable EPALWMBT. For each replicate, there are a total of 15 files per vector as shown in Table 1. Since there are 100 vectors per replicate, this amounts to a grand total of 1500 NUTS CDB files per replicate used in the release calculations.

Table 1. NUTS simulations (per vector)

Scenario	Intrusion Times (years)	Subtotal
S1	None	1
S2	100, 350	2
S3	1000, 3000, 5000, 7000, 9000	5
S4	100, 350	2
S5	1000, 3000, 5000, 7000, 9000	5
	Total	15

2.2.2 PANEL_INT

PANEL_INT simulations calculate radionuclide transport from the time of intrusion to the end of the 10,000-year regulatory period for the E1E2 intrusion (BRAGFLO scenario S6). PANEL_INT writes the radioactivity (in Curies, dissolved or sorbed to colloids) released to the Culebra during each time interval for the four radionuclides ^{241}Am, ^{239}Pu, ^{234}U, and ^{230}TH to the history variables LDCAM241, LDCPU239, LDCU234, and LDCTH230, respectively. PANEL_INT models intrusions at 100, 350, 1000, 2000, 4000, 6000, and 9000 years. For each intrusion, 100 vectors are calculated for a total of 700 output CDB files per replicate.

2.3 TRANSPORT THROUGH CULEBRA (SECOTP2D)

Radionuclide transport through the Culebra is calculated by the SECOTP2D code suite (PRESCOTP2D, SECOTP2D, and POSTSECOTP2D). A hypothetical 1kg source distributed over 0 to 50 years is placed at the center of the waste panel for several radionuclides and the fraction of the original source transported to the accessible environment over the 10,000-year regulatory period is calculated. POSTSECOTP2D stores the time history for cumulative mass discharge at the land withdrawal boundary for ^{241}Am, ^{239}Pu, ^{234}U, and ^{230}TH (both injected and as a daughter product of ^{234}U) in the history variables MT2AM241, MT2PU239, MT2U234, MT2TH230, MT2TH23A, respectively. SECOTP2D is run for two mining scenarios: full mining and partial mining. A set of 100 vectors is used for each mining scenario. Thus, for each replicate there are 200 output CDB files.

2.4 DIRECT BRINE RELEASE (BRAGFLO_DBR)

Single intrusion direct brine release volumes are calculated by the code BRAGFLO in DBR mode (referred to here as BRAGFLO_DBR). BRAGFLO_DBR output CDB files store the brine volume in the history variable BRIN_REL. Although it is stored as history variable, there is only a single time for BRIN_REL (the elapsed time for a DBR event). This time is not used in release calculations. Instead, the time of the intrusion that lead to the DBR is of interest. For each replicate, there are a total of 78 direct brine release files per vector as shown in Table 2. Since there are 100 vectors per replicate, this amounts to a total of 7800 CDB files.

Table 2. Direct brine release simulations (per vector)

Scenario	Cavities	Intrusion Times (years)	Subtotal
S1	U,L,M	100, 350, 1000, 3000, 5000, 10000	18
S2	U,L,M	550, 750, 2000, 4000, 10000	15
S3	U,L,M	1200, 1400, 3000, 5000, 10000	15
S4	U,L,M	550, 750, 2000, 4000, 10000	15
S5	U,L,M	1200, 1400, 3000, 5000, 10000	15
		Total	78

3 VERIFICATION APPROACH

The basic approach used to verify data transfer across the SUMMARIZE interface was to extract data from the CDB files using a code other than SUMMARIZE, and compare that data with corresponding entries in the CCDFGF RELTAB file. The data was extracted from the CDB files using the code GROPECDB, Version 2.12 (WIPP PA 1996). To make the extraction process tractable, traceable and reproducible, Digital Command Language (DCL) scripts were written to automate the process. For similar reasons, a Fortran utility was written to automate the process of extracting data from the RELTAB file. The two independent data sets were then loaded into a Microsoft® Access database (CheckReltab) to facilitate comparing them. The data sets could be efficiently compared by constructing database queries. Figure 2 illustrates the basic procedure followed.

A slight complication in the data comparison process was introduced by the fact that GROPECDB does not perform interpolation in time while SUMMARIZE does. If GROPECDB receives a request for the value of a variable at a time that does not exist in the CDB file, it returns the value at the nearest time. When this situation occurred, the SUMMARIZE behavior was emulated by interpolating the data within the CheckReltab database before comparison to the RELTAB data value.

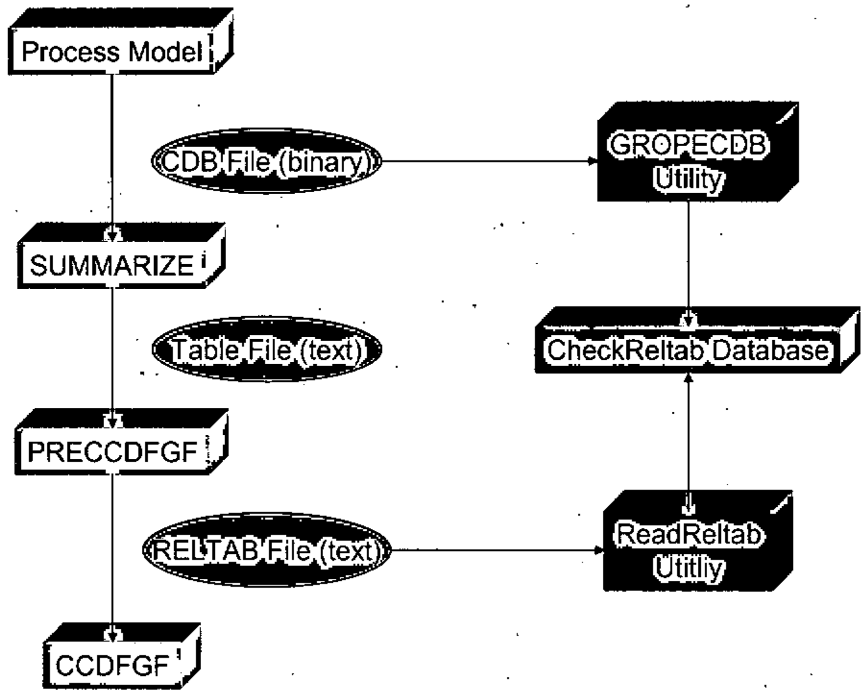


Figure 2. SUMMARIZE verification flow chart

3.1 DATA EXTRACTION FROM CDB FILES

As explained above, the code GROPECDB was used to extract relevant data values from the process model CDB files. Because many CDB variables have values for each time step, and there are usually separate CDB files for each scenario, vector, intrusion time, cavity combination, the use of GROPECDB was automated with DCL scripts. The DCL scripts loop over scenarios, vectors, intrusion times, cavities, as appropriate. For each appropriate combination, the DCL scripts perform the following operations:

- Fetch the required CDB file
- Write an input control file for GROPECDB,
- Execute GROPECDB on the CDB file,
- Extract the desired data from the GROPECDB output file
- Write extracted data to the script output file.

In general, a script output file is produced for each replicate/scenario/intrusion time/cavity/vector/CDB variable combination. The script output file contains a single value or a time series as appropriate. Table 3 shows the scripts used and the information extracted by each script. The scripts have been archived in CMS library LIBCRA1BC_SUMVERIFY. Appendix A provides a detailed description of how to execute each script to produce the full set of output files used in this work.

Table 3. DCL scripts used to extract data from CDB files

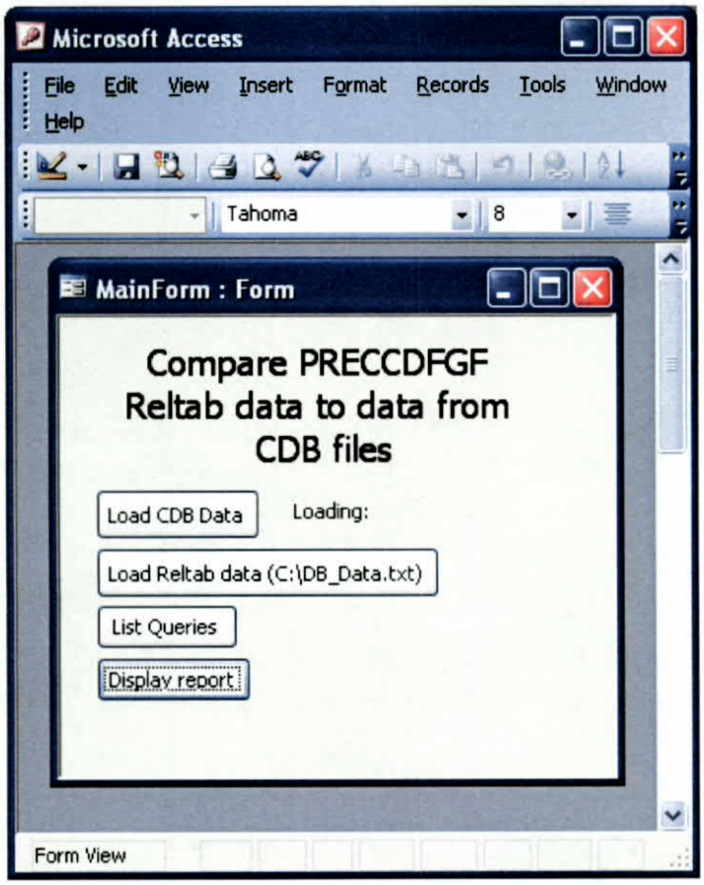
DCL Script	WIPP PA Code	Information Extracted
DCOMP_PANEL_CON.COM	PANEL_CON	Mobilized radionuclide concentrations in WIPP brines
DCOMP_PANEL_ST.COM	PANEL_ST	Colloidal mobilization fractions
DCOMP_NUT.COM	NUTS	Radionuclide released to Culebra
DCOMP_PANEL_INT.COM	PANEL_INT	Radionuclide released to Culebra
DCOMP_ST2D.COM	SECOTP2D	Fractional amount of radionuclide transported to LWB from 1kg source
DCOMP_DBR.COM	BRAGFLO_DBR	Brine volume

3.2 DATA EXTRACTION FROM RELTAB FILES

Data were extracted from the CRA-2004 PABC PRECCDFGF release table file, CCGF_CRA1BC_RELTAB_R1.DAT, using the Fortran program ReadReltab (Appendix 2). The release table file was generated by PRECCDFGF Version 1.01 and stored in the LIBCRA1BC_CCGF library on VMS. The file was extracted from the library and transferred to a PC running Windows XP using FTP. ReadRealtab reads the file and creates the file C:\DB_Data.txt. The DB_Data.txt file contains the various release values in a comma-delimited format with one header record. The header record lists the field names. The fields are the replicate, scenario, cavity, vector, intrusion time index, intrusion time, series time index, series time, release name and release value. The "scenario" field is used to identify the repository state (E0, E1, E2, E1E2) or mining scenario (full or partial mining). The "cavity" field is used to identify the location of the initial intrusion (lower, middle or upper group of panels) or subsequent intrusions (same, adjacent or non-adjacent). The intrusion time represents the initial intrusion and the series time represents the subsequent intrusion. Not all releases are associated with initial or subsequent release times and in those cases the time and associated index files are assigned values of zero. The release name is that used in the original CDB file.

3.3 DATA COMPARISON USING MICROSOFT® ACCESS

Data extracted from the CDB files and the PRECCDFGF release table were subsequently loaded into the Microsoft® Access database CheckReltab using code placed within a module of the database (see APPENDIX D). The uploads are initiated by selecting the "Load CDB Data" and "Load Reltab Data" buttons (see Figure 3). The data from the release table were placed in the table RELTAB_Data. The data from the CDB files were read and placed either in the table CDB_RawData_Nuts or in the table CDB_Data, depending on whether the data were produced by the NUTS code or other codes, respectively. The NUTS data were separated from the other data because many of the values in the PRECCDFGF release tables were derived from linear interpolation of the data contained in the CDB files. Linear interpolation was used in SUMMARIZE to produce release estimates on 50 year increments from the time of the initial release until 10,000 years. After loading the CDB data, code in CheckReltab was used to compute interpolations on the data contained within the CDB_RawData_Nuts table equivalent to those done in SUMMARIZE. The interpolated data were then appended to the CDB_Data table. In addition, PANEL_CON radionuclide concentration values for 125 and 175 years were created by interpolation of the CDB data and appended to the CDB_Data table (see Section 4.1.1).



Compare PRECCDFGF Reltab data to data from CDB files

Figure 3. The main form of the Checkreltab database

The CDB and release table data were compared using a series of queries. A pair of queries for each data source (code), e.g. CDB_ColloidData and Reltab_ColloidData, were used to select data from the RELTAB_Data and CDB_Data tables. The select query for the CDB data selected data only for the associated code and specified vectors, assigned repository state codes based on the scenario code of the data, and converted the times from seconds to years, if required. The select query for the release table data extracted data for the specified vectors of the associated code. The data in each pair of queries were compared in another query, e.g. RelPercentDiff_Colloids, by joining the data through the release name, vector, intrusion time and series times (or their respective indices). A relative percent difference (RPD) was computed for the paired values using the equation

$$RPD = \frac{|X_1 - X_2|}{(X_1 + X_2)/2} \times 100\,;\ (X_1 + X_2) > 0$$

In the case where X_1 and X_2 were zero the RPD was assigned a value of 0.

4 RESULTS

In order to verify the correct data transfer across the SUMMARIZE interface, the steps described in Section 3 were performed for replicate R1 of the CRA-2004 PABC. The following sections summarize the results of the verification exercise.

4.1 DATA EXTRACTION FROM CDB FILES

4.1.1 PANEL_CON

Table 4 lists the files and libraries used for the PANEL_CON CDB data extraction. The DCL script DCOMP_PANEL_CON.COM was used to extract time series for the CDB variables CLEAM241, CLEPU239, CLEU234, CLETH230, and CNETOTAL. The requested times in the series were: 100, 125, 175, 350, 1000, 3000, 5000, 7500, and 10000 years.

The script produced a separate output file for each variable (containing the time series for that variable). The total number of output files for a single replicate is 1000 (2 scenarios x 100 vectors x 5 variables).

Table 4. Files and libraries for PANEL_CON CDB data extraction

DCL Script	DCOMP_PANEL_CON.COM
Script Input File	DCOMP_PANEL_CON_CRA1BC_R1:INP
Script Log File	DCOMP_PANEL_CON_CRA1BC_R1.LOG
Script Output files	DC_PCON_CRA1BC_R1_S*s*_V*vvv*_*variable*.out
Script Archive	LIBCRA1BC_SUMVERIFY, class CRA1BC-0
CDB files	PANEL_CON_CRA1BC_R1_S*s*_V*vvv*.CDB
CDB Archive	LIBCRA1BC_PANEL, class CRA1BC-0

1. $s \in \{1, 2\}$

2. $vvv \in \{001, 002, ..., 100\}$

3. *variable*∈{CLEAM241, CLEPU239, CLEU234, CLETH230, CNEATOTAL}

4.1.2 PANEL_ST

Table 5 lists the files and libraries used for the PANEL_ST CDB data extraction. The DCL script DCOMP_PANEL_ST.COM was used to extract values for the CDB variables FRCINT, FRCMIC, and FRCMIN for isotopes ^{241}Am, ^{239}Pu, ^{234}U, and ^{230}Th. In this case, there is no time associated with the data values. A total of 12 data values were extracted from each CDB file.

The script produced a separate output file for each variable. The total number of output files for a single replicate is 2400 (2 scenarios x 100 vectors x 12 variables).

Table 5. Files and libraries for PANEL_ST CDB data extraction

DCL Script	DCOMP_PANEL_ST.COM
Script Input File	DCOMP_PANEL_ST_CRA1BC_R1.INP
Script Log file	DCOMP_PANEL_ST_CRA1BC_R1.LOG
Script Output files	DC_PST_CRA1BC_R1_S*s*_V*vvv*_*variable*.out
Script Archive	LIBCRA1BC_SUMVERIFY, class CRA1BC-0
CDB files	PANEL_CON_CRA1BC_R1_S*s*_V*vvv*.CDB
CDB Archive	LIBCRA1BC_PANEL, class CRA1BC-0

1. $s \in \{1, 2\}$

2. $vvv \in \{001, 002, ..., 100\}$

3. $$variable \in \left\{ \begin{array}{l} \text{AM241\_FRCINT, AM241\_FRCMIC, AM241\_FRCMIN,} \\ \text{PU239\_FRCINT, PU239\_FRCMIC, PU239\_FRCMIN,} \\ \text{U234\_FRCINT, U234\_FRCMIC, U234\_FRCMIN,} \\ \text{TH230\_FRCINT, TH230\_FRCMIC, TH230\_FRCMIN} \end{array} \right\}$$

4.1.3 NUTS

Table 6 lists the files and libraries used for the NUTS CDB data extraction. The DCL script DCOMP_NUT.COM was used to extract time series for the CDB variables A00AM241, A00PU239, A00U234, A00TH230, and EPALWMBT. The script produced a separate output file containing a times series for each variable at each scenario/intrusion time combination. The total number of output files for a single replicate is 7500 (15 CDB files x 100 vectors x 5 variables).

Table 6. Files and libraries for NUTS CDB data extraction

DCL Script	DCOMP_NUT.COM
Script Input File	DCOMP_PANEL_ST_CRA1BC_R1_Ss_1.INP
Script Log File	DCOMP_PANEL_ST_CRA1BC_R1_Ss_1.LOG
Script Output files	DC_NUT_CRA1BC_R1_Ss_2_T*ttttt*_V*vvv*_*variable*.out
Script Archive	LIBCRA1BC_SUMVERIFY, class CRA1BC-0
CDB files	ALG_NUT_ISO_CRA1BC_R1_Ss_2_V*vvv*.CDB ALG_NUT_INT_CRA1BC_R1_Ss_3_T*ttttt*_V*vvv*.CDB
Input screening files	SCREEN_NUT_SCN_CRA1BC_R1_Ss_2.OUT
CDB and screening file archive	LIBCRA1BC_NUTR1Ss_2, class CRA1BC-0

1. $s_1 \in$ {1, 2a ,2b, 3a, 3b, 4a, 4b, 5a, 5b}
2. $s_2 \in$ {1, 2, 3, 4, 5}
3. $s_3 \in$ {2, 3, 4, 5}
4. *ttttt*∈{00100, 00350} for scenarios S2 and S4
5. *ttttt*∈{01000, 03000, 05000, 07000, 09000} for scenarios S3 and S5
6. *vvv*∈{001, 002, ..., 100}
7. *variable*∈{A00AM241, A00PU239, A00U234, A00TH230, EPALWMBT}

4.1.4 PANEL_INT

Table 7 lists the files and libraries used for the PANEL_INT CDB data extraction. The DCL script DCOMP_PANEL_INT.COM was used to extract time series for the CDB variables LDCAM241, LDCPU239, LDCU234, and LDCTH230, and SDCPB210 (the SDCPB210 variable is not used in release calculations; it is a placeholder used to give NUTS and PANEL_INT SUMMARIZE tables the same format). The script produced a separate output file for each variable time series at each intrusion time. The total number of output files for a single replicate is 3500 (100 vectors x 7 intrusions x 5 variables).

Table 7. Files and libraries for PANEL_INT CDB data extraction

DCL Script	DCOMP_PANEL_INT.COM
Script Input File	DCOMP_PANEL_INT_CRA1BC_R1.INP
Script Log File	DCOMP_PANEL_INT_CRA1BC_R1.LOG
Script Output files	DC_PST_CRA1BC_R1_S6_TI*tttt*_V*vvv*_*variable*.out
Script Archive	LIBCRA1BC_SUMVERIFY, class CRA1BC-0
CDB files	PANEL_INT_CRA1BC_R1_S6_TI*tttt*_V*vvv*.CDB
CDB Archive	LIBCRA1BC_PANEL, class CRA1BC-0

1. *tttt*∈{0100, 0350, 1000, 2000, 4000, 6000, 9000}
2. *vvv*∈{001,002,...,100}
3. *variable*∈{LDCAM241, LDCPU239, LDCU234, LDCTH230, SDCPB210}

4.1.5 SECOTP2D

Table 8 lists the files and libraries used for the SECOTP2D CDB data extraction. The DCL script DCOMP_ST2D.COM was used to extract time series for the CDB variables MT2AM241, MT2PU239, MT2U234, MT2TH230, and MT2TH23A for each scenario/vector combination. The script produced a separate output file for each time series. The total number of output files for a single replicate is 1000 (2 scenarios x 100 vectors x 5 variables).

Table 8. Files and libraries for SECOTP2D CDB data extraction

DCL Script	DCOMP_ST2D.COM
Script Input File	DCOMP_ST2D_CRA1BC_R1.INP
Script Log File	DCOMP_ST2D_CRA1BC_R1.LOG
Script Output files	DC_ST2D_CRA1BC_R1_M*m*_V*vvv*_*variable*.out
Script Archive	LIBCRA1BC_SUMVERIFY, class CRA1BC-0
CDB files	ST2D_CRA1BC_R1_M*m*_V*vvv*.CDB
CDB Archive	LIBCRA1BC_ST2D, class CRA1BC-0

1. $m \in \{F, P\}$
2. $vvv \in \{001, 002, ..., 100\}$
3. $variable \in \{$MT2AM241, MT2PU239, MT2U234, MT2TH230, MT2TH23A$\}$

4.1.6 BRAGFLO_DBR

Table 9 lists the files and libraries used for the BRAGFLO_DBR CDB data extraction. The DCL script DCOMP_DBR.COM was used to extract values for the CDB variable BRIN_REL for each scenario/intrusion/cavity/vector combination. The script produced a separate output file for each value. The total number of output files for a single replicate is 7800 (7800 CDB files x 1 variable).

Table 9. Files and libraries for BRAGFLO_DBR CDB data extraction

DCL Script	DCOMP_DBR.COM
Script Input File	DCOMP_PANEL_ST_CRA1BC_R1_Ss_1.INP
Script Log File	DCOMP_PANEL_ST_CRA1BC_R1_Ss_1.LOG
Script Output files	DC_NUT_CRA1BC_R1_Ss_2_T*ttttt*_Vvvv_BRIN_REL.out
Script Archive	LIBCRA1BC_SUMVERIFY, class CRA1BC-0
CDB files	ALG3_DBR_CRA1BC_R1_Ss_2_Tttttt_c_Vvvv.CDB
CDB Archive	LIBCRA1BC_DBRR1S s_2,class CRA1BC-0

1. $s_1 \in \{1, 24 ,35\}$
2. $s_2 \in \{1, 2, 3, 4, 5\}$
3. $ttttt \in \{00550, 00750, 02000, 04000, 10000\}$ for scenarios S2 and S4
4. $ttttt \in \{01200, 01400, 03000, 05000, 10000\}$ for scenarios S3 and S5
5. $vvv \in \{001, 002, ..., 100\}$

4.2 DATA EXTRACTION FROM RELTAB FILES

PRECCDFGF release table file CCGF_CRA1BC_RELTAB_R1.DAT was fetched from CMS library LIBCRA1BC_CGF, class CRA1BC-0 on the WIPP PA Alpha cluster and transferred to the PC via FTP. The relevant data was extracted from the release table file to the text output file DB_Data.txt, using the Fortran program ReadReltab (APPENDIX B). The ReadReltab source code and binary executable, and the output file have been archived on the WIPP PA Alpha cluster in CMS library LIBCRA1BC_SUMVERIFY, class CRA1BC-0 (see APPENDIX A).

4.3 DATA COMPARISON USING MICROSOFT® ACCESS

The CheckReltab database includes a report for each code that documents all of the differences between the values extracted from the CDB files by the DCL scripts and those extracted from the RELTAB file using the ReadReltab utility. In addition, the summary report SummaryOfDifferences, lists the maximum and average difference between the data by code. The data reported herein are for a set of five vectors (5, 30, 57, 62 and 96). Vector 57 was selected to ensure that at least one vector with nonzero Culebra releases was included in the comparisons. The remaining vectors were randomly selected. The number and selection of vectors to be analyzed can be modified by changing the entries in the SelectedVectors table.

Complete listings of the RPDs for the data can be generated by selecting the "Display Report" button on the main menu of CheckReltab (see Figure 3). A report for a specific code can then be created by selecting the appropriate button in the "Display Reports" form (see Figure 4). Copies of the reports for the cases described herein were exported to Microsoft$^{\circledR}$ Word. These reports are named:

- SummaryOfDifferences.rft
- PercentRelDiff_*code*.rft

where $code \in \{\text{Colloids, Culebra, DBR, Intrusion, Nuts, Panel}\}$

These files were inserted into the RPD_Reports.zip file and archived in the CMS library LIBCRA1BC_SUMVERIFY on the WIPP PA Alpha Cluster.

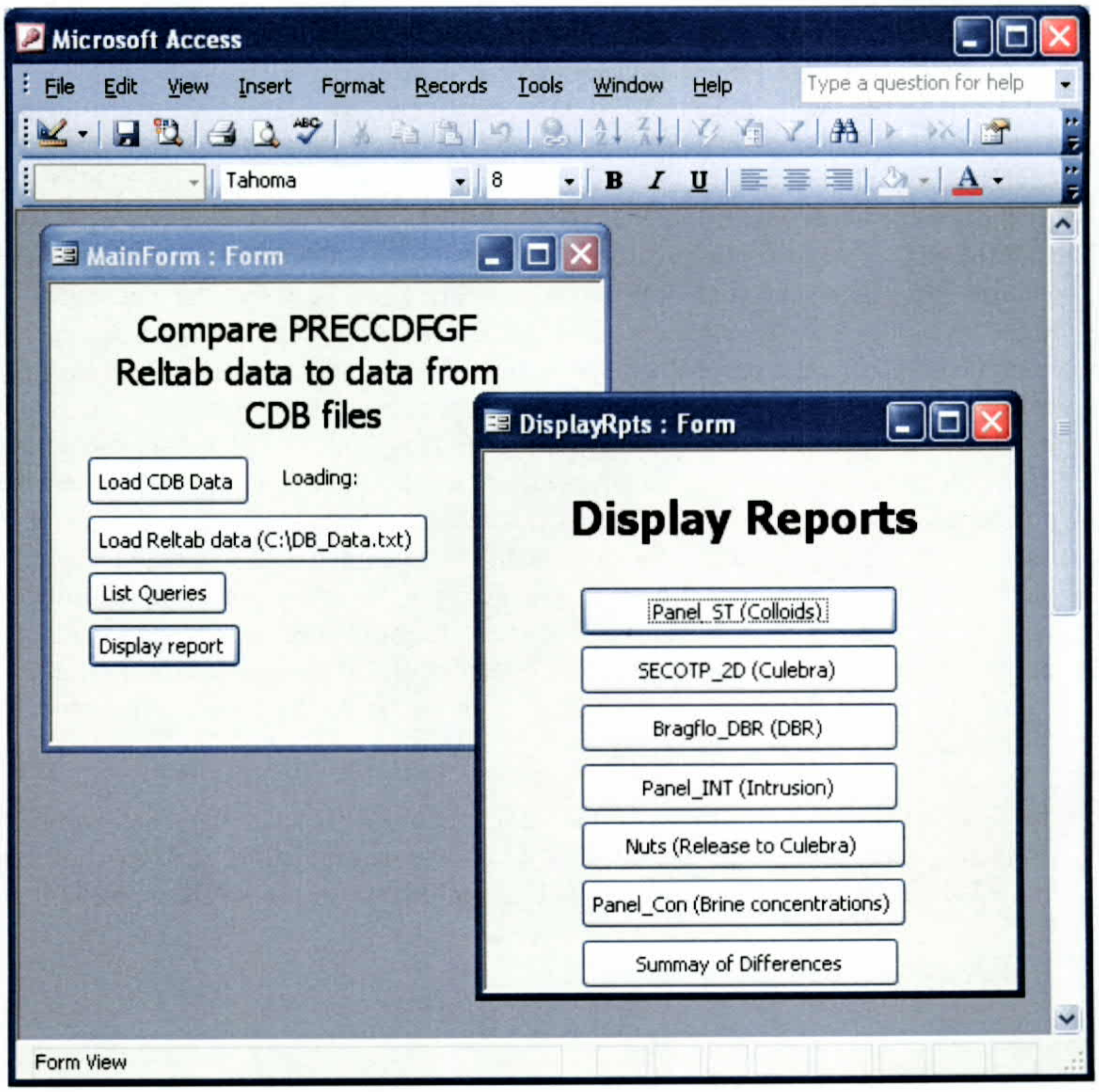


Figure 4. Display Reports form of the CheckReltab database

Table 10 summarizes the results of the data comparison for each code, displaying the number of values compared, the mean RPDs and the maximum RPDs. The mean RPDs are very small (typically on the order of 0.02% or less). The maximum RPDs are also small (typically <0.1%), except in a small number of cases for NUTS, PANEL_CON, and SECOTP2D results.

Small differences are to be expected because 1) the data are passed between SUMMARIZE and PRECCDFGF as formatted values and 2) they may have been subject to the introduction of rounding error in SUMMARIZE's linear interpolation routine. However, some of the difference observed appear to be larger than expected. For 19 NUTS values (less than 0.1% of the comparisons), 19 PANEL_CON values (~4% of the comparisons), and 37 SECOTP2D values (less than 1% of the comparisons), the maximum RPDs were orders of magnitude lager than the mean. The maximum RPDs for these cases were approximately 59%, 12%, and 35% for NUTS, PANEL_CON, and SECOTP2D, respectively.

Investigation of these results showed that these discrepancies were due to a problem in the interpolation routine of the SUMMARIZE code. This problem was due to SUMMARIZE using single precision variables in the calculation of the interpolated values and was manifested only when the difference between the dependent (Y) values was less than the smallest value that could be computed in a single precision computation. In such cases the slope is returned as zero and the upper limit of the selected interval was returned. An example calculation that illustrates this issue is included Software Problem Report 05-002 (Gilkey 2005b). Had the interpolation been computed without numerical truncation then the value returned would have been correct and the relative percent difference would have been small. Since this problem occurs only when the width of the dependent variable interval is less than the smallest value that can be represented with single precision, its impact on computed releases will be insignificant. Regardless of the insignificance of this problem, the interpolation routine in SUMMARIZE is currently being revised to perform the calculation using double precision variables to eliminate the problem in future PAs (Gilkey 2005b).

Table 10 also shows results for mean and maximum RPD when the small number of problem cases have been excluded. When the interpolation problem has been excluded, the mean RPDs are 0.01% or less and the maximum RPDs are all less than 0.1%.

Table 10. Mean and maximum percent differences by code

Source	Release	Maximum percent difference	Mean percent difference	Number
BRAGFLO_DBR	Direct release brine volume	0.0354	0.0024	390
NUTS	Cumulative discharge to Culebra	59.1345	0.0070	28370
NUTS[1]	Cumulative discharge to Culebra	0.7731	0.0024	28351
PANEL_CON	Radionuclide concentrations in brine	12.3004	0.0825	450
PANEL_CON[1]	Radionuclide concentrations in brine	0.0944	0.0107	431
PANEL_INT	Cumulative discharge to Culebra	0.0540	0.0081	19160
PANEL_ST	Mobilized colloid fraction	0.0266	0.0063	120
SECOTP_2D	Cumulative discharge from Culebra	35.3755	0.0171	9950
SECOTP_2D[1]	Cumulative discharge from Culebra	0.0496	0.0006	9913

1. Excludes RPDs > 0.1%. This excludes comparisons obviously influenced by the SUMMARIZE interpolation problem.

5 SUMMARY AND CONCLUSIONS

Data transfer across the SUMMARIZE interface in the CRA-2004 PABC has been verified by extracting the data from the process model CDB files using a code other than SUMMARIZE and comparing that data with corresponding entries in the CCDFGF RELTAB file. To make the extraction process tractable, traceable and reproducible, Digital Command Language (DCL) scripts were written to automate the CDB data extraction process. For similar reasons, a Fortran utility was written to automate the process of extracting data from the RELTAB file. The two independent data sets were then loaded into a Microsoft® Access database to facilitate comparing them. The data sets could be efficiently compared by constructing database queries. The data extraction and comparison exercise was performed on replicate R1 of the CRA-2004 PABC.

Performing the data extraction and comparison exercise revealed that, except for a minor flaw in the interpolation algorithm used in SUMMARIZE, the data values in process model CDB files matched that found in the RELTAB file very well. The flaw in the SUMMARIZE interpolation algorithm affected only a very small number of data values, and was only observed when the dependent variable interval was smaller than can be represented in the single precision data format of CDB files. Thus, the flaw should have no impact on computed releases.

Therefore, we conclude that the exercise reported herein has verified that data from process model CDB files is being transferred across the SUMMARIZE interface with sufficient accuracy.

6 REFERENCES

Chavez, M. J. 2004. Nuclear Waste Management Procedure NP 19-1: Software Requirements, Revision 11. Sandia National Laboratories, Carlsbad, NM.

Cotsworth, E. 2005. Sixth Set of EPA Comments (March 4, 2005 Letter to Ines Triay). U.S. Environmental Protection Agency.

Dunagan, S. 2004. Explanation of how SUMMARIZE and CCDFGF are checked/verified/tested for capturing the correct CDB data streams in the WIPP CRA-2004 Performance Assessment. Technical Memorandum ERMS 536767, Sandia National Laboratories, Carlsbad, NM.

Gilkey, A. P. 2005a. Change Control Form for SUMMARIZE Version 2.20. ERMS 540117, Sandia National Laboratories, Carlsbad, NM.

Gilkey, A. P. 2005b. Software Problem Report 05-002 (SUMMARIZE 3.00). ERMS 540976, Sandia National Laboratories, Carlsbad, NM.

Kanney, J. F., and C. Leigh. 2005. Analysis Plan for Post CRA PA Baseline Calculation (AP-122). ERMS 539624, Sandia National Laboratories, Carlasbad, NM.

Kirchner, T. B., and E. D. Vugrin. 2004. Errors Affecting Spallings Releases, Rev. 0. ERMS 537852, Sandia National Laboratories, Carlsbad, NM.

Piper, L. 2004. 4th Response Submittal to EPA. U.S. Department of Energy, Carlsbad, NM.

Rechard, R., A. P. Gilkey, H. J. Iuzzolino, D. Rudeen, and K. A. Byle. 1993. Programmer's Manual for CAMCON: Compliance Assessment Methodology Controller (SAND 90-1984). ERMS 224600, Sandia National Laboratories, Carlsbad, NM.

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

Vugrin, E. D. 2004. Corrected CRA Figures. Technical Memorandum ERMS 538260, Sandia National Laboratories, Carlsbad, NM.

WIPP PA. 1995. User's Manual for CAMDAT_LIB, Version 1.22. ERMS 227727, Sandia National Laboratories, Carlsbad, NM.

WIPP PA. 1996. User's Manual for GROPECDB Version 2.12. ERMS 237496, Sandia National Laboratories, Carlsbad, NM.

WIPP PA. 2005a. Design Document for PRECCDFGF Version 1.01. ERMS 539295, Sandia National Laboratories, Carlsbad, NM.

WIPP PA. 2005b. Design Document for SUMMARIZE Version 3.00. ERMS 540107, Sandia National Laboratories, Carlsbad, NM.

WIPP PA. 2005c. User's Manual for PRECCDFGF Version 1.01. ERMS 539298, Sandia National Loaboratories, Carlsbad, NM.

WIPP PA. 2005d. User's Manual for SUMMARIZE Version 3.00. ERMS 540110, Sandia National Laboratory, Carlsbad, NM.

```
 Common/ ControlFileName/ ControlFile
   Character*120 ControlFile

 Common /TRU_file/ TRU_in
 Character*30 TRU_in

 Common /CUSP_file/CUSP_S1_L_in(6),CUSP_S1_U_in(6),CUSP_S2_L_in(5),&
&                  CUSP_S2_U_in(5),CUSP_S3_L_in(5),CUSP_S3_U_in(5),&
&                  CUSP_S4_L_in(5),CUSP_S4_U_in(5),CUSP_S5_L_in(5),&
&                  CUSP_S5_U_in(5),CUSP_S1_M_in(6),CUSP_S2_M_in(5),&
&                  CUSP_S3_M_in(5),CUSP_S4_M_in(5),CUSP_S5_M_in(5)
 Character*30 CUSP_S1_L_in, CUSP_S1_U_in, CUSP_S2_L_in,             &
&                  CUSP_S2_U_in, CUSP_S3_L_in, CUSP_S3_U_in,        &
&                  CUSP_S4_L_in, CUSP_S4_U_in, CUSP_S5_L_in,        &
&                  CUSP_S5_U_in, CUSP_S1_M_in, CUSP_S2_M_in,        &
&                  CUSP_S3_M_in, CUSP_S4_M_in, CUSP_S5_M_in

 Common /SPALLING_file/ SPALLING_out
 Character*30 SPALLING_out

 Common/Cuttings/ PCH, PRH, TMCH, TMRH, CCH, CRH,                   &
&                 ACH, ARH,NTMCH, DRILDIAM,                         &
&                 NTMRH, NRH, NCH
 Integer NTMCH, NTMRH, NRH, NCH
 Double Precision, Pointer:: PCH(:), PRH(:), TMCH(:), TMRH(:)
 Double Precision, Pointer:: CCH(:,:), CRH(:,:), ACH(:)
 Double Precision, Pointer:: ARH(:), DRILDIAM(:)

 Common /Cutting_file/Cutting_out
 Character*30 CUTTING_out

 Double Precision TotalInventory, HeightRemoteHandledWaste,         &
&                 HeightContactHandledWaste,                        &
&                 VolumeFractionContactWaste,                       &
&                 VolumeFractionRemoteHandledWaste, EXVOL, ACTI,    &
&                 Avogadro, SecPerYr, BqPerCurie,                   &
&                 ContactHandledWasteArea,                          &
&                 BrinePocketPenetrationProb,                       &
&                 FinalDrillingRate,PluggingPatternProb,            &
&                 RemoteHandledWasteArea,FinalMiningRate,           &
&                 MiningTransitionTime,DrillingTransitionTime,      &
&                 InitialMiningReduction,                           &
&                 InitialDrillingReduction,BermArea,WUF,            &
&                 HalfLife, AMass, RLim
 Common /Database/TotalInventory, HeightRemoteHandledWaste,         &
&                 HeightContactHandledWaste,                        &
&                 VolumeFractionContactWaste,                       &
&                 VolumeFractionRemoteHandledWaste, EXVOL, ACTI,    &
&                 Avogadro, SecPerYr, BqPerCurie,                   &
&                 ContactHandledWasteArea,                          &
&                 BrinePocketPenetrationProb,                       &
&                 FinalDrillingRate,PluggingPatternProb(3),         &
&                 RemoteHandledWasteArea,FinalMiningRate,           &
&                 MiningTransitionTime,DrillingTransitionTime,      &
&                 InitialMiningReduction,                           &
&                 InitialDrillingReduction,BermArea,WUF,            &
&                 HalfLife(4), AMass(4), RLim(4)

 Common /DBR_files/DBR_S1_in(6,3,2), DBR_S2_in(5,3,2,2),            &
&                 DBR_S3_in(5,3,2,2)
 Character*30 DBR_S1_in, DBR_S2_in, DBR_S3_in

 Common /EPAUNI_files/ EPAUNI_in(2)
 Character*30 EPAUNI_in

 Common /CUSP_file/ CUSP_in
 Character*30 CUSP_in

 Common /LHS/ NVAR, XLHS, LHSVBP, LHSBPP, NBP
   Integer NVAR, LHSVBP, LHSBPP, NBP
   Double Precision, Pointer :: XLHS(:)

   Common /LHSdose/ LHSMMP, LHRING, LHRINH, LHTEXP, LHDFM, LHRF
   Integer LHSMMP, LHRING, LHRINH
   Integer, Pointer:: LHTEXP(:,:), LHDFM(:)

 Common/LHS_files/LHS_unit, LHS_in, LHS_VarNames
   Integer LHS_unit              !Unit number for the file
   Character*120 LHS_in   !File name
```

```
      Write(6,*) '>>>>>ERROR(S) ENCOUNTERED, EXECUTION TERMINATED',     &
     &           '101'
      Stop

!.....Premature eof encountered reading spallings release tables
 9200 Continue
      Write(6,*) '>>>>>ERROR(S) ENCOUNTERED, EXECUTION TERMINATED'      &
     &           ,'102'
      Stop

!.....Format statements
 1001 Format(2I10,1X,A)
 1002 Format(2I10,E10.3)
 1003 Format((20E10.3))
 1004 Format(I10,1X, 5(A),I2)
 1005 Format(I10,1X, 2(A),I2)
 2001 Format(A)
 3001 Format(A)
      End

!-------------------------------------------------------------------------------

      Subroutine ReadLHSVariables(IOBS, unit)
!     Author: T. B. Kirchner
!     Purpose: Read current lhs sample data and interpret into input data

      Integer IOBS, unit
      Include 'ccgf1_preccdfgf.fi'

      read(unit,4001) PBRINE
      Return
 4001 FORMAT(E16.6)
          End
```

The include file

```
      PARAMETER (NTMBLE0L=6, NTMBLE0U=6, NTMBLE1S=7,                    &
     &           NTMBLE1D=7, NTMBLE2S=7, NTMBLE2D=7,                    &
     &           NRBL=38)
      Parameter (NIND=5, MAXPTH=4)
      Parameter (NDCHAIN=4, NCOLSP=3, NTM=7)
      Integer nPanel, nSame, nEvent
      Parameter (nPanel=3, nSame=3, nEvent=3)
      PARAMETER (MXY=20, MAXREC=100, MAXBH=100, MAXRES=50, MAXBIN=161,  &
     &           MAXSPC=10, MAXOBS=100, MAXTH=10000, MAXTIM=20,         &
     &           MAXBP=5, MAXACT=1000, MAXTC=200, MAXNSI=5,             &
     &           MAXNDI=18, MAXNI=25, MAXNPI=18, MAXPNL=10, MAXPPP=5,   &
     &           MAXVAR=100, MAXREG=5, NEQM=MXY*MXY, IBWM=3*MXY+1)

      Integer E0Int, E1Int, E2Int, E12Int,                              &
     &           Same, Diff1, Diff2, Lower, Middle, Upper
      Parameter(E0Int=0,E1Int=1,E2Int=2,E12Int=3,                       &
     &           Same=1, Diff1=2, Diff2=3, Lower=1, Middle=2, Upper=3)

      Common /BAND/ NX, NY, NNODE, IBW, IBS(-1:1,-1:1,-1:1),            &
     &              IEQ(MXY,MXY)

      Logical BHFILE
      Common /BHOLE1/ IPP, NBH, NBHT, NBHTE, NBHGT, NBHGTE, NBHE1,      &
     &                NBHSMX, NBHBMX, NBHBP(MAXBP), NBHBPM(MAXBP),      &
     &                IXBH(MAXBH), IYBH(MAXBH), BHFILE

      Common /BLOWOUT_out_file/ BLOWOUT_out(2)
      Character*30 BLOWOUT_out

      Common/BlowData/ TMBLE0(NTMBLE0L), RBLE0, RBL,BlowoutTime,        &
     & CAVGD, CAVGDE, TMAVG, CAVG, NTBLs(NTMBLE1S)
      Double Precision TMBLE0
      Double Precision, Pointer :: BlowoutTime(:)
      Double Precision, Pointer :: RBLE0(:,:,:,:), TMAVG(:,:)
      Double Precision, Pointer :: RBL(:,:,:,:,:), CAVGD(:,:,:,:),      &
     &  CAVG(:), CAVGDE(:,:,:)

      Common /COLLOID_files/COLLOID_in(2)
      Character*30 COLLOID_in

      Common /CONTROL/ TEND
       Double Precision TEND
      Material = 2
      nScaleTimes = NTMCH
       iTime=0
       dTime=0
       Scenario=0
       Panel=0

!.....Read spallings release header record
      IfLAG=1
      read(unit,1001) IOBS_c, IfLAG_c !, 'SPALLINGS RELEASE TABLES'

!.....Read concentration values for spallings releases
      read(unit,1004) nScaleTimes    ! 'Number of times for Spallings concentrations'
       if (first) then
               allocate(tmavg(1:100,1:nScaleTimes),cavg(1:nScaleTimes))
               allocate(RBLE0(1:100,1:NTMBLE0L,LOWER:UPPER,1:2))
               Allocate(BlowoutTime(1000))
           allocate(rbl(1:100,1:1000,same:diff2,e1int:e2int,1:2))
       end if

      read(unit,3001) label   !'Times for spallings concentrations'
      read(unit,1003) (TMAVG(IOBS,IT),IT=1,nScaleTimes)
      read(unit,3001) label   !'Spallings concentrations'
      read(unit,1003) (CAVG(IT),IT=1,nScaleTimes)
       do IT=1,nScaleTimes
           write(10,*)IRep,",",SType(4),",",Pnl(Panel),",",Vec,",",iTime,
     &         ",",dTime,",",IT,",",TMAVG(IOBS,IT),",","Spall_Conc",",",
     &         CAVG(IT)
       end do


      Do PANEL=LOWER,UPPER      !Loop across both panels
!........Read spallings releases for E0 intrusions
          read(unit,3001) label       !'Spallings releases for E0 conditions
PNAME(PANEL)
          read(unit,1004) NTMBLE0L_c !, 'Number of times for spallings 'releases, E0
conditions ',pname(panel)
          read(unit,1003) (TMBLE0(IT),IT=1,NTMBLE0L)
          read(unit,1003) (RBLE0(IOBS,IT,PANEL,Material),IT=1,NTMBLE0L)
               do IT=1,NTMBLE0L
               write(10,*) IRep,",",SType(4),",",Pnl(Panel),",",Vec,",",
     &         ",", dTime,",",IT,",",TMBLE0(IT),",",
     &          iTime,"Spallings",",",
     &          RBLE0(IOBS,IT,PANEL,Material)
               end do
      End Do

      Do Event=E1INT,E2INT      !Loop across E1 and E2 events
          lMatch =  12
          Do MATCH=SAME, DIFF2   !Consider both same and different panels
!..............Read spallings releases
               read(unit,3001) label !'Spallings releases for EventID(Event)
conditions ',MatchName(MATCH)(1:lMatch)
               read(unit,1004) NTMBLE1S_c    !, 'Number of time series for spallings
releases, ', EventID(Event),MatchName(match)(1:lMatch)
               iStart = 1
               Do IT1=1,NTMBLE1S        !Loop across all times
                   read(unit,1005) NTBLs(IT1)        !, 'Number of times in series',
IT1
                   read(unit,1003) (BlowoutTime(IT2),IT2=iStart,        &
     &                         iStart+NTBLs(IT1)-1)
                   read(unit,1003)(RBL(IOBS,IT2,MATCH,Event, Material), &
     &                         IT2=iStart,iStart+NTBLs(IT1)-1)
                               do IT2=iStart,iStart+NTBLs(IT1)-1
                                   write(10,*)IRep,",",stype(Event),",",Mtch(Match),",
     &                ",Vec,",",iTime,",", dTime,",",IT2,",",
     &                BlowoutTime(IT2),",","Spallings",",",
     &                        RBL(IOBS,IT2,MATCH,Event, Material)
                                end do
                   iStart = iStart + NTBLs(IT1)
               End Do
               lMatch=17
          End Do
      End Do
        first=.false.
      Return

!.....I/O error encountered reading spallings release tables
 9100 Continue
```

```
        Do ICH=1,NCH
          read(unit,1003) (CCH(ICH,IT),IT=1,NTMCH)
               do IT=1,NTMCH
           write(wStream,"(I4)") ICH
           write(10,*) IRep,",",SType(4),",",Panel,",",Vec,",",iTime,
     &        ",",dTime,",",IT,",",TMCH(IT),",",
     &         "CCH_Activity_"//trim(wStream),",",CCH(ICH,IT)
               end do
        End do
        read(unit,2001) label !"RH waste stream activities"
        Do IRH=1,NRH
          read(unit,1003) (CRH(IRH,IT),IT=1,NTMRH)
           write(wStream,"(I4)") IRH
               do IT=1,NTMRH
              write(10,*) IRep,",",SType(4),",",Panel,",",Vec,",",iTime,
     &        ",",dTime,",",IT,",",TMRH(IT),",",
     &        "CRH_Activity_"//trim(wStream),",",CRH(IRH,IT)
               end do
        End do
      End If
       first=.false.
      Return

!.....I/O error encountered reading cuttings release tables
 9100 Continue
      Write(6,*) '>>>>>ERROR(S) ENCOUNTERED, EXECUTION TERMINATED 98'
      Stop

!.....Premature eof encountered reading cuttings release tables
 9200 Continue
      Write(6,*) '>>>>>ERROR(S) ENCOUNTERED, EXECUTION TERMINATED 99'
      Stop

!.....Format statements
 1001 Format(2I10,1X,A)
 1002 Format(2I10,5E10.3)
 1003 Format(20(E10.3))
 2001 Format(A)
      End

!--------------------------------------------------------------------------------

      Subroutine RWSPL (IOBS, unit)
!     Author: T. B. Kirchner (Revision)
!     Purpose:Write spallings release tables
      Include 'ccgf1_preccdfgf.fi'
       common/replicate/IREP,Vec,SType(0:4)
       Character*2 IREP
       Character*4 Vec
       Character*3 SType
      Integer IOBS           !Input: number of realization to be printed
       Integer unit            !Input: unit number for output


!.....Local variables
      Character*14 PNAME(NPANEL)
      Character*17 MatchName(nSame)
      Character*2 EventID(2)
      Integer Material, Panel, Event, Match, iStart
       character*100 label
       logical first
       COMMON/RWSLP_C/FIRST
       integer Scenario, iTime
       double precision dTime
       character*13 Mtch(3)
       character*12 PNL(0:3)

      Data Pnl/'NA','LOWER','MIDDLE','UPPER'/
       data Mtch/"Same","Adjoining","Non-adjoining"/

       data first/.TRUE./


      Data MatchName/"(SAME PANEL)","(DIF1 PANEL)","(DIF2 PANEL)"/

      PNAME(LOWER) = '(LOWER PANELS)'
      PNAME(MIDDLE) = '(MIDDL PANELS)'
      PNAME(UPPER) = '(UPPER PANELS)'
      EventID(E1INT)='E1'
      EventID(E2INT)='E2'
```

```
!-----------------------------------------------------------------------------

      Subroutine RWCUT (IOBS, unit)
!     Author: T. B. Kirchner (Revision)
!     Purpose:Read and write cuttings release tables
      Include 'ccgf1_preccdfgf.fi'
       common/replicate/IREP,Vec,SType(0:4)
       Character*2 IREP, Panel
       Character*4 Vec
       Character*3 SType
       integer Scenario,  iTime
       double precision dTime

      Integer IOBS          !Input: number of realization to be printed
      Integer unit          !Input: unit number for output
       character*100 label
       character*4 wStream
       double precision ACH_c, DRILDIAM_c, ARH_c
       common/rwcut_c/first
       logical first
       data first/.true./
       iTime=0
       dTime=0.0
       Scenario=0
       Panel="NA"
!
!.....Read cuttings release title
      If (IOBS .EQ. 1) Then
         IFLAG=1
      Else
         IFLAG=0
      End If

      read(unit,1001) IOBS_c, IFLAG_c,label  !'CUTTINGS RELEASE TABLE'
!.....Read number of times, number of waste types, number of waste
!     types to average, waste area, and drill bit diameter
!     For CH waste
      read(unit,1002) NTMCH, NCH, ACH_c, DRILDIAM_c
       if (first) then
               allocate(pch(1:NCH),TMCH(1:NTMCH),CCH(1:NCH,1:NTMCH))
       end if

!.....Read number of times, number of waste types, number of waste
!     types to average, waste area, and waste height
!     For RH waste
      read(unit,1002) NTMRH, NRH, ARH_c
       if (first) then
           allocate(PRH(0:NRH),TMRH(1:NTMRH),CRH(1:NRH,1:NTMRH))
       end if

      If (IOBS .EQ. 1) Then         !Write data for the first realization
!........Read probabilities associated with each activity level for
!        CH and RH wastes
        read(unit,2001) label !"Probability of hitting each CH waste stream"
         read(unit,1003) (PCH(ICH),ICH=1,NCH)
        do ICH=1,NCH
           write(wStream,"(I4)") ICH

           write(10,*) IRep,",",SType(4),",",Panel,",",Vec,",",iTime,
     &         ",",dTime,",",iTime,",",dTime,",","Prob_CH"//trim(wStream)
     &          ,",",PCH(ICH)
        end do

        read(unit,2001) label !"Probability of hitting each RH waste stream"
         read(unit,1003) (PRH(IRH),IRH=1,NRH)
        do IRH=1,NRH
           write(wStream,"(I4)") IRH
           write(10,*) IRep,",",SType(4),",",Panel,",",Vec,",",iTime,
     &        ",",dTime,",",iTime,",",dTime,",","Prob_RH"//trim(wStream),
     &          ",",PRH(IRH)
        end do

!........Read times for activity levels for CH waste
        read(unit,2001) label !"Times for CH waste stream activities"
         read(unit,1003) (TMCH(IT),IT=1,NTMCH)
!........Read times for activity levels for RH waste
        read(unit,2001) label !"Times for RH waste stream activities"
         read(unit,1003) (TMRH(IT),IT=1,NTMRH)
!........Read releases for all times for CH and RH waste
        read(unit,2001) label !"CH waste stream activities"
```

```
          read(unit,1014) WUF_curies  !'Total activity in Curies'
          Do ICH=1,NDCHAIN       !Loop across decay chains
              Call WriteRad(RadSpecies(ICH), unit)
          End Do

!........ Write transport from culebra release times
          read(unit,1013) NTMGWTP    !'Number of times for Culebra releases'
               NTMGWTP=NTMGWTP+1      !correct for -1 when written
               if (first) then
                      allocate(TMGWTR(1:NTMGWTP))
               end if
          read(unit,1012) (TMGWTR(ITM),ITM=1,NTMGWTP-1)
      End If

!.....Read transport from culebra releases for radionuclides
!.....Written in reverse time order so that CCDFGF doesn't have to
!.....reverse the array to calculate releases from the Culebra
      nStates=5
        if (first) then
               allocate(RRADAE(1:100,1:nStates,1:200,Partial:Fully))
        end if
        Scenario=0
        Panel="NA"
      Do I=1,nStates     !Do the first 4 primary radionuclides
              Do IM = Partial,Fully         !Loop across unmined and mined
                    read(unit,1021) RadNames(I)! MINED(IM)
                    read(unit,1012)(RRADAE(IOBS,I,ITM,IM),              &
     &                         ITM=NTMGWTP-1,1,-1)
                               do ITM=NTMGWTP-1,1,-1
                        if (I.ne.5) Then
                                       write(10,*)
IRep,",",Mined(IM),",",Panel,",",Vec,
     &                 ",",0,",",TMGWTR(1),",",ITM,",",TMGWTR(ITM),",",
     &                 "MT2"//RadNames(I),",",
     &                      RRADAE(IOBS,I,ITM,IM)
                                  else
                                       write(10,*)
IRep,",",Mined(IM),",",Panel,",",Vec,
     &                 ",",0,",",TMGWTR(1),",",ITM,",",TMGWTR(ITM),",",
     &                 "MT2Th23a",",",
     &                      RRADAE(IOBS,I,ITM,IM)
                                      end if
                                end do
              End Do
      End Do

        first=.false.
      Return

!.....Format statements
 1001 Format(A)
 1011 Format(2I10,A)
 1012 Format((8E10.3))
 1014 Format(E10.3,1x,A)
 1013 Format(I10,1x,A)
 1021 Format(31x,A)
      End

!--------------------------------------------------------------------------------

      Recursive Subroutine WriteRad(Rad, unit)
!     Author: T. B. Kirchner
!     Purpose: Write the radionuclide data

      Integer unit          !Input: unit number for output
        character*100 label
      Include 'ccgf1_preccdfgf.fi'
      Type (Radionuclide) Rad
      read(unit,1001) label
      read(unit,1002) Rad.Name, Rad.DecayRate, Rad.KgramToMoles,        &
     &              Rad.SpecificActivity, Rad.ReleaseLimit
!.    This code is no longer required
!     if (Associated(Rad.Daughter)) Then
!         Call WriteRad(Rad.Daughter)
!     End if
      Return
 1001 Format(A)
 1002 Format(A5,1x,4E10.3)
      End
```

```
              read(unit,1002) label
              read(unit,1012) (RSGWRP(IOBS,IT2,ITM,I),                  &
     &           ITM=1,NTMREL(IT2,I))
                      do ITM=1,NTMREL(IT2,I)
                         write(10,*)IRep,",",SType(I),",",Panel,",",Vec,",",IT2,
     &            ",",TMGWRP(1,IT2,1),",",ITM,",",TMGWRP(ITM,IT2,1),",",
     &              "EPALWMBT",",",(RAGWRP(IOBS,IT2,ITM,I))
                         write(10,*)IRep,",",SType(I),",",Panel,",",Vec,",",IT2,
     &            ",",TMGWRP(1,IT2,1),",",ITM,",",TMGWRP(ITM,IT2,1),",",
     &              "DeweyLake",",",RDGWRP(IOBS,IT2,ITM,I)
                         write(10,*)IRep,",",SType(I),",",Panel,",",Vec,",",IT2,
     &            ",",TMGWRP(1,IT2,1),",",ITM,",",TMGWRP(ITM,IT2,1),",",
     &              "Surface",",",RSGWRP(IOBS,IT2,ITM,I)
                      end do
         End Do
      End Do
       first=.false.
      Return

!.....I/O error encountered reading flows to culebra release tables
 9100 Continue
      Write(6,*) '>>>>>ERROR(S) ENCOUNTERED, EXECUTION TERMINATED 94'
      Stop
!.....Premature eof encountered reading flows to culebra release tables
 9200 Continue
      Write(6,*) '>>>>>ERROR(S) ENCOUNTERED, EXECUTION TERMINATED 95'
      Stop
!.....Format statements
 1001 Format(i10)
 1002 Format(A)
 1003 Format(i10, 1x, A, i3, 1x, i3,1x, i3)
 1011 Format(2I10,A)
 1012 Format((10E10.3))
 1013 Format((10I10))
 1014 Format(2x,a8,i2,10(E10.3))
 1015 Format(12x,10(a10))
 1016 Format(11x,A)
 1021 Format(A)
 1022 Format(A)
      End

!-------------------------------------------------------------------------------

      Subroutine RWCULT (IOBS, unit)
!     Author: T. B. Kirchner (Revision)
!     Purpose:Write summary tables for releases from culebra
      Include 'ccgf1_preccdfgf.fi'
       common/replicate/IREP,Vec,SType(0:4)
       Character*2 IREP
       Character*4 Vec
       Character*3 SType
       integer iTime
       Character*2 Panel
       double precision dTime
       integer Scenario

      Integer IOBS          !Input: number of realization to be printed
       Integer unit          !Input: unit number for output

      Character*2 MINED(2)
      Dimension NC(2)
      Integer Partial, Fully, IMBR
       double precision WUF_curies


       common/RWCULT_C/first
       logical first
       data first/.true./

!     Initialize storage for local variables
      Data MINED / 'MP', 'MF' /
      Data NC / 10, 10 /
      Data Partial/1/, Fully/2/

!.....Read summary release tables
      IFLAG=1
      read(unit,1011) IOBS_c, IFLAG_c, label

      If (IOBS .EQ. 1) Then         !Write a header before the first realization
!........ Write header records
```

```
     &              iTime,",",dtime,",",iTime,",",dtime,",",
     &                   trim(Radionuclides(IDC))//"_FRC"//cType(ICS),",",
     &              FRDCLE(IOBS,ICS,IDC,I)
                       end do
          End Do
      End Do

!.....Flows to culebra, marker beds, dewey lake, and surface releases
!     Flows to culebra releases
      read(unit,1022) label
      read(unit,1013) nNutsTimes
      Do IDC=1,NDCHAIN     !Loop across radionuclides
               read(unit,1016) Radionuclides(IDC)
          read(unit,1012) (RCGWRPE0(IOBS,IDC,ITM),                         &
     &                    ITM=1,nNutsTimes)
            do ITM=1,nNutsTimes
                write(10,*)IRep,",",SType(E0INT),",",Panel,",",Vec,",",
     &               iTime,",",dtime,",",ITM,",",TMGWRPE0(ITM),",",
     &               "A00"//Radionuclides(IDC),",",
     &               RCGWRPE0(IOBS,IDC,ITM)
          end do
      End Do

!     Marker bed releases
      read(unit,1002) label    !Marker beds
      read(unit,1012) (RAGWRPE0(IOBS,ITM) ,ITM=1,nNutsTimes)
      read(unit,1002) label   !Dewey lake
      read(unit,1012) (RDGWRPE0(IOBS,ITM) ,ITM=1,nNutsTimes)
      read(unit,1002) label    !surface
      read(unit,1012) (RSGWRPE0(IOBS,ITM) ,ITM=1,nNutsTimes)
        do ITM=1,nNutsTimes
            write(10,*)IRep,",",SType(E0INT),",",Panel,",",Vec,",",iTime,
     &               ",",dtime,",",ITM,",",TMGWRPE0(ITM),",",
     &               "EPALWMBT",",",RAGWRPE0(IOBS,ITM)
            write(10,*)IRep,",",SType(E0INT),",",Panel,",",Vec,",",iTime,
     &               ",",dtime,",",ITM,",",TMGWRPE0(ITM),",",
     &               "DeweyLake",",",RDGWRPE0(IOBS,ITM)
            write(10,*)IRep,",",SType(E0INT),",",Panel,",",Vec,",",iTime,
     &               ",",dtime,",",ITM,",",TMGWRPE0(ITM),",",
     &               "Surface",",",RSGWRPE0(IOBS,ITM)
        end do
      if (first) then
                allocate(RCGWRP(1:100,1:NTM,1:NDCHAIN,1:200,1:E12INT))
                allocate(RAGWRP(1:100,1:NTM,1:200,E1INT:E12INT))
                allocate(RDGWRP(1:100,1:NTM,1:200,E1INT:E12INT))
                allocate(RSGWRP(1:100,1:NTM,1:200,E1INT:E12INT))
        end if

      Do I=E1INT,E12INT     !Loop across E1, E2 and E1E2 events
          read(unit,1022) INTRUS(I)(1:NC(I))
                if (I.eq.E12INT) then
                       Pfix="LDC"                    !the Panel prefix
                else
                      PFix="A00"                     !the Nuts prefix
                end if
          Do IT2=1,NTM
!.....         Flows to culebra releases
               read(unit,1003) NTMREL(IT2,I)   !Number of times in series ',IT2
               Do IDC=1,NDCHAIN  !Loop across radionuclides
                   read(unit,1016) Radionuclides(IDC)
                   read(unit,1012) (RCGWRP(IOBS,IT2,IDC,ITM,I),            &
     &              ITM=1,NTMREL(IT2,I))

!................intrusion times are first time in table
                               do ITM=1,NTMREL(IT2,I)
                              write(10,*)IRep,",",stype(I),",",Panel,",",Vec,",",
     &               IT2,",",TMGWRP(1,IT2,I),",",ITM,",",
     &               TMGWRP(ITM,IT2,I),",",Pfix//Radionuclides(IDC),",",
     &                RCGWRP(IOBS,IT2,IDC,ITM,I)
                               end do
               End Do
!.....         Marker bed releases
               read(unit,1002) label
               read(unit,1012) (RAGWRP(IOBS,IT2,ITM,I),                    &
     &            ITM=1,NTMREL(IT2,I))
!.....         Dewey lake releases
               read(unit,1002) label
               read(unit,1012) (RDGWRP(IOBS,IT2,ITM,I),                    &
     &            ITM=1,NTMREL(IT2,I))
!.....         Surface releases
```

```
      Integer IOBS          !Input: number of realization to be printed
       Integer unit          !Input: unit number for output
       common/replicate/IREP,Vec,SType(0:4)
       Character*2 IREP, Panel
       Character*4 Vec
       Character*3 SType
       integer iTime
       double precision dTime

!.....Local variables
      Character*4 INTRUS(0:3)
       character*100 label
       character*8 colloid
       character*3 cType(3)
       character*3 Pfix
       integer nTimeSeries,nDaughters_c(6)
      Dimension NC(0:3)
      Integer Nuc
       common/rwculf_c/first
       logical first
       data first/.true./

      Data INTRUS / 'E0', 'E1', 'E2', 'E1E2' /
      Data NC / 2, 2, 2, 4 /
       data cType/"MIC","INT","MIN"/
!
      iTime=0
       dTime=0.0
       Panel="NA"

!.....Read summary release tables
      IFLAG=1
      read(unit,1011) IOBS_c, IFLAG_c, Label
      If (IOBS .EQ. 1) Then         !Write a header before the first realization
!.........Read header records
          read(unit,1001) NCOLSP_c !Number of colloid species'
          read(unit,1001) NDCHAIN_c !Number of radionuclide chains'
               read(unit,1002) label
          read(unit,1013) (nDaughters_c(Nuc),Nuc=1,NDCHAIN-1)
               read(unit,1002) label
          Do Nuc=1,NDCHAIN      !Loop across all radionuclides
              read(unit,1002) Radionuclides(Nuc)
          End Do

!.........Flows to Culebra times for E0, E1, E2, and E1E2 intrusions
          read(unit,1021) Label
          read(unit,1001) nTimeSeries        !Number of time series'
          read(unit,1001) nNutsTimes         !Number of times in this series'
               if (first) then
                       allocate(TMGWRPE0(1:nNutsTimes),
     &             TMGWRP(1:200,1:NTM,1:NCOLSP))
                       allocate(FRDCLE(1:100,1:NCOLSP,1:NDCHAIN,0:3))
                       allocate(RCGWRPE0(1:100,1:NDCHAIN,1:nNutsTimes))
                       allocate(RAGWRPE0(1:100,1:nNutsTimes))
                       allocate(RDGWRPE0(1:100,1:nNutsTimes))
                       allocate(RSGWRPE0(1:100,1:nNutsTimes))
          end if
               read(unit,1012) (TMGWRPE0(IT1),IT1=1,nNutsTimes)
          Do I=E1INT,E12INT      !Loop across E1, E2 and E1E2 events !<==== was labeled
"colloids"
              read(unit,1021) label
              read(unit,1003) NTM_c   !'Number of time series'
              Do IT2=1,NTM_c  !Loop across times
                  read(unit,1003) NTMREL(IT2,I)
                  read(unit,1012) (TMGWRP(IT1,IT2,I),IT1=1,              &
     &                                NTMREL(IT2,I))
              End Do
          End Do
      End If

!.....Species fractions attached to each colloid for E0, E1, E2, and
!.....E1E2 intrusions
      read(unit,1002) label   !Species colloid fractions'
      read(unit,1015) (Radionuclides(i),i=1,NDCHAIN)
      Do ICS=1,NCOLSP       !Loop across the colloid types
          Do IDC=1,NDCHAIN  !Loop across radionuclides      -      "Colloid" is label
              read(unit,1014) colloid, ics_c,                            &
     &            (FRDCLE(IOBS,ICS,IDC,I),I=0,3)
                       do I=0,3
                     write(10,*) IRep,",",SType(I),",",Panel,",",Vec,",",
     &             (CAVGD(IOBS,IT,I,J),IT=1,nScaleTimes)
                do IT=1,nScaleTimes
                 write(10,*) IREP,",",SType(I),",",Pnl(Panel),",",Vec,
     &                       ",",iTime,",",dTime,",",IT,",",
     &                                 TMAVG(IOBS,IT),",","CLE"//RadNames(j),
     &                      ",",CAVGD(IOBS,IT,I,J)
            end do
          End Do
         read(unit,1010) (CAVGDE(IOBS,IT,I),IT=1,nScaleTimes)
          do IT=1,nScaleTimes
              write(10,*) IREP,",",SType(I),",",Pnl(Panel),",",Vec,",",
     &                 iTime,",",dTime,",",IT,",",TMAVG(IOBS,IT),",",
     &                        "CNETOTAL",",",CAVGDE(IOBS,IT,I)
          end do
      End Do

!.....Read blowout releases E0 for intrusions
      Do Panel=Lower, Upper          !Loop across both panels
          read(unit,2001) label
          read(unit,1002) NTMBLE0L_c
          read(unit,1003) (TMBLE0(IT),IT=1,NTMBLE0L_c)
          read(unit,1003)(RBLE0(IOBS,IT,Panel,Material),IT=1,NTMBLE0L_c)
               do IT=1,NTMBLE0L_c
!...........Note that the initial intrusion time is being written as the "subsequent"
!            intrusion time to make the comparison with the CDB data easier. Since
this
!                 is an E0 event there was no previous intrusion so in reality these
are all
!            initial intrusions.
              write(10,*) IREP,",",SType(E0INT),",",Pnl(Panel),",",Vec,
     &           ",",iTime,",",dtime,",",IT,",",TMBLE0(IT),",",
     &           "BRIN_REL",",",RBLE0(IOBS,IT,Panel,Material)
               end do
      End Do

!.....Read blowout releases for E1 and E2 intrusions
      Do Event = E1INT, E2INT         !Loop across E1 and E2 events
          Do Panel=Same, Diff2        !consider both same and different panels
              read(unit,2001) label
              read(unit,1002) NTMBLE1S_c
              iStart=1
              Do IT1=1,NTMBLE1S      !Loop across times
                   read(unit,1002) NTBLs(IT1)
                   read(unit,1003) (BlowoutTime(IT2),IT2=iStart,        &
     &                         iStart+NTBLs(IT1)-1)
                   read(unit,1003) (RBL(IOBS,IT2,Panel,Event,Material), &
     &                         IT2=iStart,iStart+NTBLs(IT1)-1
                               do IT2=iStart,iStart+NTBLs(IT1)-1
                                  write(10,*) IREP,",",SType(Event),",",Adj(Panel),
     &                ",",Vec,",",IT1,",",BlowoutTime(iStart),",",
     &                IT2-iStart+1,",",BlowoutTime(IT2),",","BRIN_REL",
     &                        ",",RBL(IOBS,IT2,Panel,Event,Material)
                               end do
                   iStart=iStart+NTBLs(IT1)
              End Do
          End Do
      End Do
       first=.false.
      Return

!.....Format statements
 1001 Format(2I10)
 1002 Format(I10)
 1005 Format(A5,2x,(20E10.3))
 1003 Format((20E10.3))
 1004 Format(I10)

 1010 Format(7x,(20E10.3))

 2001 Format(A)
 3001 Format(A, A)
      End

!-------------------------------------------------------------------------------

      Subroutine RWCULF (IOBS, unit)
!     Author: T. B. Kirchner (Revision)
!     Purpose: Read and write summary tables for releases to culebra, marker beds,
!              Dewey lake, and surface
      Include 'ccgf1_preccdfgf.fi'
```

```
!........Read and write spallings release tables
         Call RWSPL (IOBS, unit)
!........Read and write blowout release tables
         Call RWBLW (IOBS, unit)
!........Read and write marker beds, dewey lake, surface, and culebra
!........Release tables
         Call RWCULF (IOBS, unit)
!........Read and write culebra transport release tables
         Call RWCULT (IOBS, unit)

      End Do
       close(10)
 1001 Format(I10)
 4001 Format(a1)
 4002 FORMAT(30x, i4)
      Close (unit)
      Stop

      End

!--------------------------------------------------------------------------------

      Subroutine RWBLW (IOBS, unit)

!     Author: T. B. Kirchner
!     Purpose:Read and write blowout release tables

      Include 'ccgf1_preccdfgf.fi'
       common/replicate/IREP,Vec,SType(0:4)
       Character*2 IREP
       Character*4 Vec
       Character*3 SType

      Integer IOBS            !Input: number of realization to be printed
      Integer unit            !Input: unit number for output

!.....Local variables
      Character*15 Pnl(0:3)
       Character*12 Adj(3)
      Integer pLen(2,3)
      Integer Panel, Event
      Integer Material
      Integer I, J
       integer iTime
       double precision dTime
       character*100 label
       common/rwblw_c/first
       logical first
       data first/.true./

      Data pLen/43,50,51,43,50,51/
      Parameter (Material = 1)

      Data Pnl/'NA','L','M','U'/
       Data Adj/'L','M','U'/  !'Same','Adjacent','Non-adjacent'/


!.....Read blowout release header record
      IfLAG=1
      Read(unit,1001) IOBS, IfLAG
!.....Read concentration values for blowout releases
      Read(unit,2001)label

      Read(unit,1004) nScaleTimes
       if (first) then
               allocate (cavgd(1:100,1:nScaleTimes,E0INT:E2INT,
     &                    1:nRadionuclides))
               allocate(CAVGDE(1:100,1:nScaleTimes,E0INT:E2INT))
       end if

      Read(unit,1010) (TMAVG(IOBS,IT),IT=1,nScaleTimes)

!.....Initialize place-holder variables
       Panel=0
       iTime=0
       dTime=0.0

      Do I=E0INT, E2INT           !Loop across E0, E1 and E2 events
         Do J=1,nRadionuclides
           read(unit,1005) RadNames(J),                                 &
```

APPENDIX B READRELTAB UTILITY SOURCE LISTING

The ReadReltab Code

```
!-------------------------------------------------------------------------------
      Program ReadReltab
!     Author: T. B. Kirchner
!     Purpose: ReadReltab reads the "RELTAB" release file produced by CCFDGF
!                    and exports the data in a comma-delimited format suitable for
!                    import into Access. This program was created by editing the
PRECCDFGF
!            code and uses its include file.

      Implicit Double Precision (A-H, O-Z)

!.....Summarize cuttings, spallings, blowout, nuts, and secotp data and
!     save on permanent file
      Call RELTAB

!.....Close the error log
      Close(11)
      End

!-------------------------------------------------------------------------------


!-------------------------------------------------------------------------------

      Subroutine RELTAB
!     Author: T. B. Kirchner (Revision)
!     Purpose:Read summary release tables to be used by ccdfgf
      Integer IOB
        Integer unit
        Character*100 label
        CHARACTER*25 fname
      Include 'ccgf1_preccdfgf.fi'

        common/replicate/IREP,Vec,SType(0:4)
        Character*2 IREP
        Character*4 Vec
        Character*3 SType

        Data unit/4/
        data fName/'ccgf_cra1bc_reltab_r1.dat'/
        data IRep/"R1"/
        data SType/"E0","E1","E2","E12","NA"/

        Data nRadionuclides/4/
!.....Open summary release table (output)
      OPEN (unit, file=fname, STATUS='OLD',recl=1610)
        open (10,file="c:\DB_Data.txt", recl=200)

!.....Write the file header
        Write(10,*) "Replicate",",","Scenario",",","Cavity",",","Vector",
     &           ",","IntrusionTimeIndex",",","IntrusionTime",",",
     &           "SeriesTimeIndex",",","SeriesTime",",",
     &           "ReleaseName",",","Value"

!       Skip the "constants" section
        Do i=1,34
            read(unit,4001) label
        End do
!.....Read number of observations
      read(unit,1001) NOBS

!.....Loop over observations
      Do IOBS=1,NOBS
          Write(*,*) "Vector ",IOBS
!.......Read sampled parameter values for this observation
          Read(unit,4002)  IOB
!.......Save the vector number as a string
          write(Vec,"('v',I3.3)") IOBS

          Call ReadLHSVariables(IOBS, unit)
!.......Read and write cuttings release table
         Call RWCUT (IOBS, unit)
```

The following steps can be used to extract the release data from the PRECCDFGF "RELTAB" file and import it into the CheckReltab database.

Step 1. Extract the .zip file containing the ReadReltab.exe program (and Fortran source) from the CMS library and transfer it using FTP in binary mode to a Windows XP computer. Use the commands

```
$ libcra1bc_sumverify

$ cms fetch element cra1bc_readreltab.zip
```

The source code is also contained in the CMS library LIBCRA1BC_SUMVERIFY and can be extracted using:

```
$ libcra1bc_sumverify
$ cms fetch element cra1bc_readreltab.for
$ cms fetch element ccgf1_preccdfgf.fi
```

Step 2. Extract the file CCGF_CRA1BC_RELTAB_R1.DAT from the CMS library using the commands:

```
$ libcra1bc_ccgf
$ cms fetch element ccgf_cra1bc_reltab_r1.dat
```

Step 3. Transfer the file from the VMS system to the PC using FTP in ASCII mode. The file should be placed in the directory containing the program ReadReltab.exe.

Step 4. Execute ReadReltab.exe
Step 5. Open the CheckReltab database using Microsoft® Access

Step 6. Select the "Load Reltab Data" button from the main form.

A.3 DATA COMAPRISON

The data are compared in the CheckReltab database by:

Step 1. Select the "Display Report" button from the main menu (Figure 3).

Step 2. Select a report for a particular code (Figure 4) to get the detailed comparison or the "Summary of Differences" report to get the mean and maximum differences for each code (equivalent to Table 10).

The script will create a subdirectory and a log file in the current working directory corresponding to each input file (WRK_DCOMP_NUT_CRA1BC_R1_S*s*, and DCOMP_NUT_CRA1BC_R1_S*s*.LOG, where $s \in \{1, 2a, 2b, 3a, 3b, 4a, 4b, 5a, 5b\}$).

Step 5. Extract the data from PANEL_INT CDB files.

```
$ @dcomp_panel_int dcomp_panel_int_cra1bc_r1.inp
```

The script will create a subdirectory in the current working directory which contains the script output files (WRK_DCOMP_PANEL_INT_CRA1BC_R1). The script will also produce a log file (DCOMP_PANEL_INT_CRA1BC_R1.LOG) in the current working directory.

Step 6. Extract the data from SECOTP2D CDB files.

```
$ @dcomp_st2d dcomp_st2d_cra1bc_r1.inp
```

The script will create a subdirectory in the current working directory which contains the script output files (WRK_DCOMP_ST2D_CRA1BC_R1). The script will also produce a log file (DCOMP_ST2D_CRA1BC_R1.LOG) in the current working directory.

Step 7. Extract the data from BRAGFLO_DBR CDB files:

```
$ @dcomp_dbr dcomp_dbr_cra1bc_r1_s1.inp
$ @dcomp_dbr dcomp_dbr_cra1bc_r1_s24.inp
$ @dcomp_dbr dcomp_dbr_cra1bc_r1_s35.inp
```

The script will create a subdirectory and a log file in the current working directory corresponding to each input file (WRK_DCOMP_DBR_CRA1BC_R1_S*s*, and DCOMP_DBR_CRA1BC_R1_S*s*.LOG, where $s \in \{1, 24, 35\}$).

Step 8. Copy all of the output files to a single directory to facilitate transferring them to the PC:

```
$ create/dir output
$ copy [.wrk_dcomp*]dc*.out [.output]
```

Step 9. Transfer all of the output files to the PC (using FTP).

A.2 EXTRACTIING DATA FROM RELTAB FILE

APPENDIX A DATA EXTRACTION AND COMPARISON

Here we describe the data extraction and comparison steps in enough detail so that someone can repeat the exercise.

A.1 EXTRACTING DATA FROM CDB FILES

The following steps will allow one to reproduce the CDB data extraction process.

Step 1. Fetch the scripts and script input files from CMS:

```
$ libcra1bc_sumverify
$ cms fetch/generation=crab1c-0 dcomp*.com
$ cms fetch/generation=cra1bc-0 dcomp*.inp
```

Step2. Extract the data from PANEL_CON CDB files:

```
$ @dcomp_panel_con dcomp_panel_con_cra1bc_r1.inp
```

The script will create a subdirectory in the current working directory which contains the script output files (WRK_DCOMP_PANEL_CON_CRA1BC_R1). The script will also produce a log file (DCOMP_PANEL_CON_CRA1BC_R1.LOG) in the current working directory.

Step 3. Extract the data from PANEL_ST CDB files:

```
$ @dcomp_panel_st dcomp_panel_st_cra1bc_r1.inp
```

The script will create a subdirectory in the current working directory which contains the script output files (WRK_DCOMP_PANEL_ST_CRA1BC_R1). The script will also produce a log file (DCOMP_PANEL_ST_CRA1BC_R1.LOG) in the current working directory.

Step 4. Extract the data from NUTS CDB files:

```
$ @dcomp_nut dcomp_nut_cra1bc_r1_s1.inp
$ @dcomp_nut dcomp_nut_cra1bc_r1_s2a.inp
$ @dcomp_nut dcomp_nut_cra1bc_r1_s2a.inp
$ @dcomp_nut dcomp_nut_cra1bc_r1_s3a.inp
$ @dcomp_nut dcomp_nut_cra1bc_r1_s3b.inp
$ @dcomp_nut dcomp_nut_cra1bc_r1_s4a.inp
$ @dcomp_nut dcomp_nut_cra1bc_r1_s4b.inp
$ @dcomp_nut dcomp_nut_cra1bc_r1_s5a.inp
$ @dcomp_nut dcomp_nut_cra1bc_r1_s5b.inp
```

WIPP PA. 2005e. Validation Document for PRECCDFGF Version 1.01. ERMS 539296, Sandia National Laboratories, Carlsbad, NM.

WIPP PA. 2005f. Validation Document for SUMMARIZE Version 3.00. ERMS 540108, Sandia National Laboratories, Carlsbad, NM.

```
       Character*120 LHS_VarNames

      Character*40 ComFileName
      Common /LogicalComFile/ ComFileName

      Integer nNames                        !The number of names stored
      Character*40 Map                  !Array of names and logicals
      Common /FileMap/ Map(2,200), nNames   !Logical and file name storage

      Integer NOBS
      Common /NumObs/ NOBS

      Common /NODE4/ FRA(MXY,MXY), CFRA(MXY,MXY), FRE(MXY,MXY),          &
     &               PBP(MXY,MXY)

      Common/NutData/ TMGWRPE0, TM(NTM, nEvent), TMGWRP, RCGWRPE0,       &
&
     &          RAGWRPE0, RCGWRP,RAGWRP,RDGWRPE0,RDGWRP,RSGWRPE0,RSGWRP,&
     &          FRDCLE, nNutsTimes, NTMREL(NTM, nEvent)
      Double Precision TM
      Double Precision, Pointer :: FRDCLE(:,:,:,:) ,RCGWRPE0(:,:,:),     &
     &      RAGWRPE0(:,:),RAGWRP(:,:,:,:), RCGWRP(:,:,:,:,:),            &
     &      RDGWRP(:,:,:,:),RSGWRP(:,:,:,:),RSGWRPE0(:,:),TMGWRPE0(:),   &
     &      RDGWRPE0(:,:), TMGWRP(:,:,:)
      Integer nNutsTimes, NTMREL

      Common /PNL_con_files/PNL_con_in(2)
      Character*30 PNL_con_in

      Common /NUTS_files/ NUTS_S1_in, NUTS_S23_in(7,3)
      Character*30 NUTS_S1_in, NUTS_S23_in

      Character*30 NUTS_out
      Common /NUTS_out_file/ NUTS_out

      Common /PSDB/SDBVAL(20)
      Double Precision SDBVAL
      Type Radionuclide
        Character*5 Name
        Double Precision AtomicMass
        Double Precision DecayRate
       Double Precision HalfLife
        Double Precision SpecificActivity
        Double Precision ReleaseLimit
        Double Precision KgramToMoles
        Type (Radionuclide), Pointer:: Daughter
      End Type

      Common /RadSpecies/Radionuclides(4), RadNames(5)
      Character*5 Radionuclides

      Common /RadParameters/nRadionuclides, nStates, RadSpecies
      Integer nRadionuclides, nStates
      Character Radnames*5
      Type (Radionuclide) RadSpecies(4)
!     RadNames includes the names of daughters as well as the original radionuclides
!     in the waste, hence it is larger than the list of radionuclides considered in
!     the models.

      Common /RELTAB_file/ RELTAB_out
      Character*30 RELTAB_out

      Common/SEC/RCOLAE,TMGWTR, NTMGWTP, RRADAE
      Double Precision, Pointer :: RRADAE(:,:,:,:)
      Double Precision, Pointer :: RCOLAE(:,:,:,:)
      Real, Pointer::TMGWTR(:)
      Integer NTMGWTP

      Common /ST2D_file/ ST2D_in(2)
      Character*30 ST2D_in

      Common /SECOPT_file/ SECOTP_out
      Character*30 SECOTP_out
C CMS REPLACEMENT HISTORY, Element CCGF1_PRECCDFGF.INC
C *1     3-JUN-2003 12:29:18 APGILKE "PRECCDFGF Version 1.00"
C CMS REPLACEMENT HISTORY, Element CCGF1_PRECCDFGF.INC
```

APPENDIX C READRELTAB UTILTY VALIDATION

Because the ReadReltab program is used simply to extract data from the PRECCDFGF RELTAB file and format it for importation into the CheckReltab database, validation of ReadReltab consists of verifying that the data in the RELTAB file are properly extracted. This validation was done by comparing data in the RELTAB_Data table to the corresponding data in the RELTAB file. These comparisons were made by loading the RELTAB file into a text editor and visually comparing the data to that in the RELTAB_Data table. Comparisons were made using non-zero values for examples of datasets for each of the codes. No differences were observed. In addition, the comparison of the RELTAB data to that extracted from the associated CDB files also constitutes a verification of the ReadReltab program.

APPENDIX D CHECKRELTAB DATABASE AND SQL QUERIES

Module UploadCode

```
Option Compare Database
Dim rep As String, scenario As String, IntrusionTime As Integer
Dim state As String, IntrusionTimeIndex As Integer, Vector As String
Dim ReleaseName As String, SeriesTimeIndex As Integer, SeriesTime As Double, SerTime
As Integer, Value As Double
Dim Cavity As String

Dim lrep As String, lscenario As String, lIntrusionTime As Integer
Dim lstate As String, lIntrusionTimeIndex As Integer, lVector As String
Dim lReleaseName As String, lSeriesTimeIndex As Integer, lSeriesTime As Double,
lSerTime As Integer, lValue As Double

Sub LoadCDB_Data(frm As Form)
'Load all the CDB files from the directory C:\CDB_Files
Dim db As Database
Dim nutsRaw As Recordset, table As Recordset
Dim filetable As Recordset

Close #1
Close #2
DoCmd.OpenQuery "Delete_CDB_FileNames"

Set db = CurrentDb

Set table = db.OpenRecordset("CDB_Data")
Set nutsRaw = db.OpenRecordset("CDB_RawData_Nuts")
Set filetable = db.OpenRecordset("CDB_FileNames")

First = True
ChDir "C:\CDB_Files"
Open "C:\CDB_Data.txt" For Output As #2
f = Dir("C:\CDB_Files\*.out")
While f <> ""
    frm.Loadname.Caption = f
    frm.Refresh
    'save the file name uploaded
    filetable.AddNew
    filetable.Fields(0) = f
    filetable.Update

    Open "C:\CDB_Files\" + f For Input As #1
    Line Input #1, s
    While Not EOF(1)
        Input #1, ReleaseName, rep, scenario, IntrusionTimeIndex, IntrusionTime,
Cavity, Vector, SeriesTimeIndex, SeriesTime, Value
        If Left(ReleaseName, 3) = "A00" Or Left(ReleaseName, 3) = "EPA" Then
            WriteRecord SeriesTime, Value, IntrusionTimeIndex, SeriesTimeIndex,
nutsRaw
            If IntrusionTime = 3000 And ReleaseName = "A00PU239" And scenario = "S3"
Then
                Debug.Print SeriesTime, Value
            End If
        Else
            WriteRecord SeriesTime, Value, IntrusionTimeIndex, SeriesTimeIndex, table
        End If
    Wend
    Close #1
    f = Dir()
Wend
Close #2
table.Close
nutsRaw.Close
filetable.Close

End Sub

Sub LoadReltab_Data(frm As Form)
```

```
'Load the reltab data. The file DB_Data.txt is created by running the Fortran program
'ReadReltab.

Dim db As Database
Dim nutsRaw As Recordset, table As Recordset

Close #1

Set db = CurrentDb

Set table = db.OpenRecordset("RELTAB_Data")

ChDir "C:\"
Open "C:\DB_Data.txt" For Input As #1
frm.Loadname.Caption = f
frm.Refresh
Line Input #1, s
While Not EOF(1)
    Input #1, rep, scenario, Cavity, Vector, IntrusionTimeIndex, IntrusionTime,
SeriesTimeIndex, SeriesTime, ReleaseName, Value
    WriteRecord SeriesTime, Value, IntrusionTimeIndex, SeriesTimeIndex, table
Wend
Close #1

table.Close

End Sub

Sub LoadCDB_FileNames(frm As Form)
'Load all the CDB file names from the directory C:\CDB_Files
Dim db As Database
Dim filetable As Recordset

DoCmd.OpenQuery "Delete_CDB_FileNames"
Set db = CurrentDb

Set filetable = db.OpenRecordset("CDB_FileNames")

ChDir "C:\CDB_Files"
f = Dir("C:\CDB_Files\*.out")
While f <> ""
    frm.Loadname.Caption = f
    frm.Refresh
    filetable.AddNew
    filetable.Fields(0) = f
    filetable.Update
    f = Dir()
Wend
filetable.Close


End Sub

Sub InterpolateNuts()
'Interpolate the NUTS data to 50 year intervals
Dim db As Database
Dim rec As Recordset, table As Recordset
Set db = CurrentDb
Set rec = db.OpenRecordset("CDB_RawData_Nuts")    'CDB_RawData_Nuts")
Set table = db.OpenRecordset("CDB_Data")
Dim IntIndex As Integer, SerIndex As Integer, t As Double, t_exp As Double
Dim v As Double

rec.MoveLast
rec.MoveFirst
rep = rec.Fields("Replicate")
scenario = rec.Fields("Scenario")
IntrusionTimeIndex = rec.Fields("IntrusionTimeIndex")
IntrusionTime = Nz(rec.Fields("IntrusionTime"), 0)
Vector = rec.Fields("Vector")
ReleaseName = rec.Fields("ReleaseName")
SeriesTimeIndex = rec.Fields("SeriesTimeIndex")
SeriesTime = rec.Fields("SeriesTime")
If SeriesTime > 20000 Then
    SerTime = SeriesTime * 0.00000003168877
Else
```

```
    SerTime = SeriesTime
    SeriesTime = SeriesTime / 0.0000003168877
End If
Value = rec.Fields("Value")
lrep = rep
lscenario = scenario
lstate = state
lIntrusionTimeIndex = IntrusionTimeIndex
lIntrusionTime = IntrusionTime
lVector = Vector
lReleaseName = ReleaseName
lSeriesTimeIndex = SeriesTimeIndex
lSerTime = SerTime
lSeriesTime = SeriesTime
lValue = Value

While Not rec.EOF And Not rec.BOF
    IntIndex = 1
    SerIndex = 1
    If SerTime <= 0 Then SerTime = 100
    If SerTime = 150 Then
        Debug.Print SerTime
    End If
    t_exp = SerTime
    t = t_exp / 0.00000003168877   't in seconds
    While t_exp <= 10000 And Not rec.EOF
        If t = SeriesTime Then      't_exp = SerTime Then
            WriteRecord t_exp, Value, IntIndex, SerIndex, table
            If Vector = "V030" And IntrusionTime = 3000 And ReleaseName = "A00TH230"
Then
                Debug.Print ReleaseName, t_exp, v, SerTime
            End If
            t_exp = t_exp + 50
            SerIndex = SerIndex + 1
            LoadRec rec
        ElseIf t < SeriesTime Then 't_exp < SerTime Then
            'interpolate to t_exp and save value
            'v = lValue + (Value - lValue) * (t_exp - lSerTime) / (SerTime - lSerTime)
            If SeriesTime = lSeriesTime Then 'first value, so rounding error is at
fault
                v = Value
            Else
                v = lValue + (Value - lValue) * (t - lSeriesTime) / (SeriesTime -
lSeriesTime)
            End If
            If Vector = "V030" And IntrusionTime = 3000 And ReleaseName = "A00TH230"
Then
                Debug.Print ReleaseName, t_exp, v, SerTime
            End If
            WriteRecord t_exp, v, IntIndex, SerIndex, table
            t_exp = t_exp + 50
            t = t_exp / 0.00000003168877   't in seconds
            SerIndex = SerIndex + 1
            If t_exp > 10000 Then
                LoadRec rec
            End If
        Else
            LoadRec rec
        End If
        If IntrusionTimeIndex <> lIntrusionTimeIndex Then IntIndex = IntIndex + 1
        If state <> lstate Or ReleaseName <> lReleaseName Or scenario <> lscenario
Then
            IntIndex = 1
            SerIndex = 1
            lrep = rep
            lscenario = scenario
            lstate = state
            lIntrusionTimeIndex = IntrusionTimeIndex
            lIntrusionTime = IntrusionTime
            lVector = Vector
            lReleaseName = ReleaseName
            lSeriesTimeIndex = SeriesTimeIndex
            lSerTime = SerTime
            lSeriesTime = SeriesTime
            lValue = Value
            t_exp = SerTime
            t = t_exp / 0.00000003168877    't in seconds
        End If
        If IntrusionTime <> lIntrusionTime Then
            IntIndex = IntIndex + 1
        End If
    Wend
Wend
End Sub

Sub LoadRec(rec As Recordset)
'Load a new record, preserving the previous values
    lrep = rep
    lscenario = scenario
    lstate = state
    lIntrusionTimeIndex = IntrusionTimeIndex
    lIntrusionTime = IntrusionTime
    lVector = Vector
    lReleaseName = ReleaseName
    lSeriesTimeIndex = SeriesTimeIndex
    lSerTime = SerTime
    lSeriesTime = SeriesTime
    lValue = Value

    rec.MoveNext
    If Not rec.EOF Then
        rep = rec.Fields("Replicate")
        scenario = rec.Fields("Scenario")
        IntrusionTimeIndex = rec.Fields("IntrusionTimeIndex")
        IntrusionTime = Nz(rec.Fields("IntrusionTime"), 0)
        Vector = rec.Fields("Vector")
        ReleaseName = rec.Fields("ReleaseName")
        SeriesTimeIndex = rec.Fields("SeriesTimeIndex")
        SeriesTime = rec.Fields("SeriesTime")
        If SeriesTime > 20000 Then
            SerTime = SeriesTime * 0.00000003168877
        Else
            SerTime = SeriesTime
            SeriesTime = CDbl(SeriesTime) / 0.00000003168877
        End If
        Value = rec.Fields("Value")
    End If

End Sub

Sub WriteRecord(t As Double, v As Double, IntIndex As Integer, SerIndex As Integer,
table As Recordset)
'save the results to a table

    table.AddNew
    table.Fields("Replicate") = rep
    table.Fields("Scenario") = scenario
    table.Fields("IntrusionTimeIndex") = IntrusionTimeIndex
    table.Fields("IntrusionTime") = IntrusionTime
    table.Fields("Vector") = Vector
    table.Fields("ReleaseName") = ReleaseName
    table.Fields("SeriesTimeIndex") = SerIndex
    table.Fields("SeriesTime") = t
    table.Fields("Value") = v
    table.Fields("Cavity") = Cavity
    table.Update
End Sub

Option Compare Database

Sub ShowSQL()
Dim db As Database
Dim q As QueryDefs
Dim qd As QueryDef
Dim rec As Recordset

Set db = CurrentDb
Set rec = db.OpenRecordset("tmpQueryDefs")
Set q = db.QueryDefs
For Each qd In q
    Debug.Print qd.SQL, qd.Name
    rec.AddNew
    rec.Fields("Name") = qd.Name
    rec.Fields("SQL") = qd.SQL
    rec.Update
Next
End Sub
```

Form code

```
Option Compare Database


Private Sub LdCDB_Click()
'On Error GoTo Err_LdCDB_Click
    Dim stDocName As String
'   Upload the input files
    DoCmd.SetWarnings False
    DoCmd.OpenQuery "Delete_CDB_Data"
    DoCmd.OpenQuery "Delete_RawDataNuts"
    DoCmd.SetWarnings True
    LoadCDB_Data Me
' Interpolate the NUTs data and append it to CDB_Data
    InterpolateNuts
' Interpolate the Panel 125 and 175 vlues and append those to CDB_Data
    DoCmd.SetWarnings False
    DoCmd.OpenQuery "MakeTmpPanelInterp"
    DoCmd.OpenQuery "AppendPanelInterp175"
    DoCmd.OpenQuery "AppendInterpPanelToCDB_Data"
    DoCmd.SetWarnings True

Exit_LdCDB_Click:
    Exit Sub

Err_LdCDB_Click:
    MsgBox Err.Description
    Resume Exit_LdCDB_Click

End Sub

Private Sub ListQ_Click()
    DoCmd.OpenQuery "Delete_tmpQueryDefs"
    ShowSQL
    DoCmd.OpenReport "QueryDefinitions", acViewPreview
End Sub

Private Sub LoadFilenames_Click()
LoadCDB_FileNames Me
End Sub

Private Sub LoadReltb_Click()
'On Error GoTo Err_LoadReltb_Click


    DoCmd.OpenQuery "Delete_DBData"
    LoadReltab_Data Me

Exit_LoadReltb_Click:
    Exit Sub

Err_LoadReltb_Click:
    MsgBox Err.Description
    Resume Exit_LoadReltb_Click

End Sub
Private Sub Command5_Click()
'On Error GoTo Err_Command5_Click

    'DoCmd.OpenQuery "DeleteNutsData"
    InterpolateNuts

Exit_Command5_Click:
    Exit Sub

Err_Command5_Click:
    MsgBox Err.Description
    Resume Exit_Command5_Click

End Sub
Private Sub DoRpts_Click()
On Error GoTo Err_DoRpts_Click
```

```
    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "DisplayRpts"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_DoRpts_Click:
    Exit Sub

Err_DoRpts_Click:
    MsgBox Err.Description
    Resume Exit_DoRpts_Click

End Sub

Option Compare Database

Private Sub ColloidRpt_Click()
On Error GoTo Err_ColloidRpt_Click

    Dim stDocName As String

    stDocName = "PercentRelDiff_Colloids"
    DoCmd.OpenReport stDocName, acPreview

Exit_ColloidRpt_Click:
    Exit Sub

Err_ColloidRpt_Click:
    MsgBox Err.Description
    Resume Exit_ColloidRpt_Click

End Sub
Private Sub CulebraRpt_Click()
On Error GoTo Err_CulebraRpt_Click

    Dim stDocName As String

    stDocName = "PercentRelDiff_Culebra"
    DoCmd.OpenReport stDocName, acPreview

Exit_CulebraRpt_Click:
    Exit Sub

Err_CulebraRpt_Click:
    MsgBox Err.Description
    Resume Exit_CulebraRpt_Click

End Sub
Private Sub DBRRpt_Click()
On Error GoTo Err_DBRRpt_Click

    Dim stDocName As String

    stDocName = "PercentRelDiff_DBR"
    DoCmd.OpenReport stDocName, acPreview

Exit_DBRRpt_Click:
    Exit Sub

Err_DBRRpt_Click:
    MsgBox Err.Description
    Resume Exit_DBRRpt_Click

End Sub
Private Sub IntrusionRpt_Click()
On Error GoTo Err_IntrusionRpt_Click

    Dim stDocName As String

    stDocName = "PercentRelDiff_Intrusion"
    DoCmd.OpenReport stDocName, acPreview

Exit_IntrusionRpt_Click:
    Exit Sub

Err_IntrusionRpt_Click:
    MsgBox Err.Description
    Resume Exit_IntrusionRpt_Click
```

```
End Sub
Private Sub NutsRpt_Click()
On Error GoTo Err_NutsRpt_Click

    Dim stDocName As String

    stDocName = "PercentRelDiff_Nuts"
    DoCmd.OpenReport stDocName, acPreview

Exit_NutsRpt_Click:
    Exit Sub

Err_NutsRpt_Click:
    MsgBox Err.Description
    Resume Exit_NutsRpt_Click

End Sub
Private Sub PanelComRpt_Click()
On Error GoTo Err_PanelComRpt_Click

    Dim stDocName As String

    stDocName = "PercentRelDiff_Panel"
    DoCmd.OpenReport stDocName, acPreview

Exit_PanelComRpt_Click:
    Exit Sub

Err_PanelComRpt_Click:
    MsgBox Err.Description
    Resume Exit_PanelComRpt_Click

End Sub
Private Sub SummRpt_Click()
On Error GoTo Err_SummRpt_Click

    Dim stDocName As String

    stDocName = "SummaryOfDifferences"
    DoCmd.OpenReport stDocName, acPreview

Exit_SummRpt_Click:
    Exit Sub

Err_SummRpt_Click:
    MsgBox Err.Description
    Resume Exit_SummRpt_Click

End Sub
```

Queries Used in the Database

The queries used in the database are listed with their SQL in Table 11.

Table 11. Queries used in CheckReltab

Name	SQL

AppendInterpPanelToCDB_Data	INSERT INTO CDB_Data (ReleaseName, Replicate, Scenario, IntrusionTimeIndex, IntrusionTime, Cavity, Vector, SeriesTime, [Value])
	SELECT tmpPanelInterpData.ReleaseName, tmpPanelInterpData.Replicate, tmpPanelInterpData.Scenario, tmpPanelInterpData.IntrusionTimeIndex, tmpPanelInterpData.IntrusionTime, tmpPanelInterpData.Cavity, tmpPanelInterpData.Vector, tmpPanelInterpData.SeriesTime, tmpPanelInterpData.Value
	FROM tmpPanelInterpData;

AppendPanelInterp175	INSERT INTO tmpPanelInterpData (ReleaseName, Replicate, Scenario, IntrusionTimeIndex, IntrusionTime, Cavity, Vector, SeriesTime, [Value])
	SELECT CDB_PanelData.ReleaseName, CDB_PanelData.Replicate, CDB_PanelData.Scenario, CDB_PanelData.IntrusionTimeIndex, CDB_PanelData.IntrusionTime, CDB_PanelData.Cavity, CDB_PanelData.Vector, ([CDB_PanelData.SeriesTime]+([CDB_PanelData_1.SeriesTime]-[CDB_PanelData.SeriesTime])/8) AS SeriesTime, [CDB_PanelData.Value]+([CDB_PanelData_1.Value]-[CDB_PanelData.Value])/8 AS [Value]
	FROM CDB_PanelData INNER JOIN CDB_PanelData AS CDB_PanelData_1 ON (CDB_PanelData.Vector=CDB_PanelData_1.Vector) AND (CDB_PanelData.State=CDB_PanelData_1.State) AND (CDB_PanelData.Scenario=CDB_PanelData_1.Scenario) AND (CDB_PanelData.Replicate=CDB_PanelData_1.Replicate) AND (CDB_PanelData.ReleaseName=CDB_PanelData_1.ReleaseName)
	WHERE (((CDB_PanelData.SerTime)=150) AND ((CDB_PanelData_1.SerTime)=350))
	ORDER BY CDB_PanelData.ReleaseName, CDB_PanelData.Scenario, CDB_PanelData.State;

CDB_ColliodData	SELECT CDB_Data.ReleaseName, CDB_Data.Replicate, CDB_Data.Scenario, ScenarioMap.State, CDB_Data.IntrusionTimeIndex, CDB_Data.IntrusionTime, CDB_Data.Cavity, CDB_Data.Vector, CDB_Data.SeriesTimeIndex, CDB_Data.SeriesTime, CDB_Data.Value
	FROM (CDB_Data INNER JOIN ScenarioMap ON CDB_Data.Scenario = ScenarioMap.Scenario) INNER JOIN SelectedVectors ON CDB_Data.Vector = SelectedVectors.Vector
	WHERE (((CDB_Data.ReleaseName) Like "*FRC*"));

CDB_CulebraData	SELECT CDB_Data.ReleaseName, CDB_Data.Replicate, CDB_Data.Scenario, CDB_Data.IntrusionTimeIndex, CDB_Data.IntrusionTime, CDB_Data.Cavity, CDB_Data.Vector, CDB_Data.SeriesTimeIndex, CInt([SeriesTime]*0.00000003168877) AS SerTime, CDB_Data.Value
	FROM CDB_Data INNER JOIN SelectedVectors ON CDB_Data.Vector=SelectedVectors.Vector
	WHERE (((CDB_Data.ReleaseName) Like "MT2*"));

CDB_DBRData	SELECT CDB_Data.ReleaseName, CDB_Data.Replicate, ScenarioMap.State, CDB_Data.Cavity, CDB_Data.IntrusionTimeIndex, CDB_Data.IntrusionTime, CDB_Data.Vector, CDB_Data.SeriesTimeIndex, CDB_Data.SeriesTime, CDB_Data.Value
	FROM (ScenarioMap INNER JOIN CDB_Data ON ScenarioMap.Scenario = CDB_Data.Scenario) INNER JOIN SelectedVectors ON CDB_Data.Vector = SelectedVectors.Vector
	WHERE (((CDB_Data.ReleaseName)="BRIN_REL"))
	ORDER BY ScenarioMap.State, CDB_Data.Cavity, CDB_Data.IntrusionTime, CDB_Data.Vector, CDB_Data.SeriesTime;

CDB_IntrusionData	SELECT CDB_Data.ReleaseName, CDB_Data.Replicate, ScenarioMap.State, ScenarioMap.Scenario, CDB_Data.IntrusionTimeIndex, CDB_Data.IntrusionTime, CDB_Data.Cavity, CDB_Data.Vector, CDB_Data.SeriesTimeIndex, CInt([SeriesTime]*0.00000003168877) AS SerTime, CDB_Data.Value
	FROM (CDB_Data INNER JOIN ScenarioMap ON CDB_Data.Scenario=ScenarioMap.Scenario) INNER JOIN SelectedVectors ON CDB_Data.Vector=SelectedVectors.Vector
	WHERE (((CDB_Data.ReleaseName) Like "LDC*"));

CDB_NutsData	SELECT CDB_Data.Replicate, CDB_Data.Scenario, ScenarioMap.State, CDB_Data.IntrusionTimeIndex, CDB_Data.IntrusionTime, CDB_Data.Vector, CDB_Data.ReleaseName, CDB_Data.SeriesTimeIndex, CDB_Data.SeriesTime, IIf([SeriesTime]>20000,CInt([SeriesTime]*0.00000003168877),[SeriesTime]) AS SerTime, CDB_Data.Value
	FROM (CDB_Data INNER JOIN ScenarioMap ON CDB_Data.Scenario=ScenarioMap.Scenario) INNER JOIN SelectedVectors ON CDB_Data.Vector=SelectedVectors.Vector
	WHERE (((CDB_Data.ReleaseName) Like "A00*" Or (CDB_Data.ReleaseName)="EPALWMBT"))
	ORDER BY CDB_Data.Replicate, CDB_Data.Scenario, ScenarioMap.State, CDB_Data.IntrusionTimeIndex, CDB_Data.IntrusionTime, CDB_Data.Vector, CDB_Data.ReleaseName, CDB_Data.SeriesTimeIndex;

CDB_NutsRawData	SELECT CDB_RawData_Nuts.Replicate, CDB_RawData_Nuts.Scenario, ScenarioMap.State, CDB_RawData_Nuts.IntrusionTimeIndex, CDB_RawData_Nuts.IntrusionTime, CDB_RawData_Nuts.Vector, CDB_RawData_Nuts.ReleaseName, CDB_RawData_Nuts.SeriesTimeIndex, CDB_RawData_Nuts.SeriesTime, IIf([SeriesTime]>20000,CInt([SeriesTime]*0.00000003168877),[SeriesTime]) AS SerTime, CDB_RawData_Nuts.Value
	FROM ScenarioMap INNER JOIN CDB_RawData_Nuts ON ScenarioMap.Scenario = CDB_RawData_Nuts.Scenario
	WHERE (((CDB_RawData_Nuts.IntrusionTime)=3000) AND ((CDB_RawData_Nuts.Vector)="v030") AND ((CDB_RawData_Nuts.ReleaseName)="A00TH230"))
	ORDER BY CDB_RawData_Nuts.Replicate, CDB_RawData_Nuts.Scenario, ScenarioMap.State, CDB_RawData_Nuts.IntrusionTimeIndex, CDB_RawData_Nuts.IntrusionTime, CDB_RawData_Nuts.Vector, CDB_RawData_Nuts.ReleaseName, CDB_RawData_Nuts.SeriesTime;

CDB_PanelData	SELECT CDB_Data.ReleaseName, CDB_Data.Replicate, CDB_Data.Scenario, ScenarioMap.State, CDB_Data.IntrusionTimeIndex, CDB_Data.IntrusionTime, CDB_Data.Cavity, CDB_Data.Vector, CDB_Data.SeriesTimeIndex, CDB_Data.SeriesTime, CInt([SeriesTime]*0.00000003168877) AS SerTime, CDB_Data.Value
	FROM (CDB_Data INNER JOIN ScenarioMap ON CDB_Data.Scenario=ScenarioMap.Scenario) INNER JOIN SelectedVectors ON CDB_Data.Vector=SelectedVectors.Vector
	WHERE (((CDB_Data.ReleaseName) Like "CLE*" Or (CDB_Data.ReleaseName)="CNETOTAL"));

Delete_CDB_Data	DELETE CDB_Data.ReleaseName FROM CDB_Data;
Delete_CDB_FileNames	DELETE CDB_FileNames.FileName FROM CDB_FileNames;
Delete_DBData	DELETE RELTAB_Data.ReleaseName FROM RELTAB_Data;
Delete_RawDataNuts	DELETE CDB_RawData_Nuts.ReleaseName FROM CDB_RawData_Nuts;
Delete_tmpQueryDefs	DELETE tmpQueryDefs.Name FROM tmpQueryDefs;

Find duplicates for CDB_Data_Intrusion	SELECT CDB_Data_Intrusion.ReleaseName, CDB_Data_Intrusion.Replicate, CDB_Data_Intrusion.Scenario, CDB_Data_Intrusion.IntrusionTimeIndex, CDB_Data_Intrusion.Vector, CDB_Data_Intrusion.SeriesTime, CDB_Data_Intrusion.IntrsuisonTime, CDB_Data_Intrusion.Cavity, CDB_Data_Intrusion.SeriesTimeIndex, CDB_Data_Intrusion.Value
	FROM CDB_Data_Intrusion
	WHERE (((CDB_Data_Intrusion.ReleaseName) In (SELECT [ReleaseName] FROM [CDB_Data_Intrusion] As Tmp GROUP BY [ReleaseName],[Replicate],[Scenario],[IntrusionTimeIndex],[Vector],[SeriesTime] HAVING Count(*)>1 And [Replicate] = [CDB_Data_Intrusion].[Replicate] And [Scenario] = [CDB_Data_Intrusion].[Scenario] And [IntrusionTimeIndex] = [CDB_Data_Intrusion].[IntrusionTimeIndex] And [Vector] = [CDB_Data_Intrusion].[Vector] And [SeriesTime] = [CDB_Data_Intrusion].[SeriesTime])))
	ORDER BY CDB_Data_Intrusion.ReleaseName, CDB_Data_Intrusion.Replicate, CDB_Data_Intrusion.Scenario, CDB_Data_Intrusion.IntrusionTimeIndex,

	CDB_Data_Intrusion.Vector, CDB_Data_Intrusion.SeriesTime;

Find duplicates for DB_Data	SELECT First(DB_Data.Replicate) AS [Replicate Field], First(DB_Data.Scenario) AS [Scenario Field], First(DB_Data.Cavity) AS [Cavity Field], First(DB_Data.Vector) AS [Vector Field], First(DB_Data.IntrusionTimeIndex) AS [IntrusionTimeIndex Field], First(DB_Data.SeriesTimeIndex) AS [SeriesTimeIndex Field], First(DB_Data.ReleaseName) AS [ReleaseName Field], Count(DB_Data.Replicate) AS NumberOfDups
	FROM DB_Data
	GROUP BY DB_Data.Replicate, DB_Data.Scenario, DB_Data.Cavity, DB_Data.Vector, DB_Data.IntrusionTimeIndex, DB_Data.SeriesTimeIndex, DB_Data.ReleaseName
	HAVING (((Count(DB_Data.Replicate))>1) AND ((Count(DB_Data.ReleaseName))>1));

MakeTmpPanelInterp	SELECT CDB_PanelData.ReleaseName, CDB_PanelData.Replicate, CDB_PanelData.Scenario, CDB_PanelData.IntrusionTimeIndex, CDB_PanelData.IntrusionTime, CDB_PanelData.Cavity, CDB_PanelData.Vector, ([CDB_PanelData.SeriesTime]+[CDB_PanelData_1.SeriesTime])/2 AS SeriesTime, ([CDB_PanelData.Value]+[CDB_PanelData_1.Value])/2 AS [Value] INTO tmpPanelInterpData
	FROM CDB_PanelData INNER JOIN CDB_PanelData AS CDB_PanelData_1 ON (CDB_PanelData.ReleaseName = CDB_PanelData_1.ReleaseName) AND (CDB_PanelData.Replicate = CDB_PanelData_1.Replicate) AND (CDB_PanelData.Scenario = CDB_PanelData_1.Scenario) AND (CDB_PanelData.State = CDB_PanelData_1.State) AND (CDB_PanelData.Vector = CDB_PanelData_1.Vector)
	WHERE (((CDB_PanelData.SerTime)=100) AND ((CDB_PanelData_1.SerTime)=150))
	ORDER BY CDB_PanelData.ReleaseName, CDB_PanelData.Scenario, CDB_PanelData.State, CDB_PanelData.Vector;

RelPercentDiff_Colloids	SELECT RelTab_ColloidData.Replicate, RelTab_ColloidData.Scenario, RelTab_ColloidData.Vector, RelTab_ColloidData.ReleaseName, RelTab_ColloidData.Value, CDB_ColliodData.Value, IIf([RelTab_ColloidData.Value]+[CDB_ColliodData.Value]=0,0,Abs([RelTab_ColloidData.Value]-[CDB_ColliodData.Value])/(([RelTab_ColloidData.Value]+[CDB_ColliodData.Value])/2)*100) AS PercentDiff
	FROM CDB_ColliodData INNER JOIN RelTab_ColloidData ON (CDB_ColliodData.Replicate = RelTab_ColloidData.Replicate) AND (CDB_ColliodData.State = RelTab_ColloidData.Scenario) AND (CDB_ColliodData.ReleaseName = RelTab_ColloidData.ReleaseName) AND (CDB_ColliodData.Vector = RelTab_ColloidData.Vector)
	ORDER BY
	IIf([RelTab_ColloidData.Value]+[CDB_ColliodData.Value]=0,0,Abs([RelTab_ColloidData.Value]-[CDB_ColliodData.Value])/(([RelTab_ColloidData.Value]+[CDB_ColliodData.Value])/2)*100) DESC;

RelPercentDiff_Culebra	SELECT CDB_CulebraData.ReleaseName, CDB_CulebraData.Replicate, CDB_CulebraData.Scenario, CDB_CulebraData.IntrusionTimeIndex, CDB_CulebraData.IntrusionTime, CDB_CulebraData.Cavity, CDB_CulebraData.Vector, CDB_CulebraData.SeriesTimeIndex, CDB_CulebraData.SerTime, CDB_CulebraData.Value, RelTab_CulebraReleases.Value, IIf([Reltab_CulebraReleases.Value]=0,0,Abs([Reltab_CulebraReleases.Value]-[CDB_CulebraData.value])/(([RelTab_CulebraReleases.Value]+[CDB_CulebraData.value])/2)*100) AS PercentRelDiff
	FROM RelTab_CulebraReleases INNER JOIN CDB_CulebraData ON (RelTab_CulebraReleases.Vector = CDB_CulebraData.Vector) AND (RelTab_CulebraReleases.SeriesTimeIndex = CDB_CulebraData.SeriesTimeIndex) AND (RelTab_CulebraReleases.ReleaseName = CDB_CulebraData.ReleaseName) AND (RelTab_CulebraReleases.Replicate = CDB_CulebraData.Replicate) AND (RelTab_CulebraReleases.Scenario = CDB_CulebraData.Scenario)
	ORDER BY
	IIf([Reltab_CulebraReleases.Value]=0,0,Abs([Reltab_CulebraReleases.Value]-[CDB_CulebraData.value])/(([RelTab_CulebraReleases.Value]+[CDB_CulebraData.value])/2)*100) DESC;

RelPercentDiff_DBR	SELECT CDB_DBRData.ReleaseName, CDB_DBRData.Replicate, CDB_DBRData.State, CDB_DBRData.IntrusionTimeIndex, CDB_DBRData.IntrusionTime, RelTab_DBRData.IntrusionTime, CDB_DBRData.Cavity, CDB_DBRData.Vector, CDB_DBRData.SeriesTimeIndex, CDB_DBRData.SeriesTime, RelTab_DBRData.SeriesTime, CDB_DBRData.Value, RelTab_DBRData.Value, IIf([CDB_DBRData.Value]+[RelTab_DBRData.Value]=0,0, Abs([CDB_DBRData.Value]-[RelTab_DBRData.Value])/(([CDB_DBRData.Value]+[RelTab_DBRData.Value])/2)*100) AS PercentDiff
	FROM CDB_DBRData INNER JOIN RelTab_DBRData ON (CDB_DBRData.Vector = RelTab_DBRData.Vector) AND (CDB_DBRData.State = RelTab_DBRData.Scenario) AND (CDB_DBRData.Replicate = RelTab_DBRData.Replicate) AND (CDB_DBRData.Cavity = RelTab_DBRData.Cavity) AND (CDB_DBRData.SeriesTime = RelTab_DBRData.SeriesTime) AND (CDB_DBRData.IntrusionTimeIndex = RelTab_DBRData.IntrusionTimeIndex)
	ORDER BY IIf([CDB_DBRData.Value]+[RelTab_DBRData.Value]=0,0, Abs([CDB_DBRData.Value]-[RelTab_DBRData.Value])/(([CDB_DBRData.Value]+[RelTab_DBRData.Value])/2)*100) DESC;

RelPercentDiff_Intrusion	SELECT CDB_IntrusionData.Replicate, CDB_IntrusionData.State, CDB_IntrusionData.IntrusionTimeIndex, CDB_IntrusionData.IntrusionTime AS CDB_IntrusionTime, Reltab_IntrusionData.IntrusionTime AS Reltab_IntrusionTime, CDB_IntrusionData.Cavity, CDB_IntrusionData.Vector, CDB_IntrusionData.SeriesTimeIndex, CDB_IntrusionData.SerTime, Reltab_IntrusionData.SeriesTime, CDB_IntrusionData.Value AS CDB_Value, Reltab_IntrusionData.Valu AS ReltabValue, IIf([CDB_IntrusionData.Value]+[Reltab_IntrusionData.Valu]>0,Abs([CDB_IntrusionData.Value]-[Reltab_IntrusionData.Valu])/(([CDB_IntrusionData.Value]+[Reltab_IntrusionData.Valu])/2)*100,0) AS PercDiff, CDB_IntrusionData.ReleaseName
	FROM CDB_IntrusionData INNER JOIN Reltab_IntrusionData ON (CDB_IntrusionData.IntrusionTimeIndex = Reltab_IntrusionData.IntrusionTimeIndex) AND (CDB_IntrusionData.Replicate = Reltab_IntrusionData.Replicate) AND (CDB_IntrusionData.State = Reltab_IntrusionData.Scenario) AND (CDB_IntrusionData.Vector = Reltab_IntrusionData.Vector) AND

	(CDB_IntrusionData.SerTime = Reltab_IntrusionData.SeriesTime) AND (CDB_IntrusionData.ReleaseName = Reltab_IntrusionData.ReleaseName)
	ORDER BY
	IIf([CDB_IntrusionData.Value]+[Reltab_IntrusionData.Valu]>0,Abs([CDB_IntrusionData.Value]-[Reltab_IntrusionData.Valu])/(([CDB_IntrusionData.Value]+[Reltab_IntrusionData.Valu])/2)*100,0) DESC;

RelPercentDiff_Nuts	SELECT CDB_NutsData.ReleaseName, CDB_NutsData.Replicate, CDB_NutsData.State, CDB_NutsData.Scenario, CDB_NutsData.Vector, CDB_NutsData.IntrusionTimeIndex, CDB_NutsData.IntrusionTime, Reltab_NutsData.IntrusionTime, CDB_NutsData.SeriesTimeIndex, Reltab_NutsData.SeriesTimeIndex, CDB_NutsData.SerTime, CDB_NutsData.SeriesTime AS CDB_Time, Reltab_NutsData.SeriesTime AS RELTAB_Time, CDB_NutsData.Value AS CDB_Value, Reltab_NutsData.Val AS Reltab_Value, IIf([CDB_NutsData.Value]+[Reltab_NutsData.Val]>0,Abs([CDB_NutsData.Value]-[Reltab_NutsData.Val])/(([CDB_NutsData.Value]+[Reltab_NutsData.Val])/2)*100,0) AS PercentDiff
	FROM CDB_NutsData INNER JOIN Reltab_NutsData ON (CDB_NutsData.IntrusionTime = Reltab_NutsData.IntrusionTime) AND (CDB_NutsData.Vector = Reltab_NutsData.Vector) AND (CDB_NutsData.SeriesTimeIndex = Reltab_NutsData.SeriesTimeIndex) AND (CDB_NutsData.State = Reltab_NutsData.Scenario) AND (CDB_NutsData.Replicate = Reltab_NutsData.Replicate) AND (CDB_NutsData.ReleaseName = Reltab_NutsData.ReleaseName)
	ORDER BY
	IIf([CDB_NutsData.Value]+[Reltab_NutsData.Val]>0,Abs([CDB_NutsData.Value]-[Reltab_NutsData.Val])/(([CDB_NutsData.Value]+[Reltab_NutsData.Val])/2)*100,0) DESC;

RelPercentDiff_Panel	SELECT CDB_PanelData.ReleaseName, CDB_PanelData.Replicate, CDB_PanelData.Scenario, CDB_PanelData.State, CDB_PanelData.Vector, CDB_PanelData.SeriesTimeIndex, CDB_PanelData.SerTime, Reltab_Panel.SeriesTime, CDB_PanelData.Value, Reltab_Panel.Value, IIf([CDB_PanelData.Value]+[Reltab_Panel.Value]>0,Abs([CDB_PanelData.Value]-[Reltab_Panel.Value])/(([CDB_PanelData.Value]+[Reltab_Panel.Value])/2)*100,0) AS PercentDiff FROM Reltab_Panel INNER JOIN CDB_PanelData ON (Reltab_Panel.Replicate=CDB_PanelData.Replicate) AND (Reltab_Panel.Scenario=CDB_PanelData.State) AND (Reltab_Panel.Vector=CDB_PanelData.Vector) AND (Reltab_Panel.ReleaseName=CDB_PanelData.ReleaseName) AND (Reltab_Panel.SeriesTime=CDB_PanelData.SerTime) ORDER BY IIf([CDB_PanelData.Value]+[Reltab_Panel.Value]>0,Abs([CDB_PanelData.Value]-[Reltab_Panel.Value])/(([CDB_PanelData.Value]+[Reltab_Panel.Value])/2)*100,0) DESC;
RelTab_ColloidData	SELECT RELTAB_Data.Replicate, RELTAB_Data.Scenario, RELTAB_Data.Vector, RELTAB_Data.ReleaseName, RELTAB_Data.Value FROM RELTAB_Data INNER JOIN SelectedVectors ON RELTAB_Data.Vector = SelectedVectors.Vector WHERE (((RELTAB_Data.ReleaseName) Like "*FRC*"));
RelTab_CulebraReleases	SELECT RELTAB_Data.Vector, RELTAB_Data.Replicate, RELTAB_Data.Scenario, RELTAB_Data.Cavity, RELTAB_Data.IntrusionTimeIndex, RELTAB_Data.IntrusionTime, RELTAB_Data.SeriesTimeIndex, RELTAB_Data.SeriesTime, RELTAB_Data.ReleaseName, RELTAB_Data.Value FROM RELTAB_Data INNER JOIN SelectedVectors ON RELTAB_Data.Vector = SelectedVectors.Vector WHERE (((RELTAB_Data.Scenario)="MF" Or (RELTAB_Data.Scenario)="MP"));

RelTab_DBRData	SELECT RELTAB_Data.Replicate, RELTAB_Data.Scenario, RELTAB_Data.Cavity, RELTAB_Data.Vector, RELTAB_Data.IntrusionTimeIndex, RELTAB_Data.IntrusionTime, RELTAB_Data.SeriesTimeIndex, RELTAB_Data.SeriesTime, RELTAB_Data.ReleaseName, RELTAB_Data.Value FROM RELTAB_Data INNER JOIN SelectedVectors ON RELTAB_Data.Vector = SelectedVectors.Vector WHERE (((RELTAB_Data.ReleaseName)="BRIN_REL")) ORDER BY RELTAB_Data.Scenario, RELTAB_Data.Cavity, RELTAB_Data.Vector, RELTAB_Data.IntrusionTime, RELTAB_Data.SeriesTime;
Reltab_IntrusionData	SELECT RELTAB_Data.Replicate, RELTAB_Data.Scenario, RELTAB_Data.Cavity, RELTAB_Data.Vector, RELTAB_Data.IntrusionTimeIndex, RELTAB_Data.IntrusionTime, RELTAB_Data.SeriesTimeIndex, RELTAB_Data.SeriesTime, RELTAB_Data.ReleaseName, [Value]*[SpecificActivity] AS Valu FROM (RELTAB_Data INNER JOIN SpecificActivity ON RELTAB_Data.ReleaseName=SpecificActivity.Nuclide) INNER JOIN SelectedVectors ON RELTAB_Data.Vector=SelectedVectors.Vector WHERE (((RELTAB_Data.ReleaseName) Like "LDC*"));
Reltab_NutsData	SELECT RELTAB_Data.ReleaseName, RELTAB_Data.Replicate, RELTAB_Data.Scenario, RELTAB_Data.Cavity, RELTAB_Data.Vector, RELTAB_Data.IntrusionTimeIndex, RELTAB_Data.IntrusionTime, RELTAB_Data.SeriesTimeIndex, RELTAB_Data.SeriesTime, [Value]*[SpecificActivity] AS Val FROM (RELTAB_Data INNER JOIN SpecificActivity ON RELTAB_Data.ReleaseName=SpecificActivity.Nuclide) INNER JOIN SelectedVectors ON RELTAB_Data.Vector=SelectedVectors.Vector WHERE (((RELTAB_Data.ReleaseName) Like "A00*" Or (RELTAB_Data.ReleaseName)="EPALWMBT"));

Reltab_Panel	SELECT RELTAB_Data.Replicate, RELTAB_Data.Scenario, RELTAB_Data.Cavity, RELTAB_Data.Vector, RELTAB_Data.IntrusionTimeIndex, RELTAB_Data.IntrusionTime, RELTAB_Data.SeriesTimeIndex, RELTAB_Data.SeriesTime, RELTAB_Data.ReleaseName, RELTAB_Data.Value
	FROM RELTAB_Data INNER JOIN SelectedVectors ON RELTAB_Data.Vector=SelectedVectors.Vector
	WHERE (((RELTAB_Data.ReleaseName) Like "CLE*" Or (RELTAB_Data.ReleaseName)="CNETOTAL"));

ReltabNames	SELECT DB_Data.ReleaseName
	FROM DB_Data
	GROUP BY DB_Data.ReleaseName;

SpecificActivity	SELECT NuclideProperties.Nuclide, IIf([halflife]=0,1,([Constants.Value]/([Constants_1.Value]*[Constants_2.Value])*Log(2))/([Halflife]/[Constants_1.Value]*[Mass])) AS SpecificActivity
	FROM Constants, NuclideProperties, Constants AS Constants_1, Constants AS Constants_2
	WHERE (((Constants.Constant)="Avagadro") AND ((Constants_1.Constant)="SecPerYr") AND ((Constants_2.Constant)="BqPerCi"));

Summary_all	SELECT "PANEL_ST" AS Source, "Mobilized colloid fraction" as Release,Max(RelPercentDiff_Colloids.PercentDiff) AS [Maximum percent difference], Avg(RelPercentDiff_Colloids.PercentDiff) AS [Mean percent difference], Count(RelPercentDiff_Colloids.PercentDiff) AS [Number]
	FROM RelPercentDiff_Colloids
	GROUP BY "Colloids"
	Union
	SELECT "SECOTP_2D" AS Source, "Cumulative discharge" as Release,Max(RelPercentDiff_Culebra.PercentRelDiff) AS [Maximum percent difference], Avg(RelPercentDiff_Culebra.PercentRelDiff) AS [Mean percent difference], Count(RelPercentDiff_Culebra.PercentRelDiff) AS [Number]
	FROM RelPercentDiff_Culebra
	GROUP BY "Culebra"
	Union
	SELECT "SECOTP_2D*" AS Source, "Cumulative discharge" as Release, Max(RelPercentDiff_Culebra.PercentRelDiff) AS [Maximum percent difference], Avg(RelPercentDiff_Culebra.PercentRelDiff) AS [Mean percent difference], Count(RelPercentDiff_Culebra.PercentRelDiff) AS [Number]

```
FROM RelPercentDiff_Culebra
WHERE (((RelPercentDiff_Culebra.PercentRelDiff)<1))
GROUP BY "Culebra"
Union
SELECT "BRAGFLO_DBR" AS Source, "Direct release
brine volume" AS Release,
Max(RelPercentDiff_DBR.PercentDiff) AS [Maximum
percent difference], Avg(RelPercentDiff_DBR.PercentDiff)
AS [Mean percent difference],
Count(RelPercentDiff_DBR.PercentDiff) AS [Number]
FROM RelPercentDiff_DBR
GROUP BY "DBR"
Union
SELECT "PANEL_INT" AS Source, "Cumulative
discharge through borehole" as
Release,Max(RelPercentDiff_Intrusion.PercDiff) AS
[Maximum percent difference],
Avg(RelPercentDiff_Intrusion.PercDiff) AS [Mean percent
difference], Count(RelPercentDiff_Intrusion.PercDiff) AS
[Number]
FROM RelPercentDiff_Intrusion
GROUP BY "Intrusion"
Union
SELECT "NUTS" AS Source,"Cumulative discharge to
Culebra" as Release,
Max(RelPercentDiff_Nuts.PercentDiff) AS [Maximum
percent difference], Avg(RelPercentDiff_Nuts.PercentDiff)
AS [Mean percent difference],
Count(RelPercentDiff_Nuts.PercentDiff) AS [Number]
FROM RelPercentDiff_Nuts
GROUP BY "NUTS"
UNION
SELECT "NUTS*" AS Source,"Cumulative discharge to
Culebra" as Release,
Max(RelPercentDiff_Nuts.PercentDiff) AS [Maximum
percent difference], Avg(RelPercentDiff_Nuts.PercentDiff)
AS [Mean percent difference],
Count(RelPercentDiff_Nuts.PercentDiff) AS [Number]
FROM RelPercentDiff_Nuts
WHERE (((RelPercentDiff_Nuts.PercentDiff)<1))
GROUP BY "NUTS"
union
SELECT "PANEL_CON" AS Source, "Radionuclide
concentrations in brine" as
Release,Max(RelPercentDiff_Panel.PercentDiff) AS
[Maximum percent difference],
Avg(RelPercentDiff_Panel.PercentDiff) AS [Mean percent
difference], Count(RelPercentDiff_Panel.PercentDiff) AS
[Number]
FROM RelPercentDiff_Panel
GROUP BY "Panel"
UNION SELECT "PANEL_CON*" AS Source,
"Radionuclide concentrations in brine" as
Release,Max(RelPercentDiff_Panel.PercentDiff) AS
[Maximum percent difference],
Avg(RelPercentDiff_Panel.PercentDiff) AS [Mean percent
```

	difference], Count(RelPercentDiff_Panel.PercentDiff) AS [Number]
	FROM RelPercentDiff_Panel
	WHERE (((RelPercentDiff_Panel.PercentDiff)<.1))
	GROUP BY "Panel";

Summary_All_Formated	SELECT Summary_all.Source, Summary_all.Release, Summary_all.[Maximum percent difference], Summary_all.[Mean percent difference], Summary_all.Number
	FROM Summary_all;

Summary_Colloids	SELECT "Colloids" AS Source, Max(RelPercentDiff_Colloids.PercentDiff) AS [Maximum percent difference], Avg(RelPercentDiff_Colloids.PercentDiff) AS [Mean percent difference], Count(RelPercentDiff_Colloids.PercentDiff) AS [Number]
	FROM RelPercentDiff_Colloids
	GROUP BY "Colloids";

Summary_Colloids_non_zero	SELECT "Colloids" AS Source, Max(RelPercentDiff_Colloids.PercentDiff) AS [Maximum percent difference], Avg(RelPercentDiff_Colloids.PercentDiff) AS [Mean percent difference], Count(RelPercentDiff_Colloids.PercentDiff) AS [Number]
	FROM RelPercentDiff_Colloids
	WHERE ((([CDB_ColliodData.Value]+[RelTab_ColloidData.Value])>0))
	GROUP BY "Colloids";

Summary_Culebra	SELECT "Culebra" AS Source, Max(RelPercentDiff_Culebra.PercentRelDiff) AS [Maximum percent difference], Avg(RelPercentDiff_Culebra.PercentRelDiff) AS [Mean percent difference], Count(RelPercentDiff_Culebra.PercentRelDiff) AS [Number]
	FROM RelPercentDiff_Culebra
	GROUP BY "Culebra";

Summary_Culebra_excluding_large	SELECT "Culebra" AS Source, Max(RelPercentDiff_Culebra.PercentRelDiff) AS [Maximum percent difference], Avg(RelPercentDiff_Culebra.PercentRelDiff) AS [Mean percent difference], Count(RelPercentDiff_Culebra.PercentRelDiff) AS [Number]
	FROM RelPercentDiff_Culebra
	WHERE (((RelPercentDiff_Culebra.PercentRelDiff)<1))
	GROUP BY "Culebra";

Summary_Culebra_non_zero	SELECT "Culebra" AS Source, Max(RelPercentDiff_Culebra.PercentRelDiff) AS [Maximum percent difference], Avg(RelPercentDiff_Culebra.PercentRelDiff) AS [Mean percent difference], Count(RelPercentDiff_Culebra.PercentRelDiff) AS [Number]
	FROM RelPercentDiff_Culebra
	WHERE ((([RelTab_CulebraReleases.Value]+[CDB_CulebraData.Value])>0))
	GROUP BY "Culebra";

Summary_DBR	SELECT "DBR" AS Source, Max(RelPercentDiff_DBR.PercentDiff) AS [Maximum percent difference], Avg(RelPercentDiff_DBR.PercentDiff) AS [Mean percent difference], Count(RelPercentDiff_DBR.PercentDiff) AS [Number]
	FROM RelPercentDiff_Culebra, RelPercentDiff_DBR
	GROUP BY "DBR";

Summary_Intrusion	SELECT "Intrusion" AS Source, Max(RelPercentDiff_Intrusion.PercDiff) AS [Maximum percent difference], Avg(RelPercentDiff_Intrusion.PercDiff) AS [Mean percent difference], Count(RelPercentDiff_Intrusion.PercDiff) AS [Number]
	FROM RelPercentDiff_Intrusion
	GROUP BY "Intrusion";

Summary_Nuts	SELECT "Nuts" AS Source, Max(RelPercentDiff_Nuts.PercentDiff) AS [Maximum percent difference], Avg(RelPercentDiff_Nuts.PercentDiff) AS [Mean percent difference], Count(RelPercentDiff_Nuts.PercentDiff) AS [Number]
	FROM RelPercentDiff_Nuts
	GROUP BY "Nuts";

Summary_Nuts_non_zero	SELECT Max(RelPercentDiff_Nuts.PercentDiff) AS [Maximum percent difference], Avg(RelPercentDiff_Nuts.PercentDiff) AS [Mean percent difference], Count(RelPercentDiff_Nuts.PercentDiff) AS [Number non-zero]
	FROM RelPercentDiff_Nuts
	WHERE (((RelPercentDiff_Nuts.PercentDiff)>0));

Summary_Panel	SELECT "Panel" AS Source, Max(RelPercentDiff_Panel.PercentDiff) AS [Maximum percent difference], Avg(RelPercentDiff_Panel.PercentDiff) AS [Mean percent difference], Count(RelPercentDiff_Panel.PercentDiff) AS [Number]
	FROM RelPercentDiff_Panel
	GROUP BY "Panel";

tmpDeleteNutsData	DELETE CDB_Data.Replicate, CDB_Data.Scenario, CDB_Data.IntrusionTimeIndex, CDB_Data.IntrusionTime, CDB_Data.Vector, CDB_Data.ReleaseName, CDB_Data.SeriesTimeIndex, CDB_Data.SeriesTime, IIf([SeriesTime]>20000,CInt([SeriesTime]*0.00000003168877),[SeriesTime]) AS SerTime, CDB_Data.Value
	FROM CDB_Data
	WHERE (((CDB_Data.ReleaseName) Like "A00*" Or (CDB_Data.ReleaseName)="EPALWMBT"));

tmpRecordsByCode	SELECT Left([FileName],7) AS Expr1, Count(CDB_FileNames.FileName) AS CountOfFileName
	FROM CDB_FileNames
	GROUP BY Left([FileName],7);

APPENDIX E CHECKRELTAB DATABASE VALIDATION

The CheckReltab database is a Microsoft® Access database. Microsoft® Access is commercial, off the shelf (COTS) software and hence exempt from the requirement of NP 19-1 (Chavez 2004). However, there are two types of calculations performed in CheckReltab that were verified. The first calculation involves the linear interpolation of the PANEL and NUTS CDB data. The PANEL interpolations were computed using two queries, MakeTmpPanelInterp and AppendPanelInterp175. These queries generate the interpolated values for times 125 and 175, respectively, and place them in the temporary table tmpPanelInterpData. These expressions computing the interpolation were inspected to verify that they were correct and checked by exporting the data to Excel and computing the interpolated values there (Table 12). The values were identical. The data in tmpPanelInterpData are subsequently appended to table CDB_Data using the query AppendInterpPanelToCDB_Data.

Table 12. Verification of PANEL interpolations

Release Name	Scenario	Vector	Computed Value	Series Time1	Series Time 2	Value 1	Value 2	Value Excel
CLEAM241	S1	V005	0.00038263	100	150	0.000382675	0.000382584	0.00038263
CLEAM241	S1	V030	0.0124616	100	150	0.0124631	0.0124601	0.0124616
CLEAM241	S1	V057	0.00467955	100	150	0.00468011	0.00467899	0.00467955
CLEAM241	S1	V062	0.01120565	100	150	0.011207	0.0112043	0.01120565
CLEAM241	S1	V096	0.0935774	100	150	0.0935885	0.0935663	0.0935774
CLEAM241	S2	V005	0.000310581	100	150	0.000310618	0.000310543	0.000310581
CLEAM241	S2	V030	0.01136515	100	150	0.0113665	0.0113638	0.01136515
CLEAM241	S2	V057	0.00450953	100	150	0.00451007	0.00450899	0.00450953
CLEAM241	S2	V062	0.010445	100	150	0.0104462	0.0104438	0.010445
CLEAM241	S2	V096	0.0755464	100	150	0.0755554	0.0755374	0.0755464

The interpolation of the NUTS values were computed within a module of the Checkreltab database. An example of the "raw" data from the CDB file was exported to Excel and values for time 300, 350, 400, 450 and 500 were computed by linear interpolation (Table 13). These results were then compared to the values computed by CheckReltab and found to be identical. Note that the interpolations are done using the time in seconds.

Table 13. Check of interpolation of NUTS data: scenario S2, vector 62, intrusion at 100 years

Release Name	CDB Time (Seconds)	CDB Time (Years)	Value	Time Required (Years)	Time required (Seconds)	Interpolated Value (Excel)	Interpolated Value (CheckReltab)
A00AM241	9466800000	300	0	300	9467076191	0.000729829	0.000729829
A00AM241	10818100000	343	3.57078	350	11044922223	4.986054358	4.986054358
A00AM241	22960500000	728	79.3342	400	12622768255	14.8311436	14.8311436
A00AM241	23667700000	750	84.2198	450	14200614287	24.67623284	24.67623284
A00AM241	41023700000	1300	145.697	500	15778460319	34.52132208	34.52132208
A00AM241	41178700000	1305	146.211				

The second type of computation in CheckReltab is the calculation of the relative percent differences. These are simple calculation done within queries. The code was inspected and an example (colloid concentrations) exported to Excel. RPD values for the CDB and RELTAB data were then computed and compared to those computed in CheckReltab. The RPD values computed in Checkreltab and Excel were identical. The RPD calculations in the queries for the other data sets are the same.

Table 14. Comparison of RPD values computed in CheckReltab to those computed in Excel

Scenario	Vector	Release Name	Reltab Value	CDB Value	RPD CheckReltab	RPD Excel
E1	v062	Pu239_FRCMIC	0.109	0.109029	0.026602	0.026602
E0	v062	Pu239_FRCINT	0.001	0.001	0.025997	0.025997
E0	v057	Am241_FRCMIN	0.01975	0.019745	0.023294	0.023294
E0	v096	Th230_FRCMIN	0.001879	0.001879	0.019161	0.019161

The Excel spreadsheets for these two verification examples (NutsInternalValidation.xls and RPDValidation.xls) are contained in the CRA1BC_ReadRel.zip file, which has been archived on the WIPP PA Alpha Cluster in CMS library LIBCRA1BC_SUMVERIFY.