README for SCALE 5 on Unix and Linux
RSICC code package C00725/MNYCP/00
June 2004

This document is the README file for the Unix/Linux version of SCALE 5, which is RSICC code package CCC-725. You can find additional information and technical support at the following websites.

http://www.ornl.gov/sci/scale (SCALE Web site)
http://rsicc.ornl.gov/ (RSICC Web site)
http://www.ornl.gov/sci/scale/scale_notebook.html (SCALE Electronic Notebook)

Table of Contents

1. SCALE Version 5 for Unix/Linux
2. Installation steps
3. User registration
4. SCALE on-line electronic manual
5. Directory structure
6. Installation scripts to build SCALE
7. Running and verifying sample problem results
8. How to execute SCALE
   - Table 1. SCALE 5 Logical Unit Master List
   - Table 2. SCALE 5 Big Endian Logical Unit Numbers for Unix/Linux
9. Qatable
10. Aliases

1. SCALE Version 5 for Unix/Linux

The Unix/Linux and Windows versions are functionally equivalent and contain all the modules, data libraries, and miscellaneous utilities in SCALE 5. This information is for the Unix/Linux version of the SCALE 5 modular code system. This distribution includes Fortran 95 and C source codes, data libraries, test cases, and Makefiles necessary to build and test the entire system. Fortran 95 and C compilers are required.

SCALE 5 was benchmarked on DEC Alpha workstations (Tru64 4.0F, 5.0, 5.1) using the Digital Fortran 95 compiler. A Makefile to build the system on IBM, DEC, SUN, HP, Linux, and Mac/OSX is provided to facilitate installation of the code system, which is distributed on two CDs in several gnu compressed tar files. Approximately 1.6 GB of disk space is required to build the system on 1 OS and execute the sample cases. Each additional OS needs about 0.5 GB, of which half is the output of the sample cases. SCALE 5 was tested only under the systems listed below and will probably require modification for any other system. Windows users should install the Windows version which is distributed by RSICC on two separate CDs. Complete instructions for Windows users are given in the Getting Started file on the SCALE 5 install disk 1 for Windows.

The GNU software is available free of charge as gzip-1.2.4.tar from the Free Software Foundation, 657 Massachusetts Ave., Cambridge, MA 02139 or via anonymous FTP from pub/gnu at prep.ai.mit.edu or http://www.gnu.org/directory/gnu/
.

Scale5 was tested on the following systems.

- DEC6600 running Tru64 UNIX V5.0A, Tru64 UNIX V5.1A with HP Fortran V5.5A and C compiler V6.1 and V6.4
- DEC6600, DEC4100, and other DEC's running Digital UNIX V4.0D, V4.0F, V4.0D with Compaq Fortran V5.5 and DEC C V5.9
- IBM RS/6000 580 running AIX 4.3 with XL Fortran Compiler 6.1.0.9 and C 4.3.0.1
- IBM RS/6000 590 running AIX 5.1 with XL Fortran 08.01.0000.0003 and C 5.0.0.0
- Sun SPARCstation-20 running Solaris 5.7 with FORTRAN 95 6.0 and C 5.1
- AMD Athlon running Red Hat Linux 7.3 with Lahey/Fujitsu Fortran 95 L6.10a and GNU gcc 2.96

Additional software updates will be posted on the SCALE website. Click on "Download" for additional utilities, verification/validation input files, and other updates. The SCALE electronic notebook on the web contains a great deal of helpful information including answers to many FAQs.

If you encounter problems, please read all this information and determine where the first problem occurred before contacting ORNL for help. Inquiries for technical assistance may be directed to the SCALE staff at scalehelp@ornl.gov. To aid in identifying the problem, try to include as much detail as possible, including input/output files and error messages. Be sure to specify your computer, operating system and compiler.

2. Installation Steps

The following commands create the SCALE 5 system under the current working directory, install the system, build libraries, and run sample problems. **Note that your current working directory '.' must be included in the user's path for the Make commands to work correctly.** One may choose any name for this directory, but changing names of any other files/directories will require changes to Makefiles and is **NOT** recommended. If you rename the directory, it should be done **before** Make is invoked to build the system.

WARNING: If space is limited on /usr/tmp, modify installscale.patn and cmds/batch5.patn to define **TMPDIR** on a disk with more free space (100 MB minimum, 0.5 to 1 GB recommended).

To install on Unix or Linux systems:

Create the head directory for the Scale5.0 system and *cd* into it.

(Change /**cdrom** to point to the CDrom and load the first Unix CD)
gunzip -c ***/cdrom****/BASE.TGZ | tar xf -* (uncompress & extract files under the present working directory)
gunzip -c ***/cdrom****/SRC.TGZ | tar xf -*
gunzip -c ***/cdrom****/BIN_XS.TGZ | tar xf -*
(unload the first CD and load the second Unix CD. This CD contains the CENTRM point data files)
gunzip -c ***/cdrom****/CEN5BIN.TGZ | tar xvf -*

make configure This step creates a directory with a system dependent name, set by the script **./cmds/sysname**, and generates a **config.data** file in this directory. If this name matches that of one of the files in the **config** directory, that file is used as the base for the **config.data** file, otherwise a dummy file is generated; and the user will need to create one by hand. For example, if you are running under AIX 5 in a head directory called ***SCALE5,*** the configuration file will be ***SCALE5****/AIX_5*/config.data.

Check that a valid **sysname/config.data** file was found for this system. If not, look at the files in the **config** directory, and create a **config.data** file in the system dependent directory by hand. Verify that the "**SCALE=**" definition in **config.data** is the directory path of the directory into which the tar files were

extracted. The directory must exist with this name on any system which wishes to run this SCALE 5 installation. Numerous subdirectories are created during the *make configure*. **If "SCALE=" or "TMPDIR=" is redefined, then redo the *make configure*, as this will fix the scripts.**

At this point, if only a partial installation is desired, edit the files **LDIRS, BDIRS, FDIRS, CDIRS, ADIRS,** and **UDIRS** in the head Scale directory to remove the undesired programs. **LDIRS** is the list of the subroutine libraries in Scale5.0. **scalelib** is required for all parts of scale, and **miplib** is required if any control program other than **htas1** is desired. **marslib** is needed for **morse**, **sas3**, and **sas4**. **senlib** is needed by the **tsunami** sensitivity sequences, while the last 3 libraries (gd-1.8.1, libpng-1.2.0, and zlib-1.1.4) are used in making 2-dimensional slices through **keno** and **morse** geometries. **BDIRS** defines the **shell** command, the scale **driver**, and the **compoz** program. COMPOZ creates and edits the **standard composition library**. Also here are **mal** and **wgt** which make the **keno** albedo and weight files. **FDIRS** is the list of Scale functional modules, and **CDIRS** is the list of Scale control modules. Check the documentation for descriptions of what these programs do. **ADIRS** contains the list of **AMPX** programs included with Scale5.0, and **UDIRS** is a list of utility programs. These are documented in sections **M15** and **M17**. If any programs are removed, remove the program names form the file **SMPL** in the head directory to make sure the sample problems for those programs will not be run.

installscale Running in background is recommended "*nohup installscale &*".

In the main directory, the **installscale** script invokes Make to process source files and build executables. (If this step aborts in TRITON, see note below on rename.c.)

makelibraries Invoke make to create standard composition, origen, and keno data libraries.

runsmpls Run sample problems defined in **SMPL** in the background "*nohup runsmpls &*" because this takes many hours. See Section 7 later in this file.

Each of these steps creates log files in **sysname**, the system dependent directory. Examine the logs to determine the success of the installation. All compilations are done under this directory, the library creations are run here, and the sample problems are run here. If the head directory is accessible on different systems using the same directory pathname, then Scale5.0 can be built and maintained on each system, as all the system dependent files are kept under the system dependent directory. The data libraries only need to be created once, as they are shared between the systems. All the data libraries are kept in "big endian" order. On "little endian" machines (Tru64, x86 Linux), the HP compiler for Tru64 and the Lahey compiler for Linux are able to read "big endian" ordered files using options specified in the config.data file. The scale5 script sets the necessary environmental variables for this to happen.

The TRITON module needs to rename files while it is running, and uses an intrinsic function **rename** to accomplish this. This intrinsic is an extension to the Fortran standard, and is not available on all machines. If the log file indicates that TRITON was not successfully built because **rename_** was an unsatisifed external reference, then perform the following steps starting from the head Scale5 directory.

cd **sysname**/triton	*where **sysname** is the system dependent directory name*
cc -c ../../work/rename.c	*where cc is the C compiler (may be xlc on AIX machines)*
ar rvs ../lib/libtriton.a rename.o	
cd ../../	*change to the head directory*
updatescale	*updatescale will continue the build from where it terminated*

The rest of this file explains some details about the make installation and files in the SCALE system. **Be sure to examine your log files to see they did not encounter fatal errors.** Some compiler warnings will be issued, and some runtime errors may be generated (floating underflows, underflow traps, output conversion errors, floating invalid, and floating invalid traps). Verify that all data libraries were created. Lists of the

bin and **data** directory contents appear later in this file. Check all ***.msgs** files in both the **sysname/work** and **sysname/smplprbs** directories. Review your ***.output** files in **work** (these contain output from the jobs that create data libraries) and **smplprbs** (these contain sample problem output that verifies that SCALE modules are installed and running correctly). Compare output in **smplprbs** to the ORNL results in the **output** directory using the **check_results** script. See "Running and verifying sample problem results" later in this document.

3. User registration

Registration of the software is highly encouraged. To that end, each user will need to access the SCALE 5 registration site http://public.ornl.gov/scale using a browser. Log in using your RSICC user ID and password. On the second screen in the field labeled "Computer name", enter the user ID that will be used when executing SCALE 5 and the SCALE 5 serial number supplied by RSICC with the software (AKA RSICC log number). An encrypted registration key will be returned. The user then either: (1) goes to the head Scale 5.0 directory and performs a "make register", or (2) executes the "register" script in the **cmds** directory. The user will be asked to enter the serial number and the encrypted key, then a file, **.scalerc**, will be generated in their home directory. To avoid transcription errors, it is suggested that one cut and paste the registration key from the output returned from RSICC.

4. SCALE online electronic manual

The SCALE manual is distributed in PDF format (in a compressed tar file) on the CD for installation on Unix workstations. Note that PDF files can be viewed on either Unix, Linux or Windows platforms. The manual is compatible with Acrobat Reader with Search. Version 6 for Windows is included on the install disk 1 for Windows users. Unix users must download the Reader from www.adobe.com. Be sure to check the box "Download the full version of Adobe Reader," if it is displayed.

Open the Table of Contents for the manual, **contents.pdf**. You may click on one of the titles in the table of contents to open that section of the manual. Each section has bookmarks and thumbnails to improve the ease in navigation of the document. If the bookmarks and thumbnails are not displayed when a section of the manual is opened, click on the tabs on the left side of the document.

The manual contains a searchable index. To activate the searchable index, open the SCALE manual. In Acrobat 6.0, click on the "Search" icon and click on "Use Advanced Search Options." Then click on "Select Index…" The SCALE 5 index is **manual/index/scale5-index.pdx**. You can search the entire SCALE Manual for words or phrases. Search returns a list of all sections of the SCALE Manual that contain the specified word(s). You may then select one of these files to view. Each occurrence of the word or phrase is listed.

5. Directory structure

This distribution includes the source code for each module in separate subdirectories under ***SCALE5*/src**, where ***SCALE5*** is the path to the head directory. Compilations are done under ***SCALE5*/sysname**. An executable is created for each module and stored in the ***SCALE5*/*sysname*/bin** directory, where ***sysname*** is the name of the system dependent directory.

Other **src** subdirectories include source code for routines common to multiple modules, such as the **scalelib** (SCALE general purpose library), **miplib** (Material Information Processor library), **klib** (KENO library), **sgmlib** (SCALE generalized geometry library), and **smclib** (SCALE Monte Carlo cross section mixing

library) subdirectories. Files in each of these are used to create an archive program library of objects which are saved in the ***sysname/*lib** directory to be accessed by the loader when building executables.

After installing the system, the ***SCALE5/*data** directory contains the SCALE Standard Composition Library, the AMPX master cross-section libraries, KENO biasing weights and albedos, ORIGEN-S binary libraries, the **aliases** file, and the ASCII data libraries for HEATING and QADCGGP. The **arplibs** directory contains the ORIGEN-ARP libraries. The **origen** directory contains the ORIGEN-S ASCII libraries. The **endf_b** directory contains the ENDF/B-V and ENDF/B-VI continuous energy cross-section libraries for CENTRM. There will also be a **data** directory under the ***sysname*** directory that includes the file **qatable** used to contain quality assurance information.

The ***SCALE5/*cmds** directory contains the scripts to execute SCALE and scripts to compile and link executables.

The ***SCALE5/*work** directory includes a Makefile, and input files to create the data libraries that are stored in directory ***SCALE5/*data**. Output files from the library creation jobs are stored in ***SCALE5/sysname/*work**.

6. Installation scripts to build SCALE

6.1. MAKE CONFIGURE

This command queries the operating system for machine type and copies the correct machine-specific file from the **config** directory to the head directory and calls it **config.data**. Compiler commands and switches are specified here. This step usually needs to be performed only once on a given machine. However, if **"SCALE="** or **"TMPDIR="** is redefined, then redo the *make configure*, as this will fix the scripts**.**

6.2. INSTALLSCALE

The **installscale and updatescale** scripts invoke Make to compile sources and to build executables. If installation is aborted or must be restarted, you may use **updatescale** to continue. Make compiles the *.f90 and *.c files in each source directory. The UNIX command **ar** is used to build relocatable object libraries from *.o files; these object libraries are created in the ***sysname/*lib** directory. For each program source directory, sources are compiled, **ar** is executed, and the compiler is run to build executables. The object files, relocatable object libraries, and executables are not deleted from the ***sysname*** subdirectories in which they were created because Make checks these original files for dependencies if it is invoked again.

Next, the LIBPNG and ZLIB files are built to create the LEGEND and WPNG executables. These are freeware toolkits for converting various image formats to and from portable formats, as documented by the README files in the associated directories. SCALE uses LEGEND and WPNG to create color PNG files containing 2-D color plots from KENO V.a, KENO-VI, CSAS, CSAS6, and PICTURE.

Another function this step is to tailor some scripts: **qatable** in the **data** directory and the **scale5**, **batch5**, **h7map**, **h7mon**, **h7tec** and other scripts in the **cmds** directory. Each of these new scripts in cmds will include your Scale5.0 directory pathname.

After all executables are built in their respective ***SCALE5/sysname*** directories, **installscale** copies executable programs to the ***SCALE5/sysname/*bin** directory and updates the file **qatable** in the **data** directory to reflect the correct code version and date on which the executables were created. There is more information about the **qatable** file later in this document. The **installscale** step creates 2 separate log files (one for creating executables and one for copying executables to the bin directory.) Be sure to examine both logs.

Make compiles source files with optimization. Problems occur in some modules with some (generally older) compilers. Compiling without optimization usually resolves these errors.

The main Makefile in the Scale5.0 head directory sets several parameters that are passed to the Makefiles in subdirectories. Make must always be invoked from the head directory. If Make fails, it prints an error message and terminates. After correcting the problem causing the error, invoke Make again from the Scale5.0 head directory using the **updatescale** script; and Make will check dependencies and continue building the system from the point where the error occurred.

A sample listing of the bin directory after making the executables is shown below. When a module is updated, the previous version is renamed with a ".last" extension. If a .last already exists, it is deleted.

```
-rwxrwxr-x   1 x4s      scale     275424 Apr 26 13:28 aim
-rwxrwxr-x   1 x4s      scale     200048 Apr 26 13:28 ajax
-rwxrwxr-x   1 x4s      scale     271232 Apr 26 13:28 ale
-rwxrwxr-x   1 x4s      scale     228864 Apr 26 13:28 alpo
-rwxr-xr-x   1 x4s      scale     246896 May  6 16:27 arp
-rwxrwxr-x   1 x4s      scale     246768 Apr 26 13:28 arp.last
-rwxrwxr-x   1 x4s      scale     129360 Apr 26 13:28 arplib
-rwxrwxr-x   1 x4s      scale      63376 Apr 26 13:28 awl
-rwxrwxr-x   1 x4s      scale     433632 Apr 26 13:28 bonami
-rwxrwxr-x   1 x4s      scale     642048 Apr 26 13:28 bonamist
-rwxrwxr-x   1 x4s      scale    2057568 Apr 26 13:28 c5toc6
-rwxrwxr-x   1 x4s      scale     239792 Apr 26 13:28 cadillac
-rwxrwxr-x   1 x4s      scale     212320 Apr 26 13:28 cajun
-rwxrwxr-x   1 x4s      scale    1158576 Apr 26 13:28 centrm
-rwxrwxr-x   1 x4s      scale     189360 Apr 26 13:28 cognac
-rwxrwxr-x   1 x4s      scale     259024 Apr 26 13:28 compoz
-rwxr-xr-x   1 x4s      scale     527920 May 10 16:28 couple
-rwxrwxr-x   1 x4s      scale     528000 Apr 26 13:28 couple.last
-rwxrwxr-x   1 x4s      scale    2158768 Apr 26 13:28 csas
-rwxrwxr-x   1 x4s      scale    2136496 Apr 26 13:28 csas6
-rwxrwxr-x   1 x4s      scale     225376 Apr 26 13:28 cutlet
-rwxrwxr-x   1 x4s      scale     131552 Apr 26 13:28 genwgts
-rwxrwxr-x   1 x4s      scale    1118448 Apr 26 13:28 gwas
-rwxrwxr-x   1 x4s      scale      90688 Apr 26 13:28 h7map
-rwxrwxr-x   1 x4s      scale      68112 Apr 26 13:28 h7maprz
-rwxrwxr-x   1 x4s      scale      51520 Apr 26 13:28 h7monitor
-rwxrwxr-x   1 x4s      scale      81296 Apr 26 13:28 h7tecplot
-rwxrwxr-x   1 x4s      scale     753184 Apr 26 13:28 heating
-rwxrwxr-x   1 x4s      scale     613264 Apr 26 13:28 htas1
-rwxrwxr-x   1 x4s      scale     751968 Apr 26 13:28 htng72
-rwxrwxr-x   1 x4s      scale     502672 Apr 26 13:28 ice
-rwxrwxr-x   1 x4s      scale     995168 Apr 26 13:28 k5tok6
-rwxrwxr-x   1 x4s      scale    1802448 Apr 26 13:28 kenova
-rwxrwxr-x   1 x4s      scale    1915712 Apr 26 13:28 kenovi
-rwxrwxr-x   1 x4s      scale     334560 Apr 26 13:28 kmart
-rwxrwxr-x   1 x4s      scale     343184 Apr 26 13:28 kmart6
-rwxrwxr-x   1 x4s      scale     297904 Apr 26 13:28 lava
-rwxrwxr-x   1 x4s      scale     601568 Apr 26 13:28 legend
-rwxrwxr-x   1 x4s      scale     233152 Apr 26 13:28 lsulib
-rwxrwxr-x   1 x4s      scale      52512 Apr 26 13:28 mal
-rwxrwxr-x   1 x4s      scale     342592 Apr 26 13:28 malocs
-rwxrwxr-x   1 x4s      scale     908272 Apr 26 13:28 modify
-rwxrwxr-x   1 x4s      scale    1021872 Apr 26 13:28 morse
```

```
-rwxrwxr-x   1 x4s      scale      35872 Apr 26 13:28 newsie
-rwxrwxr-x   1 x4s      scale    1036112 Apr 26 13:28 newt
-rwxrwxr-x   1 x4s      scale     445552 Apr 26 13:28 nitawl
-rwxrwxr-x   1 x4s      scale     696512 Apr 26 13:28 nitawlst
-rwxrwxr-x   1 x4s      scale    1021872 Apr 26 13:28 o0o006
-rwxrwxr-x   1 x4s      scale     406400 Apr 26 13:28 o0o016
-rwxrwxr-x   1 x4s      scale    1031616 Apr 26 13:28 o0o106
-rwxrwxr-x   1 x4s      scale     471024 Apr 26 13:28 ocular
-rwxrwxr-x   1 x4s      scale     451248 Apr 26 13:28 oculr72
-rwxrwxr-x   1 x4s      scale     308800 Apr 26 13:28 opus
-rwxrwxr-x   1 x4s      scale     866944 Apr 26 13:28 origen
-rwxrwxr-x   1 x4s      scale     314848 Apr 26 13:28 paleale
-rwxrwxr-x   1 x4s      scale     234064 Apr 26 13:28 perfume
-rwxrwxr-x   1 x4s      scale     406384 Apr 26 13:28 picture
-rwxrwxr-x   1 x4s      scale     442944 Apr 26 13:28 pmc
-rwxrwxr-x   1 x4s      scale     129424 Apr 26 13:28 prism
-rwxrwxr-x   1 x4s      scale     335536 Apr 26 13:28 qadcggp
-rwxrwxr-x   1 x4s      scale     845536 Apr 26 13:28 qads
-rwxrwxr-x   1 x4s      scale     406944 Apr 26 13:28 rade
-rwxrwxr-x   1 x4s      scale     188192 Apr 26 13:28 reorg
-rwxrwxr-x   1 x4s      scale     772928 Apr 26 13:28 sams
-rwxrwxr-x   1 x4s      scale    1218592 Apr 26 13:28 sas1
-rwxrwxr-x   1 x4s      scale    1047376 Apr 26 13:28 sas2
-rwxrwxr-x   1 x4s      scale    1316256 Apr 26 13:28 sas3
-rwxrwxr-x   1 x4s      scale    1529664 Apr 26 13:28 sas4
-rwxrwxr-x   1 x4s      scale      85008 Apr 26 13:27 scale
-rwxrwxr-x   1 x4s      scale      27552 Apr 26 13:27 shell
-rwxrwxr-x   1 x4s      scale    1459360 Apr 26 13:28 smores
-rwxrwxr-x   1 x4s      scale     867712 Apr 26 13:28 starbucs
-rwxrwxr-x   1 x4s      scale     259072 Apr 26 13:28 swif
-rwxrwxr-x   1 x4s      scale      62432 Apr 26 13:28 toc
-rwxrwxr-x   1 x4s      scale    2073312 Apr 26 13:28 triton
-rwxrwxr-x   1 x4s      scale    1496272 Apr 26 13:28 tsunami-1d
-rwxrwxr-x   1 x4s      scale    2350160 Apr 26 13:28 tsunami-3d
-rwxrwxr-x   1 x4s      scale    1176752 Apr 26 13:28 tsunami-ip
-rwxrwxr-x   1 x4s      scale     196736 Apr 26 13:28 wax
-rwxrwxr-x   1 x4s      scale      52432 Apr 26 13:28 wgt
-rwxrwxr-x   1 x4s      scale     369536 Apr 26 13:28 worker
-rwxrwxr-x   1 x4s      scale     344400 Apr 26 13:28 wpng
-rwxrwxr-x   1 x4s      scale     233104 Apr 26 13:28 xsdose
-rwxrwxr-x   1 x4s      scale     870272 Apr 26 13:28 xsdrn
-rwxrwxr-x   1 x4s      scale     134288 Apr 26 13:28 xseclist
```

6.3. MAKELIBRARIES

SCALE reads AMPX master libraries in binary form. This distribution includes binary libraries that are written to ***SCALE5*/data**. AIM and PERFUME are typically run to convert these libraries between binary and ASCII. AMPX master libraries are included in ASCII form on 2 separate CDs in case there are some computer systems that cannot read big endian binary files.

The control modules read the Standard Composition Library. Sect. M8 of the SCALE Manual contains information on using the Standard Composition Library. COMPOZ reads the ASCII **stdcmplib.input** file from the **work** directory and creates the binary **scale.rev*xx*.sclib** library (where *xx* is the revision number) in the **data** directory.

The standard ORIGEN-S libraries are generated from ASCII data and are written as binary libraries using ORIGEN, COUPLE and SAS2. The ASCII data are distributed in the **data/origen** directory and will be written to the **data** directory after conversion. The SCALE input for this conversion is in **origen_data.input** in the **work** directory. The REORG utility (Sect. M6 of SCALE Manual) may be used for converting ORIGEN-S and ORIGEN-ARP working libraries between ASCII and binary modes.

Two card-image libraries, **attenuat** for attenuation coefficients and **buildup** for build-up factor data, are included for use with QAD-CGGP. The **scale.rev01.qadxslib** file is in AMPX master format and contains the names of all the nuclides in the Standard Composition Library available to QADS.

The **makelibraries** command builds the binary data libraries using the Makefile in the **work** directory and puts them in the ***SCALE5*/data** directory. The MAL utility converts the KENO albedos library to binary. Likewise, the WGT program converts the KENO biasing weights library to binary. In SCALE 5, the "big endian" binary format is used on all machines for the AMPX master libraries. The Standard Composition Library is created as a random access binary file, as discussed earlier. Next, the ORIGEN binary libraries are created. ORIGEN-ARP libraries, which are also in "big endian" format, are in **data/arplibs**.

If any of these jobs abort and need to be rerun, the user must remove the ***.done** files. Targets are in the head data directory for the library targets. If, for example, the **makelibraries** job exceeds temporary disk space, the user must update the values for TMPDIR in **installscale.patn** and **cmds/batch5.patn**; delete **data/*.done**; execute **make configure**; and then run **makelibraries** again.

A sample listing of the contents of the ***SCALE5*/data** directory after running **make libraries** is shown below.

```
drwxr-xr-x   2 x4s      scale         512 Sep 12  2000 BE.xsecs
-rw-r--r--   1 x4s      scale      112000 Apr 13 13:20 albedos
-rw-r--r--   1 x4s      scale        2417 Nov 26 12:34 aliases
-rw-r--r--   1 x4s      scale        9488 Apr 14 09:24 arpdata.txt
drwxr-xr-x   2 x4s      scale        9728 Apr 13 12:49 arplibs
-r--r--r--   1 x4s      scale       27386 Sep 18  2001 attenuat
-r--r--r--   1 x4s      scale       97560 Sep 12  2000 buildup
drwxr-xr-x   4 x4s      scale         512 Apr 12 11:37 endf_b
-r--r--r--   1 x4s      scale      243405 Sep 18  2001 h7matlib
drwxr-xr-x   2 x4s      scale         512 Apr 12 13:50 keno
-rw-r--r--   1 x4s      scale           0 Apr 13 13:20 keno.done
drwxr-xr-x   2 x4s      scale         512 Apr 12 13:50 origen
-rw-rw-r--   1 x4s      scale           0 Apr 13 12:54 origen.done
-rw-rw-r--   1 x4s      scale     1136872 Apr 13 12:54 origen.rev01.maphh2ob
-rw-rw-r--   1 x4s      scale      969248 Apr 13 12:54 origen.rev01.maphnobr
-rw-rw-r--   1 x4s      scale     1136872 Apr 13 12:54 origen.rev01.maphuo2b
-rw-rw-r--   1 x4s      scale     1052064 Apr 13 12:54 origen.rev02.pwr33gwd
-rw-r--r--   1 x4s      scale        7891 Apr 14 18:10 qatable.upd
```

```
-rwxr--r--   1 x4s      scale         810 Nov 10  2003 scale.messages
-rw-r--r--   1 x4s      scale     5843192 Feb  6      scale.rev00.44groupanlcov
-rw-r--r--   1 x4s      scale     4991598 Feb  6 11:52 scale.rev00.44groupv5cov
-r--r--r--   1 x4s      scale      205120 Sep 10  1999 scale.rev01.qadxslib
-rw-r--r--   1 x4s      scale      363192 Jul 16  2002 scale.rev04.xn16
-rw-r--r--   1 x4s      scale      292584 Jul 16  2002 scale.rev04.xn22g18
-rw-r--r--   1 x4s      scale     9342720 Jul 16  2002 scale.rev05.xn218
-rw-r--r--   1 x4s      scale      836192 Jul 16  2002 scale.rev05.xn27
-rw-r--r--   1 x4s      scale     1256296 Jul 16  2002 scale.rev06.xg18
-rw-r--r--   1 x4s      scale     2791576 Jul 16  2002 scale.rev06.xn27burn
-rw-r--r--   1 x4s      scale     1448716 Mar 23 16:01 scale.rev06.xn27g18
-rw-r--r--   1 x4s      scale    95360436 Mar 26 06:55 scale.rev14.xn238
-rw-r--r--   1 x4s      scale    11351596 Mar 26 06:55 scale.rev16.xn44
-rw-rw-r--   1 x4s      scale      187560 Apr 13 12:54 scale.rev25.sclib
-rw-rw-r--   1 x4s      scale           0 Apr 13 12:54 std.done
-rw-r--r--   1 x4s      scale         358 Nov 26 14:11 update
-rw-r--r--   1 x4s      scale      109284 Apr 13 13:20 weights
```

6.4. RUNSMPLS

The **runsmpls** command runs all sample problems using the **batch5** command, which returns output to the ***sysname/*smplprbs** directory. It may then be compared with the output tables from ORNL using the **check_results** command. See “Running and verifying sample problem results” below.

This step creates *.done files in ***sysname/*smplprbs** for each module. If the sample jobs abort and need to be rerun, the user must remove the ***.done** targets before executing **runsmpls** again. If **runsmpls** terminates without error, it will delete the ***.done** files.

6.5. MAKE CLEAN

If one wants to clean up and restore the directory to its initial structure, typing **make clean** from the Scale5.0 head directory deletes all object, sample output, and executable files created in this system.

7. Running and verifying sample problem results

Runtimes for sample problems vary from about 8 hours to more than 48 hours depending on the speed of the machine. On an AMD Athlon MP 2000+ running RedHat Linux,the sample problems ran for 12 hours, of which 6.75 hours were on TSUNAMI-3D problems. On an IBM RS/6000 Model 590 running AIX, the sample problems ran for just under 34 hours, of which 24 were on TSUNAMI-3D problems.

Note that one sample problem named **centrm_large.input** is not executed by default. It models spent fuel and requires a large amount of memory (approximately 0.5 GB) and scratch disk space (approximately 1 GB). This sample problem is optional and may be run by the normal means of using the **batch5** command.

The TSUNAMI-3D sample problems account for more than 50% of the runtime. If you are not planning to use TSUNAMI-3D, you may delete ‘tsunami-3d’ from the list of names in file *SCALE5*/SMPL before invoking the runsmpls script.

Some of the TRITON and TSUNAMI problems require a large amount of memory. To increase physical memory or the size of the data area for the current shell environment on AIX, the user can issue one or both of the following commands. The syntax varies on different computers. System administrators can reset default limits.

ulimit -m unlimited

ulimit -d unlimited

ulimit –a [queries the system for user's limits]

Once the sample problems are completed, the **check_results** script calls the **make_results_table** script that extracts key data from sample problem output files and writes them into tables. The tables are then compared to similar tables from ORNL that are distributed in the ***SCALE5*/tables** directory**.** The differences are written to the file **compare_output.txt in the user's *sysname*/smplprbs directory**. This procedure significantly reduces the amount of data compared, and consequently, the time required for the installer to verify that the sample problem results are acceptable. Note that the **tables** directory contains the tables from ORNL output on several different operating systems, as labeled. If tables for your operating system are available, the script will compare with those tables. If tables do not exist for your operating system, you may want to try comparing your results to those of a different system to find which set compares most closely.

The differences for most of the non-Monte Carlo functional module sample problems should be fairly small, i.e., numerical round-off in the last one or two digits. For Monte Carlo results from KENO V.a, KENO-VI, and MORSE (this includes CSAS, CSAS6, TSUNAMI-3D, KMART, KMART6, SAS3, and SAS4), it is common for results to vary within 1 to 3 standard deviations.

Because the TSUNAMI-1D and -3D output files are very large, a special TSUNAMI-IP sample problem named **tsunami-3d_summary.input** is executed after the standard TSUNAMI-1D and -3D sample problems to generate a summary output file for comparison. This problem compares the sensitivity profiles generated by the standard TSUNAMI-1D and -3D sample problems on the test machine to the sensitivity profiles generated at ORNL by calculating the G index, which assesses the similarity of two systems based on normalized differences in the energy-dependent sensitivity data for fission capture and scatter. The resulting G values should be near 1.0 (typically within 2-3%). For comparisons of TSUNAMI-3D results for cases run on machines with the same operating system and compiler, the G indices **may** (but not necessarily) be 1.0, indicating an exact match. Because of the size of the TSUNAMI-3D6 case, the number of unique starting points in the initial KENO V.a generation varies for different runs. The resulting G index for this case should be near 1.0 (typically about 0.98), but should not be expected to exactly match the value from the ORNL case.

The COMPARE_RESULTS.TXT file should be examined very carefully. There must be corresponding lines from each output file that is being compared. In other words, if one or more sample problems fail to run, the extracted lines from the ORNL output files will be listed as differences in COMPARE_RESULTS.TXT. A warning will be printed to the log file if values are not found for a specific problem.

8. How to execute SCALE

It is strongly recommended that users add the *SCALE5*/cmds directory to their path. The command **#!/bin/ksh** which is specified at the beginning of most scripts forces them to run in the korn shell (k shell). A few scripts run in the c shell (csh). The **batch5** script in the **cmds** directory runs SCALE in batch mode and sets environmental variables necessary to run the **scale5** script and calls the **scale5** script. It also removes the temporary directory after execution. The **scale5** script copies or links the executables and data

files from their respective subdirectories to a temporary directory for execution then invokes the driver to read the **sysin** input file and executes the problem. On the local system a directory **/usr/tmp** has been allocated so that all users have write access.

It is important to note that the link (**ln –s** command) will not overwrite files that exist or are already linked, so it may be necessary to remove files before executing again a problem that failed using the **scale5** command (this is not an issue if the **batch5** script is used because it starts with a clean temporary directory). Extra data files that are needed to execute any problem must be present (or linked) in the temporary directory where the problems will be run.

To run SCALE, use one of the following commands

batch5 [-m -r -t -p -n N] inputfile [outputfile]

where
-m = print messages from KENO, XSDRN, MORSE, NEWT to the screen (optional flag)
-r = don't remove temporary working directory at end of job (optional flag)
-t = use specified existing temporary working directory (optional flag)
-p = print block letter banner pages in output (optional flag)
-n N = nice value to be used during execution (default N = 2)
]
inputfile = input file name (required) (can include input path)
outputfile = output file name (optional) (can include path - all output files will go to the same directory as the output file; if not specified, the output file will be derived from the input file, and output will be returned to the input file directory)

or

scale5 ***inputfile*** **[*outputfile*]**

where ***inputfile*** is the SCALE input file without the extension. This script works with input file names with no extension (e.g., case1), or extensions of .inp, or .input (e.g., case1.inp, case1.input). The output file will automatically be written with the .out or .output extension (e.g., case1.out or case1.output), respectively. The batch5 script also automatically saves system messages in a .msg or .msgs file (e.g., case1.msg or case1.msgs), respectively. The **batch5** script automatically sets the environmental variables needed by the **scale5** script, allocates a temporary directory for the calculation, calls the **scale5** script, and deletes the temporary directory after **scale5** completes execution. If you use the **scale5** script instead, you must set the required environmental variables and allocate the temporary directory. The directory will not be deleted when execution is completed. The environmental variables that must be set are:

SCALE - the full directory path where SCALE is installed
TMPDIR - the full path for the temporary directory

After executing the problem using the **scale5** script, output files are collected and returned to the user's directory. If there is a problem with execution, sometimes the files are not returned. The user can manually go to the temporary directory and examine the files, print, _out000n and _prt000n, etc., to find the problem. The files beginning with _ are numbered with an integer that increments each time another file is written.

The documentation for the SCALE driver is included in Section M1. When executing the driver (named **scale**), input is read from file **sysin**. The driver calls each module requested and passes the input for each module as it is called via the file **input**. The output from the driver is written to file **print**. The functional modules write to **_out0000**, **_out0001**, **_out0002**, etc. The control modules write output files called

_prt0000, **_prt0001**, **_prt0002**, etc. After each control module is finished, the driver collects the **_prt*** and **_out*** files into a unique **_tot000n** file, and removes the **_prt*** and **_out*** files.

While data can be read on logical units 1-99, some units are reserved as defaults for certain programs. When defining input, avoid using units 1-19, 21-28, and 79-99. SCALE expects to read SCALE libraries on units 21-28 and 79-88; it reads the Standard Composition Library on 89 and uses other units for temporary files. A list of unit numbers and their default assignments is included in Table 1. A list of unit numbers that are set to “big endian” in SCALE is given in Table 2.

Table 1. SCALE 5 Logical Unit Master List

Unit Number	Module	Type of File
1	BONAMI	Master library
	NITAWL	Master library
	SAMS	Master library
2	ICE	Macroscopic working library
3	XSDRNPM	Cell-weighted working library
	ICE	Cell-weighted working library
	KENO V.a	Cell-weighted working library
	KENO VI	Cell-weighted working library
	NEWT	Cell-weighted working library
	SAMS	Cell-weighted working library
	SWIF	Binary interface data file
4	NITAWL	Working library
	ICE	Working library
	KENO V.a	Working library
	KENO VI	Working library
	XSDRNPM	Working library
	TRITON	Working library
	SWIF	Working library
	NITAWLST	Working Library
	SAMS	Working Library
8	BONAMI	Random-access scratch file
	ICE	Random-access scratch file
	KENO V.a	Random-access scratch file
	KENO VI	Random-access scratch file
	XSDRNPM	Random-access scratch file
	SWIF	Auxiliary output file
9	BONAMI	Random-access scratch file
	NITAWL	Random-access scratch file
	ICE	Random-access scratch file
	KENO V.a	Random-access scratch file
	KENO VI	Random-access scratch file
	XSDRNPM	Random-access scratch file
	CSAS	Random-access scratch file
10	ICE	Random-access scratch file
	KENO V.a	Random-access scratch file
	KENO VI	Random-access scratch file
	XSDRNPM	Random-access scratch file
	CSAS	Random-access scratch file

	SENLIB	Implicit-sensitivity-data file
	SAMS	Implicit-sensitivity-data file
11	BONAMI	Master library
	CSAS	Master library
	SAS3	Master library
	SAS4	Master library
	NEWT	Master library
	ORIGENS	Scratch file
	NITAWLST	Implicit-sensitivity-data file
	SAMS	Implicit-sensitivity-data file
12	SAS4	XSDRN adjoint flux file
	SAS3	
	SAS2	Scratch file
	QADCGGP	
	SWIF	Plot data file
	BONAMIST	Implicit-sensitivity-data file
	SAMS	Implicit-sensitivity-data file
13	ORIGENS	FIDO edit of input
	SWIF	Collection of auxiliary output files
	MIPLIB	Working CENTRM point cross-section library
	CENTRM	Working CENTRM point cross-section library
	PMC	Working CENTRM point cross-section library
14	ICE	Monte Carlo library
	KENO V.a	Monte Carlo library
	KENO VI	Monte Carlo library
	SAS2	Scratch file
	SWIF	Plot data file
15	SAS2	Scratch file
	TRITON	ORIGENS library
	SWIF	Plot data file
16	KENO V.a	Scratch file
	KENO VI	Scratch file
	XSDRNPM	Angular flux file
	SMORES	Scratch file
17	XSDRNPM	Scalar flux file
	COUPLE	Scratch file
	MORSE	Scratch file
18	BONAMI	Scratch file
	NITAWL	Scratch file
	XSDRNPM	Scratch file

	COUPLE	Scratch file
19	BONAMI	Scratch file
	NITAWL	Scratch file
	COUPLE	Scratch file
	XSDRNPM	Scratch file
20	NITAWLST	Master sensitivity cross section file
	SAMS	Master sensitivity cross section file
21	ORIGENS	33 GWD PWR library
23	ORIGENS	Master Photon Data - No Bremstrahlung
24	ORIGENS	Master Photon Data - H2O Bremstrahlung
26	ORIGENS	Master Photon Data - UO2 Bremstrahlung
27	ORIGENS	BCD library - decay data
	COUPLE	BCD library - decay data
28	ORIGENS	BCD library – cross-section data
	COUPLE	BCD library – cross-section data
29	COUPLE	Input binary library
30	ORIGENS	Input binary library
31	NEWT	Binary interface data file (forward case)
	XSDRNPM	Binary interface data file (forward soln – TSUNAMI-1D)
	SAMS	Binary interface data file (forward soln – TSUNAMI-1D)
32	COUPLE	Output binary library
	NEWT	Binary interface data file (adjoint case)
	XSDRNPM	Binary interface data file (adjoint soln – TSUNAMI-1D)
	SAMS	Binary interface data file (XSDRNPM adjoint soln – TSUNAMI-1D)
33	SAS2	Output ORIGENS Multiburnup library
	TRITON	Output ORIGENS Multiburnup library
34	XSDRNPM	Flux Guess File
35	KENO V.a	Binary interface data file (forward soln – TSUNAMI-3D)

	SAMS	Binary interface data file (forward soln – TSUNAMI-3D)
36	KENO V.a	Binary interface data file (adjoint soln – TSUNAMI-3D)
	SAMS	Binary interface data file (adjoint soln – TSUNAMI-3D)
53	SAS2	Scratch file
55	SAS2	Halt file for restart
60	SAS2	Scratch file
70	SCALE/MIPLIB	User-specified master cross-section library
71	SAS2	Binary output concentrations
	ORIGEN-S	Binary output concentrations
72	SAS2	BCD output concentrations
73	XSDRNPM	Input and derived data file
74	SAS2	Scratch file
75	XSDRNPM	Activity output file
76	XSDRNPM	Balance table output file
77	MIPLIB	Master point cross-section data file
	SAMS	Cross-section covariance data library
	TSUNAMI-IP	Cross-section covariance data library
79	KENO V.a	KENO V.a albedo file
	KENO VI	KENO VI albedo file
80	KENO V.a	KENO V.a weight file
	KENO VI	KENO VI weight file
88	SCALE	Master Cross Section library
89	SCALE	Standard Composition library
90	CSAS	Random-access scratch file
	SAS3	Random-access scratch file
	SAS4	Random-access scratch file
	SAS2	Scratch file

91	SAS2	Scratch file
	NEWT	Alternate binary input file
	SAS4	Scratch file
	SMORES	Plot data file
92	SAS4	Scratch file
	SMORES	Collection of input files for restart
93	SMORES	Direct-access data file
	SAS4	PICTURE input file
94	SAS4	Scratch file
95	MORSE	Input file
	SAS4	MORSE input file
	KENOV.a	Binary restart file
	KENOVI	Binary restart file
96	MORSE	Input file (SAS4)
98	SAS4	Adjoint XSDRN input file
99	SCALE	Scratch

Table 2. SCALE 5 Big Endian Logical Unit Numbers for Unix/Linux

Unit Number	Codes	Description
21	SAS2, TRITON, ORIGEN-S	PWR library
22		
23	SAS2, TRITON, ORIGEN-S	ORIGEN-S master photon library (No bremsstrahlung)
24	SAS2, TRITON, ORIGEN-S	ORIGEN-S master photon library (H_2O bremsstrahlung)
25		
26	SAS2, TRITON, ORIGEN-S	ORIGEN-S master photon library (UO_2 bremsstrahlung)
33	SAS2, TRITON, ORIGEN-S, ARP, XSECLIST	ORIGEN-S multiburnup library
77	ARP, ARPLIB TSUNAMI-IP, SAMS	ORIGEN-S multiburnup library Cross-section covariance data
78		
79	KENO V.a, KENO-VI	KENO albedo library
80	KENO V.a, KENO-VI	KENO weighting library
88	MIPLIB	Master cross section library
89	MIPLIB	Standard composition library

9. Qatable

The ASCII file **qatable** is required for execution on all computers. It is located in the ***sysname***/**data** directory and is used by the control and functional modules to check the latest revision date and version for the modules. This is to help implement a quality assurance procedure. In order to validate the **qatable** for use in a QA procedure, the dates and versions are changed by Make to agree with the dates on which the executables were created. The changes are made by **make install** which you can invoke by running either the **installscale** or the **updatescale** script. A portion of this file, illustrating the format, is listed below:

```
  scale   5.0
aim           aim           5.0.0  p26_apr_2004/scale/scale5.0/OSF1_V5/bin
ajax          ajax          5.0.0  p26_apr_2004/scale/scale5.0/OSF1_V5/bin
ale           ale           5.0.0  p26_apr_2004/scale/scale5.0/OSF1_V5/bin
alpo          alpo          5.0.0  p26_apr_2004/scale/scale5.0/OSF1_V5/bin
arp           arp           5.0.0  p06_may_2004/scale/scale5.0/OSF1_V5/bin
arplib        arplib        5.0.0  p26_apr_2004/scale/scale5.0/OSF1_V5/bin
```

```
awl           awl           5.0.0  p26_apr_2004/scale/scale5.0/OSF1_V5/bin
bonami        bonami        5.0.0  p26_apr_2004/scale/scale5.0/OSF1_V5/bin
bonamist      bonamist      5.0.0  p26_apr_2004/scale/scale5.0/OSF1_V5/bin
c5toc6        c5toc6        5.0.0  p26_apr_2004/scale/scale5.0/OSF1_V5/bin
cadillac      cadillac      5.0.0  p26_apr_2004/scale/scale5.0/OSF1_V5/bin
cajun         cajun         5.0.0  p26_apr_2004/scale/scale5.0/OSF1_V5/bin
centrm        centrm        5.0.0  p26_apr_2004/scale/scale5.0/OSF1_V5/bin
compoz        compoz        5.0.0  p26_apr_2004/scale/scale5.0/OSF1_V5/bin
cognac        cognac        5.0.0  p26_apr_2004/scale/scale5.0/OSF1_V5/bin
couple        couple        5.0.0  p10_may_2004/scale/scale5.0/OSF1_V5/bin
csas          csas          5.0.0  p26_apr_2004/scale/scale5.0/OSF1_V5/bin
csas6         csas6         5.0.0  p26_apr_2004/scale/scale5.0/OSF1_V5/bin
```

The name of the executable files begins in column 1, followed by the module name and version number. The “p” indicates that the particular module has production status. Other valid flags are “t” for test and “m” for migration. The revision date follows in dd_mmm_yyyy format. The last field is the pathname.

10. Aliases

When executing the driver, a file called **aliases** is required. This file will be fetched by the script and accessed by the driver to determine the actual name of the executable to call. For example, it recognizes when the user specifies any one of the sequences **csasn, csas1, csasi, csas25**, or **csas4**, that the control module **csas** should be called. This file is stored in the ***SCALE5*/data** directory and is accessed by the script **scale5**.

10. Javapeno

Javapeno requires installation of version 1.3 (or later) of the Java 2 Virtual Machine (VM). The use of the latest available version is recommended. The command "java -version" lists the currently installed VM. The Java 2 Runtime Environment (J2RE) or Java 2 Software Development Kit (J2SDK) for the Java 2 Platform Standard Edition (J2SE) is adequate to execute Javapeno. This software is freely distributed by Sun and can be accessed at http://java.sun.com/downloads/index.html for Windows, Linux and Solaris systems. Other platforms should have J2RE ports available from the vendor. Some vendor links are given below.

AIX: http://www.ibm.com/developerworks/java/jdk/aix/service.html
Alpha: http://h18012.www1.hp.com/java/alpha/
HP: http://www.hp.com/products1/unix/java/index.html
Mac OS X: http://www.apple.com/downloads/macosx/apple/javaupdate.html