

**SCALE Procedure for Verified,
Archived, Library of Inputs and Data (VALID)**



T. M. Greene
W. J. Marshall

September 2024

SCALE Procedure for VALID

Prepared by:

Travis Greene

9/17/2024

Travis Greene, Research and Development Associate
Nuclear Energy and Fuel Cycle Division

Date

Reviewed by:

LaTravia Harmon, Fusion and Fission Energy and Science Directorate
Quality Manager
Quality Assurance Division

Date

Approved by:

William Wieselquist

9/17/2024

William Wieselquist, SCALE Director
Nuclear Energy and Fuel Cycle Division

Date

Approved by:

William David Pointer

9/17/2024

Dave Pointer, Director
Nuclear Energy and Fuel Cycle Division

Date

SCALE Procedure for VALID

REVISION HISTORY

Revision Number	Effective Date	Revision Description
1	09/15/2010	Added Revision History Log Sheet
1	09/15/2010	Throughout: The name of the process/library is changed from “Models and Derived Data (MADD) to “Verified, Archived Library of Inputs and Data” (VALID). Minor editorial or other non-intent changes were made.
1	09/15/2010	Section 3.1: Definition of “Model” clarified to include non-critical or hypothetical systems.
1	09/15/2010	Section 5.1.2: Examples of reasons for rejecting a request for a VALID form were added.
1	09/15/2010	Section. 5.1: Renumbered all steps following the new step 5.1.11
1	09/15/2010	Section 5.1.14: Added a note that the SQA coordinator provides backup copies of VALID forms and attachments on the ORNL Electronic Records System (ERS). This change matches current practice: the ERS is the only location where NSTD/NMDS staff can readily access these records.
1	09/15/2010	Section 5.1.15: Formerly stated that the SQA Coordinator notify “all users” of the library for each change to library. This is neither practical nor necessary. A specific list of individuals or positions to be notified is now identified.
2	05/22/2013	The purpose for this revision has been added to Section 1.
2	05/22/2013	The applicability of this revision to work started under previous revisions has been added to Section 2.
2	05/22/2013	The definition of Model has been updated in Section 3.1 and the definition of VALID Case has been added as Section 3.4.
2	05/22/2013	A new Section 5 has been added providing general guidance on minimum standards for qualification and addressing expiration of qualifications.
2	05/22/2013	The Procedure as defined in what is now Section 6 has been significantly modified to support migration of the VALID procedure from a form-based system to the FogBugz tracking system.
3	05/17/2024	Updates to Sections 1 and 2 to document the purpose of the revision
3	05/17/2024	The responsibilities of the requestor were officially defined in a new Section 5.1.
3	05/17/2024	Updates to Sections 5.2–5.5 to reflect responsibilities in the new procedure
3	05/17/2024	Significant updates to Section 7.0 to describe the new GitLab-based process incorporating use of the repository

SCALE Procedure for VALID

ACRONYM LIST

C/E	calculated/expected values
DevOps	development and operations
DP	direct perturbation
ERS	Electronic Records System
ICSBEP	International Criticality Safety Benchmark Evaluation Project
MADD	Models and Derived Data
ORNL	Oak Ridge National Laboratory
QAC	Quality assurance coordinator
SCALE	Standardized Computer Analyses for Licensing Evaluation
SDF	sensitivity data file
VALID	Verified, Archived, Library of Inputs and Data
VaNDaL	Validation of Nuclear Data Libraries
VCL	VALID change log

TABLE OF CONTENTS

REVISION HISTORY iii

ACRONYM LIST iv

1.0 INTRODUCTION..... 1

2.0 PURPOSE 1

3.0 SCOPE..... 1

4.0 DEFINITIONS 2

 4.1 Model 2

 4.2 Derived Data 2

 4.3 Repository 2

 4.4 Library 2

 4.5 Case 2

 4.6 GitLab 2

 4.7 FogBugz 2

5.0 RESPONSIBILITIES..... 3

 5.1 Requestor 3

 5.2 Originator 3

 5.3 Reviewer 3

 5.4 VALID QAC..... 3

 5.5 SCALE Director 4

 5.6 Users 4

6.0 QUALIFICATION..... 4

 6.1 Originator Qualifications..... 5

 6.2 Reviewer Qualifications 5

 6.3 Expiration of Qualifications..... 5

7.0 PROCEDURE 5

 7.1 Initial Addition of Inputs and Data to the Library 5

 Step 0 – Proposed 5

 Step 1 – Approved/Rejected 6

 Step 2 – In Progress..... 7

 Step 3 – In Staging..... 7

 Step 4 – In Review 8

 Step 5 – Final Check and Merge..... 8

 Step 6 – Notification and Closure 8

 7.2 Use of Inputs and Data..... 9

 7.3 Configuration Management of Inputs and Data 9

 7.4 Revisions or Corrections of Library Content 9

 Step 0 – Identification and Isolation 9

 Step 1 – In Review 10

 Step 2 – Revision 10

 Step 3 – Finalization/Removal 10

8.0 REFERENCES..... 10

1.0 INTRODUCTION

This procedure provides a framework for preparing, reviewing, and storing model inputs and derived data so that individuals with authorized access to the Verified, Archived, Library of Inputs and Data (VALID) repository can use the inputs and data with confidence in their analyses. This procedure uses documented checks and reviews to ensure that the inputs and data were correctly generated using appropriate references. Configuration management is implemented to prevent inadvertent modification of the inputs and data or inclusion of models that have not been reviewed. This procedure also provides guidance to be followed if errors are identified or if input or data revisions are needed.

Although this procedure addresses quality assurance for nuclear models, inputs, and derived data, it remains the responsibility of each user of the inputs and data to verify that their usage complies with any specific quality assurance requirements of their individual sites or projects.

Recent experience with code intercomparisons has shown that this procedure and its inherent strict standards yield high-quality, reliable models with fewer errors than other validation suites. This procedure is also part of the basis of the Los Alamos Benchmark Suite process and the Organization of Economic Co-operation and Development / Nuclear Energy Agency Working Party on International Nuclear Data Evaluation Co-operation subgroup 45 effort related to Validation of Nuclear Data Libraries (VaNDaL).

2.0 PURPOSE

The purpose of this procedure is to prescribe the process used to create and maintain a collection of model inputs and associated derived data for use in nuclear analyses and the validation of computational methods.

The purpose of this revision is to migrate the procedure from the use of the FogBugz issue-tracking system to a VALID project within the GitLab issue-tracking system. The GitLab interface will also be used to implement control for files in the VALID repository. The repository manages the chain of custody for the files, streamlining documentation and review. The repository also provides a more explicit change log for each file than was maintained previously. Repository use will also facilitate mirroring of the library to an external repository, providing prompt and accessible updates for external users. Future expansion of GitLab implementation at Oak Ridge National Laboratory (ORNL) may also facilitate external users having direct access to the internal system, thus allowing them to perform origination or review under this procedure. The core process of two qualified, independent individuals generating and reviewing the technical content of each VALID case remains the basis for quality in these models.

3.0 SCOPE

This procedure provides the process for preparing, reviewing, and documenting the quality of model inputs and data derived from model inputs.

Applicability of this procedure to all work in progress under previous revisions of this procedure will be determined on a case-by-case basis by the VALID quality assurance coordinator (QAC).

4.0 DEFINITIONS

4.1 Model

In this procedure, a model is a set of computational input data that is used to describe a system of interest. For example, models of critical experiments are frequently used to validate criticality safety analyses. Models may be developed for hypothetical or representative systems, such as the GBC-32 cask, or for actual process-facility applications. A model is a complete set of inputs—such as (but not limited to) geometry, materials, cross section library, cross section processing treatments—that represents the benchmark or system, volume calculations, grid meshes, boundary conditions, and start types.

4.2 Derived Data

Derived data are data produced from computer calculations using a model. Nuclide-, reaction-, and energy-dependent k_{eff} sensitivity data files are examples of data derived from criticality safety analysis models.

4.3 Repository

The VALID repository is the central storage location of information housed through GitLab.

4.4 Library

The library is the collection of verified models and derived data that is available for use.

4.5 Case

A VALID case is a set of experiments and/or data for which models and derived data are being generated for addition to the library. A VALID case can contain one or more benchmark evaluations, radiochemical assay sample calculations, application models, etc. Each VALID case moves through this procedure as a single unit. Thus, multiple models can be entered into the library during the origination and review of a single case. Each VALID case is documented in a single GitLab issue.

4.6 GitLab

GitLab is a DevOps (development and operations) software package that can develop, secure, and operate software.

4.7 FogBugz

FogBugz is a DevOps software package that tracks bugs, issues, and feature requests as cases in one place.

5.0 RESPONSIBILITIES

5.1 Requestor

The requestor is responsible for the following:

- Identifying models and data that could be added to the library
- Identifying acceptable references describing models

5.2 Originator

The originator is responsible for the following:

- Developing models and executing them to generate the resulting derived data
- Documenting methods and data used to prepare the models and derived data
- Documenting any approximations or assumptions used in the development of models, including potential inconsistencies or inadequacies inherent to the source references and any experimental or modeling uncertainties
- Committing files to be added to the library to the appropriate repository branch and creating the merge request for review
- Notifying the VALID QAC of any errors or deficiencies in the references used to create the models
- Checking the models and derived data to ensure the model inputs are correct and the derived data are correctly calculated
- Submitting models and data for addition to the library
- Resolving review comments

5.3 Reviewer

The reviewer is responsible for the following:

- Ensuring documentation prepared by the originator is complete and accurate
- Ensuring model references are appropriate
- Confirm that the models identified by the Requestor were modeled by the Originator
- Reviewing models and derived data
- Working with the originator to resolve review comments
- Documenting reviews

5.4 VALID QAC

The QAC is responsible for the following:

- Assigning originators and reviewers to prepare and review models for the library
- Ensuring individuals assigned as originators or reviewers have backgrounds and experience levels appropriate for their responsibilities under this procedure
- Considering the expected use of the models and data and determining whether the references are complete, accurate, credible, and appropriate
- Assisting with the resolution of any unresolved review comments

SCALE Procedure for VALID

- Ensuring documentation is complete
- Accepting or rejecting proposed models and derived data
- Approving merge requests to add new library files to the repository
- Reviewing problems reported with models or data in the repository
- Ensuring users are aware of their responsibilities
- Maintaining a restricted-access archive of all files permanently deleted from the VALID repository
- Assigning VALID change log (VCL) numbers through the GitLab bot to cases and maintaining records of the VCL numbers that have been issued
- Maintaining qualification lists for originators and reviewers and documentation justifying their qualifications
- Maintaining an email notification list of individuals to be informed of any changes to the VALID repository

Within the VALID repository, certain cases may contain proprietary data, requiring that access permissions for those directories be limited to specific individuals. This entails creating separate submodules in the GitLab VALID repository in which to place the proprietary data with the specified access controls. The QAC creates the submodule in the repository and applies directory access permissions as directed by the responsible SCALE project manager, and then emails notifications that are limited to the individuals of the access group.

5.5 SCALE Director

The SCALE Director is responsible for the following:

- Ensuring this procedure is maintained and followed
- Assigning a qualified staff member as the QAC
- Providing management direction in response to disagreements concerning the provisions of this procedure, such as required level of checking and reviewing or reference acceptability

5.6 Users

Users of approved models and derived data from the library are responsible for the following:

- Notifying the QAC if problems are identified with models or derived data already in the repository
- Ensuring that their use of the models or derived data obtained from the repository meets the quality assurance requirements for their work
- Requesting that their names be included in the email notification list maintained by the QAC

6.0 QUALIFICATION

As stated in Section 5.4 of this procedure, the QAC is responsible for ensuring the qualification of individuals acting as originators or reviewers for each VALID case. General guidance for minimum requirements for qualification is provided in this section. The QAC approves each

qualification, so deviations from these recommendations are allowed on a case-by-case basis. All individuals qualified under previous revisions of this procedure maintain their qualifications. Before performing work on a VALID case under this revision, however, they should review and understand the procedural changes instituted in Section 7.0 of this procedure. There is no separate qualification for each role.

6.1 Originator Qualifications

Typically, an originator will be an experienced user of the computer code or codes to be used in the VALID case. The individual will also have reviewed this procedure and provided documentation of their review and understanding to the QAC. This documentation will be combined with the résumés required to be on file for all individuals under the SCALE quality assurance plan to form the basis for qualification.

An inexperienced code user can perform work for a VALID case under the direction of a qualified originator. The qualified originator is responsible for the contents of the VALID case. Successful completion of a VALID case under supervision may also form the basis for qualification for future cases.

6.2 Reviewer Qualifications

Typically, a reviewer will be an experienced user of the computer code or codes to be used in the VALID case and will also understand the methods, strengths, and potential weaknesses of the techniques involved. The reviewer will have reviewed this procedure and provided documentation of their review and understanding to the QAC. In most cases, this documentation will already be on file to support originator qualification. Preferably, in most cases an individual will serve as an originator prior to being a reviewer, though this is not a requirement.

6.3 Expiration of Qualifications

The QAC should review the list of qualified originators and reviewers periodically to ensure that personnel who should be removed from the qualified lists are appropriately identified and removed. There are neither specific requirements to maintain qualification nor events, such as retirement and job changes, that necessitate termination of qualification. There is no set time limit for qualification expiration; each originator/reviewer is assigned by the QAC for each case, so qualification for each case is assessed on a case-by-case basis.

7.0 PROCEDURE

7.1 Initial Addition of Inputs and Data to the Library

If the models or data are already represented in the library, then revisions and corrections are handled according to Section 7.4 of this procedure.

Step 0 – Proposed

7.1.1 The requestor identifies models and derived data to be added to the library and opens a new issue in the VALID project within GitLab. The requestor provides a

brief description of the case(s) in the **Description** section of the issue template. If the requestor cannot be the originator, then the requestor should suggest an appropriate originator in the **Comments** section.

Step 1 – Approved/Rejected

- 7.1.2** The QAC approves or rejects the request. If the issue is approved, then the QAC adds the originator in the **Comments** section. Any qualified individual, including the requestor or the QAC, may be the originator.
- 7.1.3** The QAC assigns a VCL number to the case through the GitLab bot, which docket the case and details the VCL number in the **Comments** section. The docket number begins with *VCL* and is assigned a sequence number in the form *VCL-YYYY-NNN*. *YYYY* is the current fiscal year, and *NNN* represents a sequence of positive integers that begins with 001 and increases by 1 for each VCL identifier assigned. The complete numbering sequence restarts at the beginning of each fiscal year.
- 7.1.4** The QAC opens a “create branch (Issue *XXX-VCL-YYYY-NNN*) from master” or associates an existing branch in the repository with the issue and edits the issue field to include the appropriate numbers. In this case, *XXX*, is the VALID project GitLab issue number.
- 7.1.5** The QAC may then assign the issue an originator. If an originator is assigned, then the QAC adds the label “in progress” to the case. If no originator is available, then the QAC adds the label “approved” to the case.
- 7.1.6** If the issue is rejected, then the QAC notifies the requestor that the issue has been rejected, documents the notification in the **Comments** section, and closes the issue. Example reasons for rejection include,
 - (a)** the requested models or data are not appropriate for library inclusion,
 - (b)** supporting references are inadequate, or
 - (c)** work cannot commence because of funding or work priority issues.
- 7.1.7** Models or derived data may be generated for proprietary data. Proprietary information may be archived with appropriate access restrictions but should not be included directly in the GitLab issue or the attachments. This entails creating separate submodules in the GitLab VALID repository in which to place the proprietary data with the specified access controls. The QAC creates the submodule in the repository and applies directory access permissions as directed by the responsible ORNL project manager. The SCALE Director and responsible ORNL project manager determine an appropriate storage and access strategy on a project-by-project basis.

Step 2 – In Progress

- 7.1.8** The originator prepares, performs, and checks the necessary calculations for the issue.
- 7.1.9** The originator completes documentation of the model(s) and supplies the necessary calculations, including direct perturbation (DP) spreadsheets, calculated/expected (C/E) values, and so forth by using a web page markdown table and/or by attaching the documents directly to the issue. The originator also includes a notification in the **Comments** section. The originator includes a summary of the model and a brief description of key parameters in the GitLab issue in the appropriate sections. The originator also documents any revision number or other unique identifier(s) associated with the reference(s) used in model creation.
- 7.1.10** If the originator notes potential discrepancies in the source reference descriptions, such as a potential issue with Section 3 of an International Criticality Safety Benchmark Evaluation Project (ICSBEP) Handbook report [1], then the originator edits the field to **yes**. The originator documents the issue(s) in the **Comments** section. The QAC notifies the appropriate personnel responsible for further action and documents the notification in the **Comments** section.
- 7.1.11** The originator commits the files to the branch and opens a merge request (!Merge request number). This indicates in the issue that origination is complete. The originator assigns the GitLab issue to the QAC.
- 7.1.12** The originator removes the “in progress” label and adds the “in staging” label in GitLab.

Step 3 – In Staging

- 7.1.13** The QAC determines the necessary level of review using a graded approach. In determining the level of review, the QAC may consider the model source, the quality of the model reference, the rigor of other reviews already performed, and the expected use of the models and data.
- 7.1.14** If guidance is needed regarding the type of review required, then the QAC provides the necessary information in the **Comments** section and edits the field to **yes**. If no guidance is needed regarding the type of review, then the QAC edits the field to **no**. For example, rerunning inputs already in the library using a new version of SCALE or a new data library could be a **yes** scenario because less than the full level of review may be justified.

7.1.15 The QAC assigns the issue to the reviewer, removing the “in staging” label and adding the “in review” label. If no reviewer is available, then the issue retains the “in staging” label.

Step 4 – In Review

7.1.16 The reviewer reviews the models, derived data, and other documentation and provides comments in the **Comments** section (either in the **Issue** or **Merge Request [commit message – addressing feedback]** section). The originator performs necessary rework and/or addresses reviewer comments and provides comments in the **Comments** section (the **Issue** or **Merge Request** section, whichever is consistent with the option selected by the reviewer). The reviewer indicates in the **Comments** section (**Issue** or **Merge Request** section) that the comments have been addressed.

7.1.17 If difficulties are encountered during comment resolution by the reviewer, then the QAC is notified. If significant work is needed to resolve the comments, then the QAC assigns the case to the originator, and the case is returned to Step 2 – In Progress of this procedure. The QAC documents these actions in the **Comments** section. Moderate amounts of work to address reviewer comments can be performed without impact to the GitLab issue status.

7.1.18 The reviewer indicates that all comments have been resolved and the review is complete in the **Comments** section and assigns the case to the QAC. The reviewer then removes the “in review” label, and the “final check” label is added.

Step 5 – Final Check and Merge

7.1.19 The QAC reviews the issue to ensure that all necessary actions have been performed and documented. If no deficiencies are identified, then the QAC documents that the issue is complete and approved in the **Comments** section. The QAC removes the “final check” label and adds the “notification” label.

7.1.20 The QAC merges the branch onto the master branch in the repository.

Step 6 – Notification and Closure

7.1.21 The QAC mirrors the updated repository on code-int to code and emails the notification list of changes to the repository.

7.1.22 For the addition of revised models or data to the VALID library, as addressed by Section 7.4 of this procedure, the configuration control list is updated and the notification email will identify that the models or data are replacements for files previously removed from the library. The configuration control list will identify the revision numbers for the files.

7.2 Use of Inputs and Data

7.2.1 Users of inputs and derived data obtained from the VALID library ensure that quality assurance requirements are satisfied for the analyses for which they use the data.

7.2.2 Users report any errors or deficiencies in the data to the QAC.

7.3 Configuration Management of Inputs and Data

7.3.1 File access permissions are set so that only the QAC may modify or delete folders and files in the library.

7.3.2 The QAC ensures that the configuration control list is backed up on a schedule consistent with the configuration of the hardware where the archive is maintained.

7.4 Revisions or Corrections of Library Content

Revisions of inputs and data in the library may be necessary when references are revised, model improvements are identified, modeling approximations are determined to be inappropriate, modeling errors are discovered, methodology errors are discovered, or nuclear data errors are identified. Revisions are conducted in accordance with this procedure.

Step 0 – Identification and Isolation

7.4.1 The requestor identifies potential errors in inputs or derived data in the library and opens a new issue in the VALID project within GitLab. The requestor provides a brief description of the case(s) and reports the specific error(s) in the **Description** section of the issue template.

7.4.2 The QAC identifies and isolates by quarantining the inputs or data from the library on the appropriate branch on the master or by password-protecting the inputs or data so that users cannot access the potentially flawed models or data. This may entail moving a file(s) or directory to *file.x.error* or *directory.error* in the same directory on the branch. The QAC adds an additional description in the **Description** section of the issue template. This includes links to the original issue and branch location if the issue is under the current revision. If the issue is under a previous revision, then the issue does not link but rather references the previous documentation using the VCL or VALID form number.

7.4.3 The QAC sends out a notification of file restrictions and a description of the error to VALID users and documents the action in the **Comments** section.

7.4.4 The QAC assigns the issue to an individual, regardless of qualification, and documents the notification in the **Comments** section.

Step 1 – In Review

- 7.4.5** The staff member reviews the potential error or proposed revision and recommends “no revision needed,” “revision recommended,” or “removal of inputs and/or data from library recommended” by editing the level-one heading and documenting the supporting reasons in the **Comments** section.
- 7.4.6** If no revision is needed, then the QAC removes access restrictions or returns the inputs or derived data by moving the file(s) or directory to the original location and documents the release in the **Comments** section. The QAC sends out a notification to VALID users that the files have been unrestricted and documents the action in the **Comments** section. The QAC notifies the requestor that the issue has been resolved, documents the notification in the **Comments** section, and closes the issue.
- 7.4.7** If a revision is recommended, then the QAC opens a new issue, labels the issue as a revision, and links the revision issue in the **Comments** sections of both the defect and revision issues (GitLab issue numbers). The QAC assigns the revision issue to an originator and documents the notification in the **Comments** section. The originator proceeds with the revision as described in Section 7.1 of this procedure. The QAC changes the label of the issue reporting the defect to “in testing” and documents the revision action in the **Comments** section.

Step 2 – Revision

- 7.4.8** The new revision is processed as described in Section 7.1 of this procedure.
- 7.4.9** The QAC indicates in the **Comments** sections of both the defect and revision issues that the replacement files have been added to the library on the appropriate branch on the master (Issue *NNN-VCL-YYYY-XXX*).

Step 3 – Finalization/Removal

- 7.4.10** If the inputs or data should be permanently removed from the library, then the QAC archives the isolated or restricted files and indicates the action in the **Comments** section. The QAC sends out a notification of the permanent removal of the files to VALID users and documents the action in the **Comments** section. The QAC indicates in the **Comments** sections of both the defect and revision issues that the issue is resolved and closes both issues.

8.0 REFERENCES

- [1] *International Handbook of Evaluated Criticality Safety Benchmark Experiments*, NEA/NSC/DOC (95)03, Organization for Economic Co-operation and Development, Nuclear Energy Agency, Paris, France (2022).