

**Configuration Management Plan
for the SCALE Code System**

Prepared by
S. M. Bowman

Nuclear Analysis Methods and Applications
Nuclear Science and Technology Division
Oak Ridge National Laboratory

Date Revised: September 19, 2002

Approvals:

SCALE Project Leader

Date

**Group Leader,
Nuclear Analysis Methods and Applications**

Date

Quality Program Manager, NSTD

Date

INTRODUCTION

The SCALE (Standardized Computer Analyses for Licensing Evaluation) computational system is a modular system consisting of a driver module, control modules, functional modules, utility modules, data libraries, and subroutine libraries. The SCALE system draws heavily from basic neutron-transport, data-processing, and heat-transfer methods technology developed at Oak Ridge National Laboratory (ORNL) over the past three decades. It is designed for evaluating the criticality, shielding, and thermal aspects of nuclear fuel facility and/or spent fuel cask designs. The overall goal of the SCALE project has been to develop easy-to-use analytical sequences that are automated to perform the necessary data processing and analysis using well-established computer codes and data libraries.

The driver module remains active at all times and transfers the control and functional modules to and from the central processor unit. Each analytic sequence is driven by a control module. The control module automates the necessary data processing, generates the input to the functional modules, initiates module execution in proper order, and performs any needed post-processing. The functional modules are the analytical codes which provide the specific capabilities required to solve the problem of interest. They may be executed as part of an analytical sequence when called by a control module, or in stand-alone mode.

Utility modules allow users to combine problem-dependent cross-section libraries, convert their non-SCALE cross-section libraries to obtain libraries with the format and nuclide identifiers required by a SCALE module, or perform other edit, data checking, or post-processing functions.

The subroutine libraries contain subroutines which are accessed by multiple modules in SCALE. Whenever a change is made to one of these subroutines, every module that accesses that subroutine must be re-linked with the modified subroutine. The source code for each of the impacted modules remains the same, but the executable code is changed.

The data libraries are validated data files which contain cross sections, material compositions, or physical properties. The data can be easily accessed via keyword input to a control or functional module.

Since SCALE is a modular code system with interrelated modules and libraries, configuration management is a complex and difficult task. This configuration management plan seeks to implement a procedure for ensuring the integrity, reliability, and maintainability of the baselined SCALE code system maintained at ORNL without discouraging continual improvements. This plan was implemented with the publicly released SCALE-4.0 version serving as the initial baseline configuration. By definition, the baseline configuration is considered to be validated and verified based on years of experience using the system for routine applications, analysis of measurements and benchmarks, and code comparison studies. In addition, the baseline configuration was verified by analysis of the sample problems prepared by the code managers. Changes to the system (computer code and documentation) will be controlled by this configuration management plan under the direction of the SCALE Project Leader.

1.0 PURPOSE

To describe the methods used for recommending, approving, verifying, implementing, documenting, and controlling changes to the SCALE computational system which is controlled by the Nuclear Analysis Methods and Applications (NAMA) Group in the Nuclear Science and Technology Division (NSTD).

2.0 SCOPE

This plan applies to the modification of existing software or the development of new software in the baselined SCALE computational system controlled by NAMA only on the designated ORNL computer systems.

3.0 REFERENCES

- 3.1 "Quality Assurance Plan for the SCALE Computational System," QAP-005,R1.
- 3.2 "SCALE Code Manager Procedure for Module Revision Reports," SCALE-CMP-002.
- 3.3 "SCALE Code Manager Procedure for Data Revision Reports," SCALE-CMP-003.
- 3.4 "SCALE Procedure for SCALE Discrepancy Reports," SCALE-CMP-004.
- 3.5 "SCALE Software Coordinator Procedure for Module Revisions on UNIX Workstations," SCALE-CMP-007.
- 3.6 "SCALE Software Coordinator Procedure for Data Revisions on UNIX Workstations," SCALE-CMP-008.

4.0 REQUIREMENTS

Documentation and verification are required for all computer codes or other software modified or developed in the SCALE computational system. A validation report is required for major modifications (as determined by the Project Leader) or new software (except utility modules) that are added to SCALE. The Project Leader also has the option, at his discretion, to require a validation report for other modifications.

5.0 DEFINITIONS

- 5.1 **Computer Code** - A set of instructions that can be interpreted and acted upon by a programmable digital computer (also referred to as a module or a computer program).
- 5.2 **Source Code** - A computer code in its originally coded form, typically in text file format. For programs written in a compilable programming language, the uncompiled program.

- 5.3 **Object Code** - A computer code in its compiled form. This applies only to programs written in a compilable programming language.
- 5.4 **Executable Code** - The user form of a computer code. For programs written in a compilable programming language, the compiled and loaded program. For programs written in an interpretable programming language, the source code.
- 5.5 **Data Library** - A data file for use with an executable code that is created and maintained by the controlling organization and is not intended for modification by the user.
- 5.6 **Sample Input** - Input data for a designated sample problem which is maintained by the controlling organization for distribution to users.
- 5.7 **Test Case Input** - Input data for a test case used to verify a modification to a module or a data library.
- 5.8 **Software** - Computer codes and data in electronic file format.
- 5.9 **Design Requirements** - Description of the methodology, assumptions, functional requirements, and technical requirements for a software system.
- 5.10 **Discrepancy** - The failure of software to perform according to its documentation.
- 5.11 **Validation** - Assurance that a model as embodied in a computer code is a correct representation of the process or system for which it is intended. This is usually accomplished by comparing code results to either physical data or a validated code designed to perform the same type of analysis.
- 5.12 **Verification** - Assurance that a computer code correctly performs the operations specified in a numerical model or the options specified in the user input. This is usually accomplished by comparing code results to a hand calculation or an analytical solution or approximation.
- 5.13 **Version Number** - A number of the form X.Y.Z used to identify the version of a computer code. A version number is assigned when the computer code is implemented for production use. For modifications between baselined versions of SCALE, the last field is increased by one. When a new SCALE baseline is established, the X.Y fields are increased to the new version number of the entire SCALE computational system.

6.0 RESPONSIBILITIES

- 6.1 **Project Leader** - The person responsible for managing the maintenance and development of the SCALE computational system. The Project Leader acts as the Configuration Control Board (CCB) and is responsible for approval of all changes to the baseline. As such, the Project Leader is responsible for the configuration management and quality assurance of the SCALE software system. The NAMA Group Leader may sign for the Project Leader in his absence.

- 6.2 **Code Manager** - The staff member responsible for the development, maintenance, verification, and documentation of changes to a module or a data library.
- 6.3 **Software Coordinator** - The staff member responsible for the implementation, protection, and maintenance of the SCALE software system.

7.0 PROCEDURE

7.1 Configuration Identification and Integrity

- 7.1.1 Each computer code is identified by a unique name and version. These are displayed in the code output in order that users may verify that they are using the correct code version.
- 7.1.2 The SCALE Software Configuration Control Lists (CCLs) identify the system baseline and are maintained by the Software Coordinator for each computer system designated by the Project Leader. Copies of the CCL's are kept in a log book and updated by the Software Coordinator whenever a change is made. The CCL's are signed and filed by the Software Coordinator and Project Manager on a quarterly basis. They include all production source modules, object modules, executable modules, sample input files, and data libraries.

The source modules, object modules, executable modules, sample input files, and data libraries are maintained on each computer system designated by the Project Leader. The Software Coordinator maintains documentation of the version, revision date, and location of each executable module. The revision date and location are documented for each source module, object module, sample input file, and data library. The Software Coordinator also maintains a log of all completed and outstanding Module Revision Reports, Data Revision Reports, Discrepancy Reports, Code Design Requirements, and Technical Review Forms to monitor the status of configuration changes.

- 7.1.3 All source codes, executable codes, sample input files, and data libraries are protected by the Software Coordinator to ensure their integrity from unauthorized modification. The Software Coordinator is responsible for appropriate backups of these files.
- 7.1.4 Stand-alone computer codes supplemental to the SCALE system but not normally included therein may be included and controlled by this procedure if so designated by the Project Leader.

7.2 Configuration Change Control

- 7.2.1 Users or Code Managers may request changes to the SCALE software configuration by notifying the Project Leader in the appropriate written form as described below.
- 7.2.2 When change requests are received, the Code Manager completes Part I of the Module Revision Report (MRR) or Data Revision Report (DRR). The report is

assigned a report number and docketed by the Software Coordinator. (Reference 3.5 or 3.6, Paragraph 3.1)

- 7.2.3 If the code changes are significant or a new code is being developed, the Code Design Requirements (CDR) form must be completed by the Code Manager, approved by the Project Leader, and attached to the revision report. Otherwise, the specifications in Part I of the MRR serve as the code design requirements. These requirements may be modified at any time during the development process with the approval of the Project Leader by modifying the appropriate form(s).
- 7.2.4 The Project Leader reviews Part I of the revision report and approves or disapproves the request by completing Part II of the report. If the request is approved, the Project Leader assigns a Code Manager for the report. The Project Leader assigns another individual as an independent Technical Reviewer who will review the software changes and documentation upon completion. The Project Leader may not act as Code Manager but may act as Technical Reviewer. If the request is disapproved, the Project Leader provides an explanation for disapproval, forwards the original revision report to the Software Coordinator, and sends a copy to the requestor.
- 7.2.5 The Code Manager obtains the controlled module(s) or data file(s) from the Software Coordinator; performs the changes; and tests the revised software (Reference 3.2, Paragraphs 3.3 through 3.7 or Reference 3.3, Paragraphs 3.3 through 3.7). The revised software must be stored in a development area to prevent accidental use by production software users until final approval by the Project Leader for implementation.
- 7.2.6 The Code Manager uses the sample inputs obtained from the Software Coordinator, if possible. The existing sample inputs may require modification or the Code Manager may have to develop new test inputs to illustrate the effects of the changes. Testing should include any other modules that may be affected by the changes. If existing sample inputs are used to test the revised software, the Code Manager compares the results and documents the explanation of any differences. If revised sample inputs or new test inputs are used, the Code Manager documents in the report how the changes affected the output. The Code Manager is responsible for:
 - a) the technical validity and completeness per the code design requirements
 - b) the adequacy of the verification and/or validation
 - c) the adequacy of the documentation and its consistency with the existing system documentation
 - d) the adequacy and the completeness of the sample problems for inclusion in the publicly released software version.
- 7.2.7 The Code Manager completes Part III of the revision report (marking "N/A" in portions that are not applicable) and attaches a report containing documentation of changes, testing, revised sample and new test case inputs, and results of the verification and/or validation (Reference 3.2, Paragraphs 3.7 and 3.8 or Reference 3.3, Paragraphs 3.7 and 3.8). Verification is required for all software modifications and must include a minimum of one test problem. If there are major modifications or a new code, all sample problems for a module must be

executed and validation is required. For a new data library, validation is required. The appropriate level of verification / validation is determined by the Project Leader based on the complexity of the modifications. The Code Manager submits the revision report and all attachments to the Project Leader. The revision report number should be included on all attachments.

- 7.2.8 The technical reviewer(s) reviews all documents to determine:
- a) the technical validity and completeness per the code design requirements
 - b) the adequacy of the verification and/or validation
 - c) the adequacy of the documentation and its consistency with the existing system documentation
 - d) the adequacy and the completeness of the sample problems for inclusion in the publicly released software version.
- 7.2.9 The reviewer completes Part IV of the revision report (Reference 3.2, Paragraph 3.10 or Reference 3.3, Paragraph 3.10) and returns all documents to the Project Leader. A Technical Review Form (TRF) must be attached to the revision report for major modifications or validation, or if the Technical Reviewer has any findings. Any findings or recommendations of the reviewer(s) are documented on the TRF, if applicable. Further technical review is required only if the recommendations are deemed major by the Project Leader or if the code design requirements are not met. The Project Leader is responsible for ensuring that the recommendations are addressed satisfactorily.
- 7.2.10 After all findings and recommendations have been resolved, the Project Leader approves the changes by completing Part V of the revision report. The Project Leader also specifies which test problems shall be run by the Software Coordinator after a new production module is installed. A minimum of one test problem must be run on each computer system to verify a new production module after installation.
- 7.2.11 The Project Leader notifies the Code Manager that the changes have been approved. The Code Manager transmits all files containing updated source code, data libraries, and test/sample inputs, and the revision report with attachments to the Software Coordinator.
- 7.2.12 The Software Coordinator updates the affected source modules, object modules, executable modules, data libraries, and sample input files on each designated computer system. (Reference 3.5, Paragraphs 3.3 through 3.5 or Reference 3.6, Paragraph 3.2).
- 7.2.13 The Software Coordinator runs all test problems specified in Part V of the revision report and verifies that the results agree with those obtained previously by the Code Manager. If the results do not agree, the Software Coordinator notifies the Code Manager or Project Leader and appropriate action is taken to rectify the situation (Reference 3.5, Paragraph 3.6 or Reference 3.6, Paragraph 3.3).
- 7.2.14 The Software Coordinator completes Part VI of the revision report (marking "N/A" in portions that are not applicable). The Software Coordinator updates the

SCALE Software Configuration Control Lists log book and files the revision report with attachments (Reference 3.5, Paragraph 3.7 or Reference 3.6, Paragraph 3.4).

- 7.2.15 The Project Leader ensures that the Radiation Safety Information Computational Center (RSICC), funding sponsors, and local users are notified of the update via the SCALE Newsletter and that any document changes are included in the next documentation release. The Software Coordinator updates the electronic bulletin board on each computer system with a notification message specified by the Project Leader.
- 7.2.16 When a new version of the entire SCALE system is configured on a computer system, the new baseline is documented by the Software Coordinator using a Baseline Revision Report (BRR). Lists of all MRR numbers and DRR numbers since the previous baseline are attached to the report. Copies of all Configuration Control Lists for the new baseline are also attached. All sample problems for all modules are executed and the output files are reviewed by the Software Coordinator or a designated reviewer. If significant changes have been made to a module, the Code Manager must review the output. Verification of the sample problem output is documented using the Sample Problem Verification Form. Copies of these forms are attached on the report and the output files are included in the new baseline.

7.3 Discrepancy Reports and Corrections

- 7.3.1 Problems encountered with SCALE are reported to the Code Manager or the Project Leader.
- 7.3.2 Upon confirmation of a discrepancy, the Code Manager or SCALE user completes Part I of the Software Discrepancy Report (SDR). The report is assigned a report number and docketed by the Software Coordinator (Reference 3.4, Paragraph 4.1).
- 7.3.3 The Code Manager evaluates the impact and significance of the discrepancy and completes Part II of the form (Reference 3.4, Paragraph 4.2).
- 7.3.4 The Project Leader grants or denies approval for corrective action in Part III of the form (Reference 3.4, Paragraph 4.3). If approval is granted, the Code Manager performs the corrective action according to step 7.3.5.
- 7.3.5 The Code Manager documents the resolution of the discrepancy by completing Part IV of the form. If the corrective action requires any changes to the SCALE software configuration, the changes are made according to Section 7.2 of this procedure. A copy of the Discrepancy Report must be attached to the Module or Data Revision Report and the original is given to the Software Coordinator to be filed. If the corrective action only requires changes to the SCALE documentation, a copy of the revised document pages is attached to the SDR (Reference 3.4, Paragraph 4.4).
- 7.3.6 The Project Leader ensures that RSICC, funding sponsors, and local users are notified of the corrective action and its impact on users via the SCALE

Newsletter. In addition, steps are taken to ensure that any document changes (with revision dates on new/revised pages and sections) are included in the next documentation release.

7.3.7 The SDR is filed by the Software Coordinator.

7.4 Configuration Assessments

7.4.1 A configuration assessment of the SCALE code system is performed at least once each year by a cognizant NAMA staff member designated by the Project Leader. The purpose of the assessment is to verify compliance with this configuration management plan.

7.4.2 The results of the assessment are reported in writing to the Project Leader.

7.4.3 The Project Leader ensures that any necessary corrective actions resulting from the assessment are completed and documented in accordance with the SCALE QA Plan (Ref. 3.1).

7.5 Status Accounting

7.5.1 The Software Coordinator maintains a log of all completed and outstanding Module Revision Reports, Data Revision Reports, Software Discrepancy Reports, Code Design Requirements, and Technical Review Forms to monitor the status of configuration changes.

7.5.2 The Software Coordinator provides the Project Leader with a monthly report to account for the status of all changes made to the SCALE code system during that time period. This report also lists all changes that are currently pending to SCALE modules.

7.5.3 The Software Coordinator compiles and transmits a report to RSICC on a quarterly basis to account for all changes to the SCALE code system during that time period.