

**SCALE SQA Coordinator Procedure  
for Module Revisions on Unix and Linux Workstations**

Prepared by


S. Y. Walker  
S. M. Bowman

Nuclear Systems Analysis, Design, and Safety (NSADS)  
Nuclear Science and Technology Division (NSTD)  
Oak Ridge National Laboratory

Date Revised: March 30, 2007

Approvals:

  
\_\_\_\_\_  
**SCALE Project Leader** 3/30/07  
\_\_\_\_\_  
**Date**

  
\_\_\_\_\_  
**Quality Program Manager, NSTD** 3/30/07  
\_\_\_\_\_  
**Date**

# SCALE Software QA Coordinator Procedure for Module Revisions on Unix and Linux Workstations

## 1.0 PURPOSE

To describe a standardized operational procedure to be followed for implementation of SCALE module revisions on Unix and Linux workstations where SCALE is configuration controlled. The procedure outlined in this document complies with the Configuration Management Plan (CMP) for the SCALE code system that is controlled by the NSTD Nuclear Systems Analysis, Design, and Safety (NSADS) Group.

## 2.0 SCOPE

This procedure covers the modifying and testing of SCALE executable modules on Unix and Linux workstations.

## 3.0 PROCEDURE

### 3.1 Generation of MRR Form

- 3.1.1 A SCALE Code Manager or user should request an MRR form from the SCALE Software Quality Assurance (SQA) Coordinator when modifications need to be made to a SCALE module/library.
- 3.1.2 The SQA Coordinator completes the following items in Part 1 of the MRR form:
  - 3.1.2.1 **MRR Report No.** – Assign a sequence number “YY-NNN” to the MRR. “YY” represents the last two digits of the current year and “NNN” represents a three digit sequence number beginning as “001” and increasing by one each time each time a new MRR number is assigned. This complete sequence of numbering restarts at the beginning of each calendar year.
  - 3.1.2.2 **Module/Library Name** – Enter the name of the executable module (e.g., KENO-VI) or the library name (e.g., Material Information Processor Library).
  - 3.1.2.3 **Version** – Identify the current version of the executable module requiring modification and enter it on the MRR form. The current version number is located in a text file named “qatable” in SCALE data directory. The version is identified by a number format of “X.Y.Z” (e.g., version number 5.0.1), where “X.Y” represents the version number of the entire SCALE computational system and “Z” represents the module revision number (i.e., the number of

times the module has been revised since the baseline of SCALE version X.Y). Note that version numbers do not apply for libraries. Enter “N/A” in the space provided when modifying libraries.

- 3.1.3 Complete entries for module/library name, version, and request date on the MRR log.
- 3.1.4 Give the MRR form to the Code Manager.
- 3.1.5 The Code Manager must complete Part I of the MRR form and obtain the Project Leader’s approval in Part II.
- 3.1.6 The Code Manager must complete Part III and the Technical Reviewer must complete Part IV of the MRR form. The signature of the Project Leader in Part V is required before the SQA Coordinator may incorporate the software revisions as specified in the remainder of this procedure.

### **3.2 Adding a New Module to the Revision Control System (RCS)**

Step 3.2 applies to the addition of a new module to the SCALE RCS. If an existing module in the RCS is being modified, skip to Step 3.3.

- 3.2.1 Create a new directory using the new module name (e.g., windex) in the src directory. Invoke the “mkdir” command to make the new module directory and change directory to the new directory, e.g.:

```
cd /scale/scale5/src/  
mkdir -p windex/RCS  
cd windex
```

- 3.2.2 Create the “Author” and “desc.file” text files. The Author file contains the name of the code manager for the module and desc.file contains a brief description of the module.

- 3.2.3 Copy the “Makefile.patn” file located in the /scale/scale5/src directory to the file “Makefile” in the new module directory, e.g.:

```
cp ../Makefile.patn Makefile
```

Edit the Makefile to add the new program module name (e.g., windex). Delete extra lines of text not needed for running the Makefile for the new module.

- 3.2.4 Execute the ”qacomment.pl” script located in the /scale/scale5/cmds directory, as shown in the following example

```
/scale/scale5/src/windex> qacomment.pl *.f90
```

Note that \*.f90 can be replaced with \*.f or individual source file names (e.g., kenovx.f90). The qacomment.pl script will create a backup copy of each source file with a extension of “.old” (e.g., kenovx.f90.old).

- 3.2.6 Place source files (e.g., kenovx.f90) in the RCS directory by using the initial check-in script “intci” located in /scale/scale5/cmds directory, e.g.:

```
/scale/scale5/src/windex> intci *.f90
```

Note that all checked-in RCS source files are given “.v” extensions (e.g., kenovx.f90.v) and the corresponding source files become “working” files (e.g., kenovx.f90). The intci script prompts user to enter a description for the file, a tag will remain with it for the lifetime during which it is under RCS control. The description should be a synopsis of the contents of the source file. Terminate the description by typing either a period “.” or a Ctrl-D. The initial check-in is assigned the initial revision number 1.1.

- 3.2.7 Use the initial checkout script “intco” to retrieve the files from the RCS file to be compiled, e.g.:

```
/scale/scale5/src/windex> intco
```

Remove all backup files with “.old” extensions that were created in step 3.2.5, e.g.:

```
/scale/scale5/src/windex> rm *.old
```

- 3.2.8 Execute “mksrc5” script located in /scale/scale5/cmds directory, e.g.:

```
/scale/scale5/src/windex> mksrc5> sources
```

where “sources” is a file created by mksrc5 that contains a list of the new source files.

- 3.2.9 Make a new “.depend” file by running the “makedepf90” program located in the local/bin directory, e.g.:

```
makedepf90 *.f90 ../DLD/*.f90 ../scalelib/*.f90> .depend
```

where “DLD” represents the Dependency Library Directory of any other library that the module depends upon. The .depend file then needs to be edited to remove the entries for files in the other directories.

- 3.2.10 Edit file “qatable” to insert the new module name (e.g., windex). The “qatable” is located in /scale/scale5/data directory.

3.2.11 The new module name (e.g., windex) will need to be added to the SCALE head “Makefile” located in /scale/scale5. Invoke the following command line, e.g.:

```
make SUBDIRS=windex configure
```

3.2.12 Add the new module alphabetically to the appropriate directory list in the head SCALE directory (e.g., FDIRS, CDIRS, etc). as specified by the Code Manager.

3.2.13 Go to procedure Step 3.4, “Compiling Source Files.”

### **3.3 Updating Source Files in the Revision Control System (RCS)**

3.3.1 If the module or library is already in the SCALE Revision Control System (RCS), use the RCS check out command “co” to retrieve the latest revision from the module RCS subdirectory (e.g., src/windex/RCS for a module named “windex”) and write it into working files in the module subdirectory (e.g., src/windex) as seen in the example below.

```
/scale/scale5/src/windex> co -l *.f90
```

where the “-l” flag is only used to lock the file when the file is being checked out for modification. (Note that individual source files may be specified instead of \*.f90.)

3.3.2 Copy source files from their locations specified in Part III of the MRR form to the /scale/scale5/src source module directory.

3.3.3 Place revised source files (e.g., kenovx.f90) under configuration control in the RCS directory by using the “checkin” script:

```
/scale/scale5/src/windex> checkin kenovx.f90
```

Note that all checked-in RCS files are given “.v” extensions (e.g., kenovx.f90.v) and the corresponding source files become “working” files (e.g., kenovx.f90). The checkin script prompts the user to enter a log message. The log message should explain the reason for the modified revision. Terminate the log message with a line containing a period “.” or Ctrl-D.

The checkin script also retrieves the updated RCS files (e.g., /scale/scale5/src/windex/RCS/kenovx.f90.v) and writes source files into the “working” file directory (e.g., /scale/scale5/src/windex) to be compiled.

- 3.3.4 If new source files are to be added or old files are to be deleted, run the “mksrc5” script, e.g.:

```
/scale/scale5/src/windex> mksrc5> sources
```

where “sources” is a file created by mksrc5 that contains a list of all source files for the module.

- 3.3.5 If dependencies have changed, make a new “.depend” file by running the “makedepf90” program located in the local/bin directory, e.g.:

```
makedepf90 *.f90 ../DLD/*.f90 ../scalelib/*.f90> .depend
```

where “DLD” represents the Dependency Library Directory of any other library that the module depends upon. The .depend file then needs to be edited to remove the entries for files in the other directories.

### **3.4 Compiling Source Files**

- 3.4.1 Execute the “updscale” script from the SCALE head directory (e.g., /scale/scale5). This script uses the “make” utility to update all affected libraries and modules. The script also writes a complete log file (e.g., log\_make.yyyy\_mm\_dd\_hh\_mm.machine-name) of all actions performed. Note that make uses the date/time stamp of the files to determine which files need to be updated. For each of those files, make issues the commands in the Makefile and compiles the files in each of the source module directories, producing object files in a machine-specific build directory that are used to build a relocatable object library. Once the library has been updated, a new executable is created. The updscale script also executes the “make install” script, creating a new executable module.
- 3.4.2 Verify the revision dates and version numbers for the updated modules in the “qatable” file in the “data” subdirectory on each computing platform. Print copies of the qatable files, highlight the updated modules, and attach to the MRR.
- 3.4.3 Print the objects and source for each computing platform. Highlight the updated files and attach to the MRR.
- 3.4.4 Enter the production implementation date information in Part VI of the MRR for source, object, and executables that were updated.

### **3.5 Executing the Test Problem(s)**

- 3.5.1 Copy all test problems specified in Part V of the MRR form to the test directory. The naming convention for test problems is mrrYY-NNN.inp, where YY-NNN is the MRR sequence number.
- 3.5.2 Run all test problems.
- 3.5.3 Use a text editor to examine the output and message files to verify successful execution of the test cases. The results should be compared with those obtained previously by the Code Manager.
- 3.5.4 If the results do not agree, notify the Code Manager. Appropriate action to rectify the problem is taken under the direction of the Code Manager or Project Leader.
- 3.5.5 If the results agree, enter comments such as “Results of testing were as expected” in Part VI of the MRR form.
- 3.5.6 Sign and date the “Performed by” and “Tested by” blocks in Part VI of the MRR form.

### **3.6 Completing Documentation**

- 3.6.1 Copy the QA Comments from the MRR documentation into an email notification message to ORNL SCALE users. The email notification should be addressed to [scale@home.ornl.gov](mailto:scale@home.ornl.gov). (Local users may request to be added to the email list by contacting the SQA Coordinator or the Project Leader.) Also enter the information into the summary of SCALE updates for local users on the internal ORNL website.
- 3.6.2 If document changes to the SCALE Manual are attached or pending, attach a copy to the MRR and send the original to the SCALE Manual reports specialist.
- 3.6.3 Update the MRR log, recording dates for the updated source files, object library, executable module and completion date for test problems.
- 3.6.4 File the original MRR form in the Module Revision Report binder and an electronic copy on the SQA Coordinator’s PC.
- 3.6.5 Enter the MRR report number into the Quality Assurance Log.

### **3.7 Quarterly Reports**

- 3.7.1 Generate quarterly Configuration Control Lists (CCLs) for program source, object modules, and executables with updated SCALE-CCL

version numbers.

- 3.7.2 Enter the new SCALE-CCL numbers into the Quality Assurance Log.
- 3.7.3 Sign and date the CCLs. Obtain the Project Leader's signature.
- 3.7.4 File the original CCLs in file cabinet labeled "Quality Assurance Records" under the appropriately labeled folder.
- 3.7.5 File copies of each CCL, Quality Assurance Log, and all MRR forms processed during the quarter at the duplicate file location.