

**SOFTWARE VALIDATION PLAN  
FOR  
TOTAL-SYSTEM PERFORMANCE ASSESSMENT  
(TPA) CODE VERSION 5.1**

**Prepared by**

**James Winterle  
Christopher Grossman  
Ron Janetzke**

**March 2007**

## **1. SCOPE OF VALIDATION**

### **1.1 INTRODUCTION**

This software validation plan for the Total-System Performance Assessment (TPA) Code Version 5.1 has been prepared in accordance with the requirements of Section 5.8 of the Geosciences and Engineering Division Technical Operating Procedure, TOP-18 (Geosciences and Engineering Division, 2005).

This software validation plan describes the tests to be executed on the TPA Code Version 5.1Beta. TPA version 5.1Beta is an update to the previously validated TPA version 5.0. It is the result of numerous software changes that were made in consultation with the U.S. Nuclear Regulatory Commission (NRC) and Center for Nuclear Waste Regulatory Analyses (CNWRA) technical staffs. These discussions resulted in the identification of several code changes that were needed. All software changes are documented in a Software Change Reports (SCRs), which will be archived in CNWRA quality assurance records upon completion of the validation.

It is expected that, during validation testing of TPA Version 5.1Beta, needs for additional code changes are likely to be identified and implemented (e.g., bug fixes, or refinements to the numerical implementation of process conceptual models). When NRC and CNWRA staff are in agreement that all of the code changes that are necessary for a fully validated code have been identified, the Software Requirements Description, which was previously developed with a “draft” designation (Janetzke, et al., 2005) will be updated and finalized. If needs for code changes are identified after some portions of the validation test activities have already been completed, an assessment will be made as to whether previously conducted validation tests may need to be repeated. A validation test activity would only need to be repeated if a software change directly affects that portion of the code that was tested. After validation of the code is complete, the Beta designation will be removed and the code will be delivered as TPA Version 5.1. A software release notice will be issued for TPA Version 5.1 as a validated code prior to delivery to NRC.

### **1.2 Scope**

The scope of validation is limited to those components of the TPA code affected by modifications since the last complete validation testing was conducted for TPA Version 5.0 (Janetzke and Scherer, 2003). SCRs initiated since TPA Version 5.0s have been reviewed and the testing activities described in the following sections summarize the testing needed to validate TPA Version 5.1. Previous testing conducted during software changes and documented in Software Change Reports does not need to be repeated unless subsequent changes affect the confidence in the results of a particular test.

## **2. REFERENCES**

Geosciences and Engineering Division. Technical Operating Procedure, TOP-018, Development and Control of Scientific and Engineering Software.” Revision 10, Change 0. August 5, 2005.

Janetzke, R., J. Winterle, R. Benke, O. Pensado, G. Adams, R. Rice, and C. Scherer. “Draft Software Requirements Description for the Total-System Performance Assessment Version 5.1 Code.” San Antonio, Texas: CNWRA. November 1, 2005.

Janetzke, R. and C. Scherer. Software Validation Report for the Total-System Performance Assessment Version 5.0 Code. San Antonio, Texas: CNWRA. December 5, 2003.

Ibarra, L., T. Wilt, G. Ofoegbu, R. Kazban, F. Ferrante, and A. Chowdhury. "Drip Shield-Waste Package Mechanical Interaction." San Antonio, Texas: CNWRA. June, 2006.

### **3. ENVIRONMENT**

#### **3.1 Software**

The software to be used for this validation process is the TPA version 5.1Beta code. The TPA code can be obtained locally from host machines Alby\TPA\_Share (PC version) or Spock/solapps/cnwra (UNIX version). The programs will be compiled using Lahey LF95 version 7.1 (PC version) or the SUNWsprow Fortran compiler version 5.0 (UNIX version). Note, however, that all validation tests are to be conducted using the PC version of the code, which is the preferred platform. At the time this validation plan was prepared, the latest update to the software was designated TPA Version 5.1BetaT. It is expected, however, that subsequent updates (versions 5.1BetaU, 5.1BetaV, etc.) will be made if additional SCRs are implemented during the validation process. Code testers should ensure they are using the latest version of the software when they begin their test activities. Specific testing activities would only need to be repeated for a subsequent update if the implemented code change directly affects the code functionality that was tested by that activity. All software changes initiated during the validation effort will be reviewed by a committee consisting of Ron Janetzke, James Winterle, and Christopher Grossman (NRC) to determine if the change necessitates additional or repeated validation testing. This review will be noted by the Software Developer (Ron Janetzke) on the SCR.

#### **3.2 Hardware**

All validation tests will be conducted using the PC version of the code using the Microsoft/Windows XP or Windows 2000 operating system. Before release of the UNIX Version of the code, a simple comparison of end results from a multiple realization reference case input data using the validated PC version will be made to a code version compiled on a SUN/sparc processor using the SUN OS version 5.8 operating system.

### **4. PREREQUISITES**

The execution of the TPA code requires the setting of two environment variables, TPA\_TEST and TPA\_DATA. These should point to the parent directory for the 'codes' and 'data' subdirectories that contain the executable code and input data files. Additionally, PCs used for testing require a combined total of RAM and Virtual memory of approximately 2.0 Gb.

### **5. ASSUMPTIONS AND CONSTRAINTS**

TPA Code version 5.0s is the last version of the TPA code to have been fully validated. It is also assumed that any TPA code modules or subroutines that have not been modified since version 5.0s do not need to be tested during the version 5.1 validation. The list of validation test activities in Section 6 of this plan are the result of a review of all software changes that have been implemented since version 5.0s. Most of the major modules have been modified to some extent and will, therefore, require validation testing.

## 6. TEST CASES

As indicated in the previous section, the software validation test cases described in the following subsections are intended to test essentially all of the major functional capabilities of TPA version identified in the Draft Software Requirements Description (Janetzke, et al., 2005) as well as additional requirements specified in subsequent correspondence from the NRC clients. [Note that the Draft Software Requirements Description is being revised and finalized to include these additional requirements ]. To accomplish this objective, two general categories of tests will be performed: process-level and system-level.

Process-level tests will evaluate the results of individual modules, submodules, or subroutines. Process-level tests identified in the task list in Section 6.1 will include a review of related SCRs that have been completed since TPA Version 5.0s to avoid repeating process-level testing that may already have been done. All process-level tests will be documented in separate Software Validation Test Reports (SVRs) that will be combined with documentation of the system-level testing into a final SVR for the entire TPA Version 5.1 code.

System-level tests evaluate the main results of the system taken as whole. For example, system-level testing will likely include several test cases to be developed from the TPA input files, such that the outputs from the different modules and subroutines can be checked for reasonableness of results (i.e., results are explainable) and code stability. System-level testing also can be used to provide confidence that changes to one module or auxiliary code do not introduce problems or errors in a different code module. A summary of system level test activities is provided in Section 6.2 of this plan. SVRs will be prepared to document the system-level testing and combined with documentation of the process-level testing into a final SVR for the entire TPA Version 5.1 code.

It should be noted that, because the great number of functions and complex couplings to be tested in the TPA Version 5.1, the process- and system-level tests described in the following subsections 6.1 and 6.2, it is not practical to follow the usual practice of specifying for every test case the required inputs, outputs, and bases for objective comparison of results to known solutions. Rather, this plan is intended to specify the functional requirements of the TPA Version 5.1 code to be tested for each of the major code modules and functionalities. This approach permits flexibility to permit the validation teams specified for each task to develop one or more tests cases for each validation task. The descriptions of inputs, outputs, and objective comparisons of results to known solutions will, therefore, be provided in the SVRs that will be prepared to document each task. The technical and programmatic reviews that will be conducted on each of the SVRs will be used to ensure that these descriptions are sufficiently documented. Because TOP-018 requires the validation plans to be reviewed in accordance with QAP-002, the plans documented in each SVR will receive a documented technical review using form (QAP-6) to ensure the planned test cases for each validation task are adequate.

Some process- and system-level testing may result in identification of errors that necessitate additional software changes. As previously mentioned, all such software changes initiated during the validation effort will be reviewed by a committee consisting of Ron Janetzke, James Winterle, and Christopher Grossman (NRC) to determine if the change necessitates additional or repeated validation testing. This review will be noted by the Software Developer (Ron Janetzke) on the SCR. In-progress and completed SVRs and SCRs will be made available via a shared network drive so that validation teams may benefit from the ability to review results from the other validation tests that may affect their own tests. Frequent email communications

and meetings also will be utilized to keep teams apprised of code changes made during the validation process.

## 6.1 Process-Level Testing

Table 6.1-1 lists process-level validation tasks for the major TPA code functionalities that have been significantly revised since TPA Version 5.0s. These code functionalities may be represented by one or more individual modules, submodules, or routines. Table 6.1-1 also provides a list of the SCRs since Version 5.0s. In general, testing conducted in conjunction with individual software changes that is documented in the SCRs should be focused on verification that the changes were correctly implemented. In practice, however, some of the software change testing documented in the SCRs contains more in-depth analyses of the theoretical constructs and the information that is passed between modules, which is a goal of software validation testing. Accordingly, the SCRs listed in Table 6.1-1 should be reviewed to avoid any unnecessary repetition of tests that may already have been conducted during acceptance testing of software changes. Note that a few SCRs that document relatively minor bug fixes or input data changes are not included in Table 6.1-1. In addition to the SCR reviews, validation teams should also review the requirements specified in the Software Requirements Description for TPA Version 5.1.

The last column in Table 6.1-1 identifies the technical staff members who will likely be involved in the validation test design, execution, or technical review of each task. These names could change depending on staff availability. Separate SVRs will be prepared for each of these tasks. A template cover sheet to be used for each SVR is provided in the Appendix.

The subsections following Table 6.1-1 provide descriptions of the scope for each of the process-level testing tasks. Technical staff assigned to work on each task may identify additional tests as they deem appropriate after reviewing the listed SCRs. All tests should be documented in the SVRs to include test descriptions, code inputs, outputs, and bases for objective comparison of results to known solutions. Each SVR will receive technical and programmatic reviews to ensure the requirements of Technical Operating Procedure TOP-018 are met.

Table 6.1-1. Process-Level Validation Tasks			
Task	Functionality to be Tested	Relevant Software Change Reports	Validation Team
P-1	<u>Executive functions</u> : module execution and management of input/output data ( <i>exec.f</i> , <i>reader.f</i> , <i>sampler.f</i> )	470, 477, 530, 560, 565, 570, 575, 587, 590, 594, 598, 600, 602, 603, 612, 656, 657, 667, 668, 669, 671, 672, 673, 675, 677, 679, 680	R. Janetzke *J. Winterle
P-2	<u>Routine and subroutine libraries</u> : ( <i>numrecip.f</i> , <i>nrutil.f</i> )	553, 586	R. Janetzke *J. Winterle
P-3	<u>Inventory control</u> : fuel age at $t_0$ ; <i>burnup.dat</i> data file; glass fraction	483, 554, 572	O. Povetko R. Rice *J. Winterle

Table 6.1-1. Process-Level Validation Tasks			
Task	Functionality to be Tested	Relevant Software Change Reports	Validation Team
P-4	Climate and subarea-averaged net infiltration: ( <i>uzflow.f</i> )	488, 602, 665	R. Nes S. Stothoff *H. Basagaoglu
P-5	<u>Near field chemical and temperature conditions:</u> ( <i>nfenv.f</i> )	478, 480, 488, 520, 553, 555, 562, 589, 591, 604, 605, 622, 658	J. Myers K. Das M. Juckett R. Fedors *C. Manepally
P-6	<u>Drift degradation:</u> ( <i>driftfail.f</i> )	610, 627, 664	J. Mancillas *G. Ofoegbu
P-7	<u>Mechanical failure of drip shields and waste packages:</u> ( <i>mechdrive.f, mechfail.f</i> )	480, 481, 609, 629, 660, 664, 681	J. Mancillas A.-B. Ibrahim *L. Ibarra
P-8	<u>Drip shield corrosion:</u> ( <i>dsfail.f, dsfailt.f</i> )	478, 562, 607	X. He P. Shukla *J. Mancillas
P-9	<u>Waste package corrosion:</u> ( <i>ebsfail.f, failt.f</i> )	478, 571, 577, 578, 593, 599, 607, 609, 651, 660	P. Shukla A. Jung *J. Mancillas
P-10	<u>Releases from the engineered barrier system:</u> ( <i>ebsrel.f, releaset.f</i> )	546, 549, 556, 587, 595, 597, 608, 609, 622, 628, 651, 652, 653, 658	H. Basagaoglu R. Rice R. Fedors *D. Pickett
P-11	<u>Colloid facilitated releases from the engineered barrier system:</u> ( <i>ebsrel.f, releaset.f</i> )	567, 611	D. Pickett S. Painter *F. Osidele
P-12	<u>Unsaturated zone flow and transport:</u> ( <i>uzft.f, nefmks.f</i> )	484, 489, 517, 523, 578, 585, 614	J. McMurry G. Peters R. Fedors *F. Osidele
P-13	<u>Saturated zone flow and transport:</u> ( <i>szft.f, nefmks.f</i> )	484, 489, 517, 523, 529, 561	J. Winterle P. Bertetti G. Peters *F. Osidele
P-14	<u>Intrusive volcanism:</u> ( <i>volcano.f</i> )	490, 522, 564, 574, 585	D. Basu R. Janetzke *N. Adams

Table 6.1-1. Process-Level Validation Tasks			
Task	Functionality to be Tested	Relevant Software Change Reports	Validation Team
P-15	<u>Volcanic ash remobilization:</u> ( <i>ashremob.f</i> )	482, 522, 523, 568, 588, 619, 620, 654, 682	R. Benke A. Simpkins
P-16	<u>Dose from direct and inhalation radiation exposure:</u> ( <i>dcags.f</i> )	482, 528, 566, 618, 621,	L. Howard P. Laplante *J. Durham
P-17	<u>Dose from groundwater radiation exposure:</u> ( <i>dcagw.f</i> )	528, 566, 618, 621	P. Laplante L. Howard *J. Durham
P-18	<u>Atmospheric Transport and Deposition of Radionuclides:</u> ( <i>ashplumo.f</i> , <i>ashrmovo.f</i> )	482	D. Hooper R. Rice *R. Benke
* Denotes team member who will serve as a technical reviewer; this member should remain independent of the planning and testing process for each test case.			

### 6.1.1 Validation Task 1: Executive Functions:

The executive (*exec.f*) and several associated routines (*reader.f*, *sampler.f*) handle time stepping, execution of various modules and external codes, and passing of input/output data between various other code modules, routines, and external codes. Much of the functionality of the executive will be validated by the system-level testing described in Section 6.2 of this plan. This process-level should include a review of changes related changes.

**Scope:** The validation team will review SCRs indicated in Table 6.1-1 and conduct additional testing, as necessary to verify the following:

- New routines for assigning inventory to drift segments and for assigning drift segments to subareas are working as intended.
- Inventory is appropriately apportioned between subareas.
- Input, output, and intermediate files are reasonably transparent and traceable.
- Values reported in *tpamin.dat*, *tpameans.dat*, and *tpamax.dat* are correct.

The validation team also may conduct any additional testing deemed appropriate to provide confidence that the executive functions are implemented as intended.

### 6.1.2 Task 2: Subroutine Libraries

Subroutine libraries *numrecip.f* and *nrutil.f* have been updated to include new or revised routines. Much of the functionality of subroutine libraries will be validated by the system-level testing described in Section 6.2 of this plan. The following process-level test should be conducted.

**Scope:** The validation team will review SCRs indicated in Table 6.1-1 and conduct additional testing, as necessary to verify the following:

- New routines FindRootRanges(), FindRootRanges1(), RootFind(), and RootFind1() in *numrecip.f*, as described in SCR 586, are correctly implemented.
- New routines in *nrutil.f* to calculate drip shield outer surface and waste package temperatures, as described in SCR 553, are correctly implemented.

The validation team also may conduct any additional testing deemed appropriate to provide confidence that the utility routines are implemented as intended.

### 6.1.3 Validation Task 3: Inventory Control

The contents of the *burnup.dat* file have been updated as well as their interaction with control parameters in the *tpa.inp* input file. The 'AverageYearofEmplacement' is a new parameter designed to interact with the new design of the *burnup.dat* file. The file now specifies a year of applicability for the file, which is the point in time for which the data was calculated. The year of emplacement can now be changed to an arbitrary value without the need to recalculate the data in the *burnup.dat* file. This interaction should be included in the validation tasks by comparing the output of a run using an adjusted year of emplacement value with the output of a base case run. The initial inventories and final dose values should be appropriately adjusted for the change in the year of emplacement.

**Scope:** Review SCRs indicated in Table 6.1-1 and conduct additional testing, as necessary to verify the following:

- Calculated temperatures for the engineered barrier system are consistent with specified heat outputs in *burnup.dat*.
- When year of emplacement specified in *tpa.inp* is changed to a later time, the computed inventories in *ebspac.nuc* and doses for specific radionuclides in *rgsnr.tpa* should reflect the appropriate exponential decay.

The validation team also may conduct any additional testing deemed appropriate to provide confidence that the inventory control functions are implemented as intended.

### 6.1.4 Validation Task 4: Climate and Subarea-Averaged Net Infiltration

The climate and infiltration abstraction was modified to permit analyses consistent with the proposed 10 CFR Part 63 specification to assume a constant net infiltration rate after 10,000 years. Accordingly, *uzflow.f* and related input data and subroutines have been modified to permit this specification. Other changes include modification to the ITYM standalone code, which resulted in generation of new *maidtbl.dat* and *smaidtbl.dat* data files.

**Scope:** Review SCRs indicated in Table 6.1-1 and conduct additional testing, as necessary to verify the following:

- Parameter values in *tpa.inp* for average net infiltration after 10,000 years are consistent with specified range in draft 10 CFR Part 63 (or final rule if it is available prior to end of validation).
- Correct value for subarea averaged net infiltration is used for constant climate after



- 10,000 years when flag in *tpa.inp* is set to 1 and the correct value based on scaling factor in *climato2.dat* is calculated when the flag is set to 0.
- TPA output either reports or permits calculation of subarea averaged net infiltration at each time step.
- Spatial variability of infiltration among subareas is preserved when constant infiltration after 10,000 years is specified (i.e., specified or sampled value is for entire repository, but different average infiltration rates should be calculated for each subarea).
- Mean annual infiltration after 10,000 years is correctly computed when a correlation to mean annual infiltration at the start is specified.

The validation team also may conduct any additional testing deemed appropriate to provide confidence that the climate and infiltration functions are implemented as intended.

### 6.1.5 Validation Task 5: Near Field Chemical and Temperature Conditions

Numerous changes have been implemented to *nfenv.f* and related input data and subroutines. These changes affect calculated drift-wall, drip-shield and waste-package temperatures, relative humidity, and the chemical environment.

**Scope:** Review SCRs indicated in Table 6.1-1 and conduct additional testing as necessary to verify the following:

- When drift degradation occurs slowly, for reflux computations, the drift is considered open, and for faster rates of drift degradation, the drift is considered degraded. The appropriate dryout thickness data should be selected as a function of the specified drift degradation rate.
- Temperatures for the waste package, drift wall, and drip shield are consistent and include the effects of drift degradation and increasing rubble thickness with time.
- Temperature calculations appropriately account for and distinguish between time of ventilation and time of closure.
- Estimates of seepage versus time are consistent with reflux abstraction and the specified seepage onset temperature.
- Numerical resolution issues or time stepping algorithms that may affect sampled waste package, drip shield, or drift wall temperatures do not result in errors in temperature values used in calculations.
- Temperature, relative humidity, and seepage onset information are correctly used to determine near-field chemical environments. When the relative humidity is low (below a threshold value), the system is considered dry and aqueous corrosion does not take place. If seepage does not contact the waste package, Environment I is assumed to form. Environment II is assumed to occur when the relative humidity is above a threshold value and seepage contacts the waste package. Environment III occurs when the relative humidity is high (close to 90%) and seepage contacts the waste package.
- Changes in pH, chloride and nitrate concentrations with time are correctly computed.

The validation team also may conduct any additional testing deemed appropriate to provide confidence that the near field environment functions are implemented as intended.

### 6.1.6 Validation Task 6: Drift Degradation

The *driftfail.f* module calculates the timing and magnitude of vertical pressure, and degraded

drift heights and equivalent diameters for the radial geometry characterization, using the chimney and trapezoidal degraded-drift configurations. This module replaces the previous drift degradation routine called *eqvdia()*. Vertical pressures are supplied to the *mechfail* code which calculates mechanical drip shield and waste package failure. Equivalent drift diameters are supplied to the *nfenv* module which calculates surface temperatures for the waste package and drip shield.

**Scope:** Review SCRs indicated in Table 6.1-1 and conduct additional testing as necessary to verify the following:

- Drift degradation can be turned off to permit comparisons of scenarios with and without drift degradation.
- Calculated drift degradation times should be consistent with the input distribution for the degradation-only scenario (no seismicity) and the degradation should occur at earlier times when seismicity is included.
- Vertical pressures for the trapezoidal configuration should approach the pressures for the chimney configuration when the trapezoid base angle is increased to 90 degrees.
- Drift heights, vertical pressures, and equivalent diameters are correctly calculated. The following table illustrates eight scenario permutations that should be tested:

Mode	Verify drift height, vertical pressures, and equivalent diameters for these mechanisms.	
	Chimney	Trapezoidal
Drift Degradation only	1	2
Drift Degradation & Default Seismicity	3	4
Drift Degradation & Early Seismicity	5	6
Seismicity only	7	8

The validation team also may conduct any additional testing deemed appropriate to provide confidence that the drift degradation functions are implemented as intended.

#### 6.1.7 Validation Task 7: Mechanical Failure of Drip Shields and Waste Packages

The *mechdrive.f* module executes the *mechfail.f* standalone code to calculate drip shield and waste package failures based on rock load and seismic acceleration inputs. This functionality has been significantly revised to incorporate more recent design information and independent analyses of mechanical interactions between drip shields and waste packages (Ibarra, et al., 2006), and to improve user control of seismic hazard curve specifications.

**Scope:** Review SCRs indicated in Table 6.1-1 and conduct additional testing as necessary to verify the following:

- The *mechdrive.f* module provides correct input to the *mechfail.f* code
- Time of drip shield failure is correctly calculated by *mechfail*.
- Seismic frequency and magnitude distributions are correctly calculated in terms of peak ground velocity, peak ground acceleration, and compaction factors.
- Mechanical failures do not occur prior to repository closure.
- Rock load inputs from the *driftfail.f* module, drip shield-waste package contact area, and increase in load caused by seismic acceleration are properly calculated.
- The number of calculated waste package failures within 10ky, 100ky, and 1Myr time frames are compared to independent calculations to ensure results are consistent with the sampled input functions for drift degradation rates, bulking factors, contact angle of drip shield bulkheads with the waste package, and the sampling of seismic events.
- Changes in waste package and drip shield thickness information passed from the drip shield and waste package corrosion abstractions are correctly utilized in failure calculations.
- No drip shield-waste package mechanical failures occur after the time when drip shields corrosion failure is calculated.
- Drip shield mechanical failure and resulting mechanical interaction can be turned off to permit comparison of scenarios with and without mechanical failures.

The validation team also may conduct any additional testing deemed appropriate to provide confidence that the mechanical interaction and seismic event sampling functions are implemented as intended.

#### **6.1.8 Validation Task 8: Drip Shield Corrosion**

The *dsfail.f* module executes the *dsfailt.f* code to compute changes in drip shield thickness with time due to corrosion and to determine the time of failure due to corrosion. Changes to *dsfailt.f* include the addition of time dependent drip shield thickness, moving several inputs from the *tpa.inp* file to the *dsfailt.def* file, and addition of a failure depth threshold to permit specification of fractional values of the drip shield initial thicknesses as a failure criterion.

**Scope:** Review SCRs indicated in Table 6.1-1 and conduct additional testing as necessary to verify the following:

- The *dsfail.f* module provides correct input to the *dsfailt.f* code
- Corrosion rates, initial thickness, and other values specified in *dsfailt.def* are appropriate.
- Time of drip shield corrosion failure is correctly calculated by *dsfailt.f*.
- Changes to specified failure depth thresholds result in proportional changes to calculated corrosion failure time.

The validation team also may conduct any additional testing deemed appropriate to provide confidence that the drip shield corrosion functions are implemented as intended.

#### **6.1.9 Validation Task 9: Waste Package Corrosion**

The *ebsfail.f* module executes the *failt.f* code to compute waste package corrosion as a function of the time. Separate calculations are performed for general corrosion on the waste package body, localized corrosion penetration of weld areas, and localized corrosion penetration of the waste package body. Changes to *ebsfail.f* include updated equations for corrosion potential,

temperature dependence of passive current density, consideration of effects of chloride/inhibitor ratio in localized corrosion, subarea variability of chemistry parameters, and addition of a failure depth threshold to permit specification of fractional values of initial waste package thicknesses as a failure criterion.

**Scope:** Review SCRs indicated in Table 6.1-1 and conduct additional testing as necessary to verify the following:

- The *ebsfail.f* module provides correct input to the *failt.f* code
- Time varying corrosion potentials and passive current densities are correctly calculated.
- Modification to the critical potential to account for the action of inhibiting anions is correctly calculated.
- Times of failures by general corrosion and localized corrosion on welds and the waste package body are correctly reported by *failt.f*.
- Changes to specified failure depth thresholds (i.e., fraction of total thickness at which waste package is assumed to have failed) result in proportional changes to calculated corrosion failure times.
- Localized corrosion does not occur until the drift wall temperature falls below a threshold value for onset of seepage.

The validation team also may conduct any additional testing deemed appropriate to provide confidence that the waste package corrosion functions are implemented as intended.

#### **6.1.10 Validation Task 10: Releases from the Engineered Barrier System**

The *ebsrel.f* module executes the *releaset.f* code to compute releases of radionuclides from the engineered barrier system. Separate calculations are performed for the amount of seepage contacting waste, waste form exposure and dissolution rates, and mass releases of radionuclides from waste packages. Changes to *ebsrel.f* include consideration of drift degradation effects on seepage, integration of flow diversion factors with mechanical and corrosion processes, accounting for uncertainty in the number of corrosion and mechanical failures contributing to release, addition of a threshold temperature above which water does not contact the waste form, modification of the representative waste package volume, removal of carbon-14 parameters, reallocation of source terms to permit separate accounting of releases when mechanical and corrosion failures are predicted in the same subarea during a realization, and selection of the flow-through model as the default for all failure modes.

**Scope:** Review SCRs indicated in Table 6.1-1 and conduct additional testing as necessary to verify the following:

- The numbers of waste packages contributing to release by mechanical and localized corrosion failure modes are adjusted to account for uncertainties related to the number of packages that may fail by that mode and the number failed packages that would allow water to enter.
- Seepage does not contact the waste form until the waste package temperature falls below the specified or sampled threshold value.
- Seepage reduction factors appropriately account for time varying protection offered by drip shields and waste packages as corrosion and other degradation processes proceed.
- Mass balance between total inventory and amount of radionuclides released is

- preserved and accounts for radioactive decay.
- The use of a waste package control volume in mass release calculations does not bias calculated releases.
- Diffusive releases are deactivated for the reference case by setting the `FractionOfWPsWithDiffusionTilt[]` input parameter to a constant value of 0.0.
- Instantaneous exposure of the waste form, as opposed to gradual exposure, is included in the reference case by setting the `FuelRodHalfLength[m]` input parameter to 0.0.

The validation team also may conduct any additional testing deemed appropriate to provide confidence that the engineered barrier source term functions are implemented as intended.

#### **6.1.11 Validation Task 11: Colloid Facilitated Releases from the Engineered Barrier System**

In addition to the functionality described for Task 10, The *ebsrel.f* module and *releaset.f* code have been modified to account for irreversible colloid attachment of radionuclides released from the engineered barrier subsystem. Irreversible attachment can be simulated by specifying a fraction of the release for a particular radionuclide. This fraction is assigned to a set of new (artificial) radionuclides, referred to as J-species, to represent colloidal species that possesses transport properties appropriate for colloids. The four elements allowed to bind irreversibly to colloids are plutonium, americium, thorium, and curium. The modified abstraction also accounts for reversible sorption to colloids by adjustment of solubility limits for plutonium, americium and uranium.

**Scope:** Review SCRs indicated in Table 6.1-1 and conduct additional testing as necessary to verify the following:

- Adjustments to effective solubility of U, Am, and Pu to account for reversible sorption to colloids are appropriately implemented.
- Finite sorption capacity of colloids is appropriately implemented and calculations for competitive sorption on colloids correctly accounts for the sorption affinities of the different nuclides.
- Colloid-facilitated releases can be turned off to permit comparison of scenarios with and without colloids.

The validation team also may conduct any additional testing deemed appropriate to provide confidence that the colloid source term functions are implemented as intended.

#### **6.1.12 Validation Task 12: Unsaturated Zone Flow and Transport**

The *uzft.f* module executes the *nefmks.f* code (i.e., NEFTRAN II) to calculate the transport of released radionuclide species from the unsaturated zone to the water table. Changes to this abstraction include code modifications to account for both reversible and irreversible colloid-facilitated transport, colloid filtration, addition of new repository subareas with new input values, modifications to the layer-selection algorithm to prevent skipping of unsaturated zone transport calculations, creation of new *uz\_kdrd.out* and *uz\_revers.out* intermediate output files for traceability of retardation coefficient calculations, modifications to include at least one matrix transport layer in every transport calculation, and changes to the fast-flow bypass calculation so that only the portion of flow that cannot be accommodated by the most permeable matrix layer is bypassed.

**Scope:** Review SCRs indicated in Table 6.1-1 and conduct additional testing as necessary to verify the following:

- The most permeable matrix layer in each subarea is selected for use in each transport calculation.
- The mass fraction of the release term allowed to bypass unsaturated zone transport calculation is equal to the portion of flow that cannot be accommodated in the most permeable matrix layer.
- Partition and retardation coefficients are correctly adjusted to account for reversible colloid transport and correctly reported in *uz\_revers.out* are correctly calculated.
- Calculated partition and retardation coefficients for actinides are correctly reported in *uz\_kdrd.out*.
- Colloid filtration is appropriately applied based on the selected transport layer with the highest specified or sampled value of the filtration factor.

The validation team also may conduct any additional testing deemed appropriate to provide confidence that the unsaturated zone transport functions are implemented as intended.

#### **6.1.13 Validation Task 13: Saturated Zone Flow and Transport**

The *szft.f* module executes the *nefmks.f* code (i.e., NEFTRAN II) to calculate the transport of released radionuclide species from the water table beneath the repository to the receptor location. Changes to this abstraction include code modifications to account for both reversible and irreversible colloid-facilitated transport, addition of new repository subareas with new input values, modification of the flow path data specified in the *strmtube.dat* file, and creation of new *sz\_kdrd.out* and *sz\_revers.out* intermediate output files for traceability of retardation coefficient calculations.

**Scope:** Review SCRs indicated in Table 6.1-1 and conduct additional testing as necessary to verify the following:

- Releases to the saturated zone for each repository subarea are assigned to the correct stream tube in *strmtube.dat* for use in transport calculations.
- Repository Partition and retardation coefficients are correctly adjusted to account for reversible colloid transport and correctly reported in *sz\_revers.out* are correctly calculated.
- Calculated partition and retardation coefficients for actinides are correctly reported in *sz\_kdrd.out*.

The validation team also may conduct any additional testing deemed appropriate to provide confidence that the saturated zone transport functions are implemented as intended.

#### **6.1.14 Validation Task 14: Intrusive Volcanism**

The geometric model of igneous waste package failure was modified to include the 'magma filled drift' mechanism. For this model all waste packages contained in a drift intersected by a dike are failed. Determining the number failed in a subarea requires knowledge of dike, panel, drift, and subarea geometries. The subarea geometry is provided in the *tpa\_include.inp* file for the base case data set. The drift geometry is computed and provided in the *drifts.dat* file. The panel geometry is provided in the *repdes.dat* file. The dike geometry (center, angle, and length)

is sampled from the *tpa.inp* parameters.

**Scope:** Review SCRs indicated in Table 6.1-1 and conduct additional testing as necessary to verify the following:

- The algorithm that predicts the timing of volcanic events provides results consistent with statistical probability and is not unduly biased.
- The geometric and distribution models for computing the number of failed waste packages for any subarea is correctly implemented and number of failed packages is consistent with number of intersected drifts.
- Counting intersected waste packages as both a groundwater source and an ejected source does not unduly bias results.
- The number reported in the screen print should agree with the number displayed in the *wpsfail.res* file.

The validation team also may conduct any additional testing deemed appropriate to provide confidence that intrusive volcanism functions are implemented as intended.

#### **6.1.15 Validation Task 15: Volcanic Ash Remobilization**

A new model for calculating exposures due to redistribution of volcanic ash was implemented through the *ashremob.f* module and associated utilities and subroutines. This model is now the preferred model for evaluating effect of extrusive volcanism; however the capability of the previous model, implemented through the *ashplumo.f*, *ashrmovo.f*, and *dcags.f* modules, is still preserved in TPA Version 5.1.

**Scope:** Review SCRs indicated in Table 6.1-1 and conduct additional testing as necessary to verify the following:

- Outdoor and indoor contributions to the total inhalation dose function as intended.
- Waste concentration factor functions properly and produces the intended effect on the resulting doses for realizations with an initial tephra deposit at the receptor location.
- Fluvial contributions work as intended (e.g., compare test cases with 100 percent fluvial contribution and evaluate how long Fortymile Wash yields remobilized tephra).
- Eolian contributions work as intended (e.g., compare test cases with 100 percent eolian contribution and evaluate the magnitude of the eolian dilution factor).

The validation team also may conduct any additional testing deemed appropriate to provide confidence that the ash remobilization functions are implemented as intended.

#### **6.1.16 Validation Task 16: Dose from Direct and Inhalation Radiation Exposure**

The *dcags.f* module is used to calculate doses based on direct or inhalation radiation exposures. This module was modified to interface with the updated *ashremob.f* module. Additionally, input values for dose coefficients were updated to be consistent with International Commission on Radiological Protection (ICRP)–72 Methodology.

**Scope:** Review SCRs indicated in Table 6.1-1 and conduct additional testing as necessary to verify the following:

- Dose values from ASHREMOB in file *rgsnr.tpa* are consistent with the dose conversion factors in *gs\_\*.dat*.
- Dose coefficients are consistent with ICRP 72 Methodology.

The validation team also may conduct any additional testing deemed appropriate to provide confidence that the direct and inhalation dose calculation functions are implemented as intended.

#### 6.1.17 Validation Task 17: Dose from Groundwater Radiation Exposure

The *dcagw.f* module is used to calculate doses based on exposures from groundwater pathways. The *gnewdf.dat* dose coefficient input data file was updated to be consistent with ICRP-72 Methodology.

**Scope:** Review SCRs indicated in Table 6.1-1 and conduct additional testing as necessary to verify the following:

- Dose coefficient values in *gnewdf.dat* are consistent with ICRP 72 Methodology.
- Age groups selected in the *tpa.inp* file result in selection of appropriate values in *gw\_\*.dat* files for the age group selected.

The validation team also may conduct any additional testing deemed appropriate to provide confidence that the groundwater pathway dose calculation functions are implemented as intended.

#### 6.1.18 Validation Task 18: Atmospheric Transport and Deposition of Radionuclides

Although a new model for calculating exposures due to redistribution of volcanic ash was implemented through the *ashremob.f* module, the capability of the previous model, implemented through the *ashplumo.f* and *ashrmovo.f* modules, is still preserved in TPA Version 5.1.

**Scope:** Review the SCR indicated in Table 6.1-1 and conduct additional testing as necessary to verify the following:

- The alternative model that utilizes the *ashplumo.f* and *ashrmovo.f*, as implemented in previous version of the TPA code, is still functional.

Note that Validation Task 16 will likely require execution of *ashplumo.f* and *ashrmovo.f*, thus, it may be possible to accomplish this simple verification by reference to Task 16. The validation team also may conduct any additional testing deemed appropriate to provide confidence that the extrusive volcanism functions are implemented as intended.

## 6.2 System-Level Testing

Table 6.2-1 lists system-level validation tasks for the major TPA code functionalities that have been significantly revised since TPA Version 5.0s. These code functionalities may be represented by one or more individual modules, submodules, or routines. The last column in Table 6.2-1 identifies the technical staff members who will likely be involved in the validation test design, execution, and technical review of each task. These names could change depending on staff availability. All tests should be documented in the SVRs and major testing



results will be presented to the TPA Technical Advisory Panel and Yucca Mountain Team. Each SVR will receive technical and programmatic reviews to ensure requirements of Technical Operating Procedure TOP -018 are met. A template cover sheet to be used for the SVR is provided in the Appendix.

System-level testing of the TPA Code Version 5.1 will be accomplished concurrent with completion of the process-level tasks. This test activity will include several test cases that evaluate various combinations of input parameters and disruptive event scenarios that can be used to ensure stable operation (i.e., code does not crash) and reasonableness of results (i.e., results are explainable), and to provide confidence that changes to one module or auxiliary code do not introduce problems or errors in a different code module. This activity will take a risk-informed approach by giving closer examination to those model parameters and model abstractions that have the most significant effect on the model results.

<b>Table 6.2-1. System-Level Validation Tasks List</b>		
<b>SVR #</b>	<b>Functionality to be Tested</b>	<b>Validation Team</b>
S-1	<u>Waste Package Failure Modes:</u> ( <i>ebsfail.f</i> , <i>mechdriver.f</i> , <i>faulto.f</i> , <i>volcano.f</i> )	C. Grossman, J. Gwo, O. Pensado, *S. Mohanty
S-2	<u>Radionuclide Release Rates:</u> ( <i>ebsrel.f</i> , <i>uzft.f</i> , <i>szft.f</i> , <i>ashremob.f</i> )	O. Pensado, C. Grossman, J. Gwo, T. Ghosh, D. Codell, *S. Mohanty
S-3	<u>Radionuclide Dose Contributions:</u> ( <i>dcagw.f</i> , <i>dcags.f</i> , <i>ashremob.f</i> )	C. Grossman, T. Ghosh, K. Compton, *S. Mohanty
S-4	<u>Numerical Stability:</u>	R. Rice, C. Grossman, D. Codell, *S. Mohanty
*Denotes team member who will serve as a technical reviewer; this member should remain independent of the planning and testing process for each test case.		

### 6.2.1 Validation Task S-1: Waste Package Failure Modes

The software has been modified to reconsider various processes potentially leading to failure of the waste packages including manufacturing defects, localized corrosion, mechanical loading from drift degradation and seismicity, general corrosion, igneous intrusion, and faulting. This task seeks to understand which failure modes dominate and the causes for the magnitude and timing of the failures. The following system-level tests should be conducted.

**Scope:** The validation team will conduct the testing, as necessary, to understand the following:

- For the nominal scenario, reference case, which failure modes (i.e., initial defects, localized corrosion, mechanical, or general corrosion) dominate for various time periods of interest (e.g., 10,000 years or 1,000,000 years) and which input data have the greatest effect on occurrence of those failure modes.
- For the disruptive scenario, both igneous and faulting cases, which failure modes

- dominate for various time periods of interest (e.g., 10,000 years or 1,000,000 years) and which input data have the greatest effect on occurrence of those failure modes.
- For realizations that return multiple failure modes in a single subarea (e.g., initial defects, localized corrosion, and mechanical), resulting source terms should be consistent with the numbers of packages failed by each mode (i.e., no double counting).

The validation team should demonstrate a reasonable understanding of the causes for dominant and minor failure modes for the given cases of the scenarios tested and that the resulting failures are consistent with the understanding of the expected timing and magnitude of the particular failure mechanisms. The validation team should indicate in the SVR a brief summary of this testing and associated results. The validation teams may conduct additional testing, to be documented in the SVR, as necessary to understand cases leading to the resulting failure modes.

### **6.2.2 Validation Task S-2: Radionuclide Release Rates**

This task seeks to understand which radionuclides dominate releases from failed waste packages, from the unsaturated zone, and from the saturated zone during various time periods of interest and which processes affect radionuclide release rates. The following system-level tests should be conducted.

**Scope:** The validation team should conduct the following types of evaluations:

- For the nominal scenario, evaluate the releases of key radionuclides (e.g., Tc-99, Np-237, and Pu-239/240), dissolved and colloidal, from a single failed waste package (e.g., initial defects failure) at various time periods of interest (e.g., 1,000 years, 10,000 years, and 100,000 years) including unsaturated and saturated zone releases.
- For the nominal scenario, evaluate the releases of key radionuclides (e.g., Tc-99, Np-237, and Pu-239/240), dissolved and colloidal, from failed waste packages (e.g., initial defects failure) in a single subarea at various time periods of interest (e.g., 1,000 years, 10,000 years, and 100,000 years) including unsaturated and saturated zone releases.
- For the nominal scenario, evaluate the releases of key radionuclides (e.g., Tc-99, Np-237, and Pu-239/240), dissolved and colloidal, from failed waste packages for the entire repository at various time periods of interest (e.g., 1,000 years, 10,000 years, and 100,000 years) including unsaturated and saturated zone releases.

Comparisons of the above test cases should be conducted to evaluate whether dose results scale proportionally based on the numbers of waste packages affected and to understand whether distribution of failures among subareas has any significant effect on results. The validation team should demonstrate that code results permit determination of the causes for key radionuclide releases. This comparison of tests should demonstrate that radionuclide releases from failed waste packages are consistent with inventories, solubility limits, water flux, colloid concentrations and fuel degradation rates. Releases of radionuclides from the unsaturated zone should be reasonably consistent with distribution coefficients and earlier engineered barrier system releases. Releases of radionuclides from the unsaturated zone should be reasonably consistent with retardation coefficients and unsaturated zone releases. Releases from the entire repository should be reasonably consistent with releases from subareas and individual waste packages accounting for appropriate scaling effects. Releases should also be reasonably consistent for various time periods of interest. The validation team should indicate in the SVR a brief summary of this testing and associated results. The validation teams may

conduct additional testing, to be documented in the SVR, as necessary to understand cases leading to the resulting failure modes.

### **6.2.3 Validation Task S-3: Radionuclide Dose Contributions**

This task seeks to understand which radionuclides dominate dose rates to the RMEI during various time periods of interest and which processes affect dose rates. The following system-level tests should be conducted.

**Scope:** The validation team will conduct the testing, as necessary, to understand the following:

- For the nominal scenario, evaluate the dose rates of key radionuclides (e.g., Tc-99, Np-237, and Pu-239/240), dissolved and colloidal, from a single failed waste package (e.g., initial defects failure) at various time periods of interest (e.g., 1,000 years, 10,000 years, and 100,000 years).
- For the nominal scenario, evaluate the doses from key radionuclides (e.g., Tc-99, Np-237, and Pu-239/240), dissolved and colloidal, from failed waste packages (e.g., initial defects failure) in a single subarea at various time periods of interest (e.g., 1,000 years, 10,000 years, and 100,000 years).
- For the nominal scenario, evaluate the doses from key radionuclides (e.g., Tc-99, Np-237, and Pu-239/240), dissolved and colloidal, from failed waste packages for the entire repository at various time periods of interest (e.g., 1,000 years, 10,000 years, and 100,000 years).
- For the disruptive scenario, igneous and faulting cases, evaluate the doses from key radionuclides from failed waste packages for the entire repository at various time periods of interest.

The validation team should demonstrate a reasonable understanding of the causes for key radionuclide doses. This tests should demonstrate that radionuclide doses are reasonably consistent with saturated zone releases from Task S-2. Doses from the entire repository should be reasonably consistent with doses from subareas and individual waste packages accounting for appropriate scaling effects. Doses should also be reasonably consistent for various time periods of interest. The validation team should indicate in the SVR a brief summary of this testing and associated results. The validation teams may conduct additional testing, to be documented in the SVR, as necessary to understand cases leading to the resulting failure modes.

### **6.2.4 Validation Task S-4: Numerical Stability**

The software should exhibit reasonable statistical stability in the main results for a variety of simulation time periods of interest. The following system-level tests should be conducted.

**Scope:** The validation team will conduct the testing, as necessary, to verify the following:

- Stability of the mean dose results for the nominal scenario for various time periods of interest and numbers of realizations.
- Stability of the mean dose results for the igneous and faulting cases of the disruptive scenario for various time periods of interest and numbers of realizations.
- Stability of dose results as a function of time step size.

The validation team should verify mean dose results are stable within a reasonable range. The validation team should indicate in the SVR a brief summary of this testing and associated results. The validation teams may conduct additional testing, to be documented in the SVR, as necessary to understand cases leading to the resulting failure modes.

## **7. SCHEDULE**

The following schedule indicates key progress tracking points during the validation process that should be met in order to ensure the June 22, 2007 deadline is met for delivery of the TPA Version 5.1 code to NRC. Except for the final delivery date, the scheduled objectives do not represent contractual milestones. Rather, the schedule dates will be used to assess whether satisfactory progress is being made and to identify when it is necessary to reallocate resources to achieve on-time delivery.

- |          |   |  |
|----------|---|--|
| March 12 | — | Final validation plan and validation-ready code are available to test teams. Validation teams begin to review SRD requirements and SCRs and to plan specific test cases for process- and system-level tasks. Planned test cases should be submitted for technical review as soon as they are developed. Testing should proceed while awaiting technical review feedback. |
| March 23 | — | Technical reviews are completed for planned test cases for all process-level tasks (Table 6.1-1) and SVRs have been updated as necessary to address technical review comments. Testing continues.  |
| March 29 | — | System level test cases and preliminary results are presented for review by TPA Advisory Panel. Documentation of the technical review of planned system level is accomplished concurrently.  |
| March 30 | — | Planned test cases for all system-level tasks (Table 6.2-1) have been updated in SVRs as necessary to address technical review comments. Testing continues.  |
| April 13 | — | Preliminary validation test results are obtained and any problems or errors are identified for all process- and system-level test cases.   |
| April 15 | — | Software changes are initiated for all problems or errors that were identified.  |
| April 27 | — | Testing for validation tests that have passed are documented in SVRs and submitted for technical and programmatic reviews.   |
| May 18   | — | Software changes to correct problems or errors are completed and validation testing on those items is repeated. (Simple review of SCR testing of corrected problem may be sufficient in some cases). Change review committee (see section 6.1) reviews all SCRs to determine whether any previous validation tests need to be repeated.                                  |
| May 25   | — | Technical and programmatic reviews are completed for the initial set of SVRs that passed testing. Testing that required software changes is completed and of SVRs are submitted for technical and programmatic reviews.  |

- June 13 — Technical and programmatic reviews for all SVRs are completed.
- June 20 — SVR and SCR are bundled and provided for QA review and records. All code configuration control needs are completed and code is compiled for delivery. A software release notice for TPA version 5.1 is issued.
- June 21 — Transmittal letter prepared and TPA Version 5.1 code is delivered to NRC on CDROM.

## Appendix

### SOFTWARE VALIDATION REPORT

SVR#:	Project#:	
Software Name:		Version:
Test ID:	Test Series Name:	
<b>Test Method</b> <input type="checkbox"/> code inspection <input type="checkbox"/> output inspection <input type="checkbox"/> hand calculation <input type="checkbox"/> spreadsheet <input type="checkbox"/> graphical <input type="checkbox"/> comparison with external code results		
Test Objective:		
<b>Test Environment Setup</b> Hardware (platform, peripherals): Software (OS, compiler, libraries, auxiliary codes or scripts): Input Data (files, data base, mode settings):		
Assumptions, constraints, and/or scope of test:		
Test Procedure:		
<b>Test Results</b> Location: Test Criterion and Analysis of Results:  Test Evaluation (Pass/Fail):		
Notes:		
Tester:	Date:	