

LA-UR-13-24008

Approved for public release; distribution is unlimited.

Title: MCNP6(TM) Release 1.0: Creating and Testing the Code Distribution

Author(s): Cox, Lawrence J.
Matthews, Scott D.

Intended for: Distribution thru RSSIC with the MCNP6 Release 1.0 Package
Report
Web

Issued: 2013-05-31



Disclaimer:

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

MCNP6™ Release 1.0: Creating and Testing the Code Distribution

¹*L. J. Cox, ²S. D. Matthews*

¹*XCP-DO, ²HCP-1*

Los Alamos National Laboratory, Los Alamos, NM

Abstract

This report documents the preparations for and testing of the production release of MCNP6™ through RSICC at ORNL. It addresses tests on supported operating systems (Linux, MacOSX, Windows) with the supported compilers (Intel, Portland Group International, PathsScale and gfortran). Verification and Validation test results are documented elsewhere. This report *does not* address in detail the overall packaging of the distribution. Specifically, it does not address the nuclear and atomic data collection, the other included software packages (MCNP5 and MCNPX) and the collection of reference documents.

The MCNP6 Portion of the RSICC Distribution

MCNP6 is being prepared for release through the Radiation Safety Information Computational Center (RSICC). The RSICC distribution of MCNP6 will contain the following components:

- MCNP6: with or without source code
- Nuclear and atomic data files
- Reference documentation collection
- MCNP5: with or without source code
- MCNPX: with or without source code

This paper addresses the generation and testing of the MCNP6 component. Other components are briefly discussed, but will be or have been documented elsewhere.

A CVS branch was created in the MCNP project's TeamForge® repository to isolate the production release files. This was done to provide a controlled, stable version for testing and release preparation. Development continued on the main CVS trunk of the repository. Fixes and features to the main CVS trunk were merged into the release branch as needed.

The name of the branch is **MCNP6_RELEASE1_branch**, which is case-sensitive. There is a tag on the branch, **MCNP6_RELEASE1** and will point to the final distribution point. Post-release fixes (if any) will be separately tagged as needed.

Changes to the Source Code

For the release (on the branch), the ACODE capability in MCNP6 was removed via preprocessing. This feature is a compile-time option in the main CVS repository that is not intended for external distribution. Some obsolete source code directories were also removed from the release branch. Other than these changes, any source differences should be due to changes made to the main CVS trunk that have not been migrated to the release branch.

The omission of the ACODE feature leads to small changes in the output for any criticality tests, therefore many templates on the release branch differ from those on the main CVS trunk.

Unclassified

Changes in the Testing Directory

There are also changes in the `Testing/` directory from the main CVS trunk to the release branch. Paralleling the removal of `ACODE` from the source code, the `ACODE` validation test suite has been removed. In addition, all of the special data files were removed from `Testing/xsec_data`. Still present are the testing directory and library, the files needed for testing TYPE-2 data and the test-specific `xmdir` files. All standard data files must now be made available via the `DATAPAPH` search.

Note: The data directory layout has been modified to accommodate MCNP specific data files in addition to the nuclear and atomic data files provided by LANL's Data Team (XCP-5). The details of the new layout are documented elsewhere.

Exporting for Release

To extract the files for release from the CVS repository, execute the following commands:

```
> cvs export -r MCNP6_RELEASE1 [-d <dirname>] MCNP6
> cd <dirname>
> Utilities/purge_export purge
```

The optional `[-d <dirname>]` allows you to rename the directory created. If not specified, `<dirname>` defaults to the module name, `MCNP6`.

Executing the `Utilities/purge_export` script as shown removes the files not intended for distribution. This purge includes LANL-use-only scripts and test suites and other unnecessary files. This script also deletes itself being no longer needed after the export and purge.

Test Plan

This section addresses testing on the supported operating systems with supported compilers and parallelism options. It briefly touches on the addition V&V testing, but those results are reported in reference [1].

Build Configurations and Operating Systems

MCNP6 is supported on Linux, MacOS and Windows. The compilers that are supported for building MCNP6 on each system are listed in Table 1. The release testing included these operating systems and compilers run in many modes. Full details of the testing are provided later in this document.

Table 1: MCNP6 Supported Compilers. Defaults are shaded in green.

Compiler-specific Configuration Keys			Version(s) supported	OS		
Category	Key (defaults)	Description		Linux	MacOS	Win
Fortran Compiler Selection	intel ^{1,3}	ifort	11.1.x 12.1.5	X	X	X
	gfortran	gfortran	4.7.x	X		X
	portland	pgf90	12.10	X		
	pathscale	pathf90	4.0	X		
C Compiler Selection	gcc ²	Gnu C (gcc)	4.7.x	X	X	X
	msvc ²	Microsoft Visual C	Visual Studio 2010			X
	win64	64bit gcc compiler	—			X
Notes	1) The default build configuration on all platforms is “intel omp plot”. 2) The default C compiler for Linux and MacOS is gcc. The default C compiler for Windows is Microsoft Visual C. 3) Intel versions 13.x will not compile MCNP6 as of the MCNP6 release date.					

Test Suites

The test suites used for the release testing are listed in Table 2. Some of these test suites will be excluded from the distribution, but were kept in the branch to permit their use in the release testing. The regression-like test suites comprise most of our continuous build & test system (CBTS), which is run whenever changes occur in the CVS repository. The V&V test suites are run less frequently (but still often) to provide comparison to benchmark data: analytic, experimental or production-release baseline values.

Table 2: Test Suites Used for Production Release Testing

Regression-like Test Suites	Description: Compared to templates to ensure MCNP6 works as expected.
REGRESSION	The standard MCNP6 regression test suite routinely run upon any changes and during installation.
PHOTONS	Photon transport tests
ELECTRONS	Electron transport tests
MUONS	Muon transport tests
MAG_FIELDS	Magnetic field transport tests
PHTVR	Pulse-height tally variance reduction tests

Table 2: Test Suites Used for Production Release Testing

Regression-like Test Suites	Description: Compared to templates to ensure MCNP6 works as expected.
FEATURES/ DAWWG FMESH_INC MODEL_DEV POINT_DETECTORS UNC	Five (5) test suites for special MCNP6 features
MCNPX_65 MCNPX_EXTENDED/	The MCNPX regression suite plus twenty-two (22) additional test suite developed for MCNPX. A total of 541 tests overall.
V&V Test Suites (*) Not included in distribution	Description: Compared to analytic, experimental and/or MCNP5 results
KOBAYASHI	Six (6) benchmarks for simple geometries with void regions
VALIDATION_CRITICALITY	Thirty-one (31) criticality benchmarks compared to MCNP5 and experimental results.
VALIDATION_CRIT_EXPANDED	Additional criticality benchmarks (119) compared to MCNP5 and experimental results.
VALIDATION_CRIT_SBCS (*)	The LANL Criticality-Safety Group’s 194 problem test suite compared to MCNP5 and experimental results.
VALIDATION_ROSSI_ALPHA	Fourteen (14) validation tests for the Rossi-alpha capability.
VERIFICATION_KEFF	Analytic verification tests (75) of MCNP criticality algorithms.
VALIDATION_SHIELDING	Shielding benchmarks (18) compared to experimental data.
VERIFICATION_SHIELDING (*)	Shielding benchmarks (78) compared to MCNP5 results.

The regression-like test suites contain 1227 problems overall. The REGRESSSION suite by itself contains 147 problems. The listed V&V test suites contain 535 problems.

During MPI execution, there are four (of 1227) tests that are not run. These are tests in the MCNPX_EXTENDED suites, avr and class. For testing with gfortran, there is one test that is skipped: REGRESSION/inp124. This test is known to have issues in any execution mode on Linux with gfortran-built executables.

OS/Compiler Test Results

This section summarized the results of running the regression-like test suites. Test results are broken down by operating system, compiler, build-configuration and execution-mode, in that order.

Suites are run with the ‘nofuzzy’ option for Intel-compiler builds and with the “fuzzy” option for other compilers. The nofuzzy option results in character by character comparison between templates and results. The fuzzy option reduces the precision of real numbers by two significant digits to provide “looser” comparison of the differences that appear with different compilers.

For Windows and MacOS systems, only Intel and gfortran are currently supported. Table 3 lists the build and execution modes to be carried out and which test suites will be used for which modes (Regression-like and/or V&V suites). Regression-like suites will be run on all platforms with all supported compilers in all supported parallelism modes.

The templates for each regression-like suite are created with the default build—CONFIG=“intel omp plot”—on Linux and Windows run sequentially. The Linux-created templates are used during MacOS testing. Results from all other builds and execution modes

are compared to these templates. V&V suites are only run using the Intel/MPI executable in parallel because those results are compared to benchmark data of one form or another.

In general all four compilers (two on Windows/MacOS) work for sequential build and execution. The Intel Fortran compiler (the MCNP6 default on all systems) will also build for parallel execution using OpenMP, OpenMPI, mixed parallelism using both.

OpenMP: MCNP6 supports OpenMP threading, but only through the Intel compiler. The other compilers can be used to build threaded versions, but the execution fails on many tests. For the other compilers, the tests that run to completion with OpenMP seem to give valid answers. User Beware.

OpenMPI: Testing was primarily done using the OpenMPI version 1.4.5. A limited amount of testing was done with OpenMPI v.1.6.3. The Intel, PGI and gfortran compilers work well with OpenMPI. PathScale cannot currently be used for OpenMPI parallelism as that compiler does not currently support a Fortran-2003 feature that MCNP6 uses for MPI communication. The defaults are shaded (light green) and in bold.

Table 3: Execution Modes and Tests to be Used for Linux Systems

Compiler	Build	Modes				Test Suites
		seq	mpi	omp	mixed	
Intel	Sequential	✓				All Regression-like
	MPI	✓	✓			All Suites, incl. V&V
	OpenMP	✓		✓		All Regression-like
	MPI-OpenMP	✓	✓	✓	✓	All Regression-like
gfortran	Sequential	✓				All Regression-like
	OpenMPI	✓	✓			All Regression-like
Portland	Sequential	✓				All Regression-like
	MPI	✓	✓			All Regression-like
Pathscale	Sequential	✓				All Regression-like

Linux

The default Linux templates for the regression-like test suites are generated on the Godzilla Linux cluster using Intel Fortran 12.1.5. The default build configuration—CONFIG='intel omp plot static_link'—is used for template generation. This version is executed sequentially (one thread) to create the templates. Godzilla consists of four DELL-815, 12-core AMD nodes running *Red Hat Enterprise Linux 5.9*.

Most Linux-based compiler/execution testing is conducted on the LANL Moonlight system. Moonlight is an ASC *Tri-Lab Capacity Computing Cluster* (TLCC2) [2] running the *Linux Chaos* [3] operating system. There are thirteen (13) test problems where the Moonlight results differ from the default templates. The differences are small and are attributed to the different system configurations.

After verifying that the default templates are tracked with the default build/mode on Godzilla, a set of Moonlight-based templates for these 13 problems is created to facilitate comparison with the other builds and execution modes on that system.

For the non-Intel compilers, the results from a sequential build of each—CONFIG='<compiler> plot'—are compared to the default templates (either the Godzilla or Moonlight templates, depending on the system used). From these results, a set of compiler-

specific templates are created for comparison to other builds and execution modes with that compiler. The results are summarized by compiler, build and execution mode below.

Intel Fortran v12.1.5 (Godzilla and Moonlight)

CONFIG='intel omp plot static_link' (the default)

Execution Mode: Sequential (the default for template creation)

Godzilla: All results track with zero differences (by definition).¹

Moonlight:

The following list of test problems show statistically small differences between Godzilla and Moonlight runs with the default build. The differences are most commonly round off differences although a few of these tests show a different number of random numbers used between Godzilla and Moonlight.

- FEATURES/DAWWG/DAWWG: gdv_b
- FEATURES/MODEL_DEV: inp035, inp038, inp043
- MCNPX_65: inp023
- MCNPX_EXTENDED/heavyions: inp71, inp71
- MCNPX_EXTENDED/test27b: inp05
- MCNPX_EXTENDED/testdndg: inp24
- MCNPX_EXTENDED/testincl: inp04, inp104, inp106, inp116

This indicates that the version of Linux (and packages installed) can affect results in approximately 1% of the tests. These 13 tests were rerun sequentially to create a set of Moonlight-tracking templates (for these tests only).

Execution Mode: Threaded

Threaded parallelism was tested on the Moonlight system using the Moonlight-tracking templates. The number of threads tested includes 2, 4, 8 and 16. Not all test problems actually run in parallel because some MCNP6 options force threading to be turned off.

With threading, most (1212 of 1227) of the output (OUTP) files show differences that consisting of additional messages, round-off differences, etc. However, only fifteen (15) of the tests (~%1) show tally (MCTAL) file differences. These few differences have been closely examined and the results are in statistical agreement with the default templates.

The *locations* and *number* of non-tracking tally results remain constant for different numbers of threads. The differences themselves vary slightly with thread count. The test suites that show tally-file differences when run threaded include:

Suite	mctal
ELECTRONS	4
FEATURES/DAWWG/DAWWG	1
PHVTR	4
MCNPX_65	1
MCNPX_EXTENDED/avr	1
MCNPX_EXTENDED/phys	2
MCNPX_EXTENDED/push	1

¹ **Note:** An executable built on Godzilla runs successfully on Moonlight and exhibits the same differences as and executable built on Moonlight itself. A Moonlight executable runs on Godzilla and tracks the default templates exactly.

CONFIG='intel plot seq'

The difference between this version and the OpenMP version run sequentially lies only in the external libraries: standard or thread-safe.

Execution Mode: Sequential

All regression-like suites run to completion (1227 tests). Of these fifty six (56) output files and thirty four (34) tally files show differences from the templates. These differences are commonly seen between a threaded version run sequentially and a sequential version and are similar in scope to the differences that arise from threaded execution.

CONFIG='intel openmpi omp plot'

This executable can be run in four modes: sequential, threaded, MPI and threaded MPI. Parallel executions show some tally-file differences. This is expected behavior for OpenMP, MPI or mixed parallelism as some physics in MCNP6 is not parallel-safe.

Mode: Sequential

No differences expected or seen.

Mode: Threaded

When run threaded only (NMPI=1, NTRD>1), there are twenty five (25) tally results that exhibit differences from the Intel/Moonlight templates. Essentially all (1221) of the output files show differences, which is expected in parallel runs.

Suite	outp	mctal
REGRESSION	144	1
ELECTRONS	94	4
PHTVR	209	4
FEATURES/DAWWG/DAWWG	45	2
FEATURES/MODEL_DEV	45	3
MCNPX_65	85	1
MCNPX_EXTENDED/heavyions	9	2
MCNPX_EXTENDED/push	21	2
MCNPX_EXTENDED/test27b	13	1
MCNPX_EXTENDED/testdndg	16	1
MCNPX_EXTENDED/testincl	44	4

Mode: MPI

When run with MPI only (NMPI=16, NTRD=1), there are forty four (44) tally results that exhibit differences from the Intel/Moonlight templates.

Suite	mctal
REGRESSION	2
ELECTRONS	4
MAG_FIELDS	1
PHTVR	4
FEATURES/DAWWG/DAWWG	1
FEATURES/MODEL_DEV	3
MCNPX_65	1
MCNPX_EXTENDED/avr	2

Unclassified

<u>Suite</u>	<u>mctal</u>
MCNPX_EXTENDED/phys	2
MCNPX_EXTENDED/push	2
MCNPX_EXTENDED/test27a	3
MCNPX_EXTENDED/test27c	1
MCNPX_EXTENDED/test27e	9
MCNPX_EXTENDED/testdndg	8
MCNPX_EXTENDED/testincl	1

Mode: MPI with Threads

When run with MPI and threads, one MPI task is the master task and the remainder are worker tasks (slaves). Each worker task uses threading for its portion of transport. For NMPI=5, NTRD=4, there is one master and four workers consisting of four threads each for a total of 17 processes. As the master is idle when all the workers are in transport mode, this is a viable configuration for running on a 16-core compute node.

With NMPI=3, NTRD=4, there are forty nine (49) tally files that exhibit differences from the Intel/Moonlight templates.

<u>Suite</u>	<u>mctal</u>
REGRESSION	2
ELECTRONS	4
MAG_FIELDS	1
PHTVR	4
FEATURES/DAWWG/DAWWG	2
FEATURES/MODEL_DEV	5
MCNPX_65	1
MCNPX_EXTENDED/heavyions	2
MCNPX_EXTENDED/push	2
MCNPX_EXTENDED/test27a	3
MCNPX_EXTENDED/test27b	1
MCNPX_EXTENDED/test27c	1
MCNPX_EXTENDED/test27e	8
MCNPX_EXTENDED/testdndg	8
MCNPX_EXTENDED/testincl	5

This version was also run with NMPI=3/NTRD=4 (3x4) and NMPI=9/NTRD=2 (9x2) with similar results.

CONFIG='intel openmpi plot'

This version has two execution modes: sequential and MPI parallel. All regression-like suites were run in both modes. This is not the standard build configuration for MPI build. The standard build that includes MPI also includes OpenMP. This build is tested only for completeness.

Mode: Sequential

The executable run in this mode is effectively the same as the 'intel plot seq' version, not having OpenMP. However, the tracking performance is slightly better than that of the sequential build. There are twenty four (24) tally-file differences across ten (10) suites.

<u>Suite</u>	<u>mctal</u>
REGRESSION	1
MAG_FIELDS	1

Unclassified

FEATURES/DAWWG/DAWWG	2
FEATURES/MODEL_DEV	3
MCNPX_EXTENDED/heavyions	3
MCNPX_EXTENDED/phys	1
MCNPX_EXTENDED/push	2
MCNPX_EXTENDED/test27e	2
MCNPX_EXTENDED/testincl	8
MCNPX_EXTENDED/zrecoil	1

Mode: MPI

With sixteen MPI processes (NMPI=16), there are sixty one (61) tally file differences.

Suite	mctal
REGRESSION	3
ELECTRONS	4
MAG_FIELDS	2
PHTVR	4
FEATURES/DAWWG/DAWWG	3
FEATURES/MODEL_DEV	5
MCNPX_65	1
MCNPX_EXTENDED/heavyions	3
MCNPX_EXTENDED/phys	3
MCNPX_EXTENDED/push	3
MCNPX_EXTENDED/test27a	3
MCNPX_EXTENDED/test27c	1
MCNPX_EXTENDED/test27e	9
MCNPX_EXTENDED/testdndg	8
MCNPX_EXTENDED/testincl	8
MCNPX_EXTENDED/zrecoil	1

gfortran v4.7.0

CONFIG='gfortran plot seq'

MCNP6 builds successfully with gfortran/gcc 4.7.0 on Moonlight.

Mode: Sequential

All regression-like suites run successfully.

One test, REGRESSION/inp124, is excluded for gfortran testing as it is known to fail with gfortran builds. This is a known issue and appears to be a compiler bug.

There are one hundred five (105) tally files (of 1226) that differ across the regression like suites.

Suite	mctal
REGRESSION	6
ELECTRONS	11
MAG_FIELDS	2
MUONS	4
PHTVR	27
FEATURES/DAWWG/DAWWG	11
FEATURES/UNC	1

Unclassified

<u>Suite</u>	<u>mctal</u>
FEATURES/MODEL_DEV	7
MCNPX_65	2
MCNPX_EXTENDED/avr	1
MCNPX_EXTENDED/heavyions	5
MCNPX_EXTENDED/phys	1
MCNPX_EXTENDED/push	1
MCNPX_EXTENDED/test27a	1
MCNPX_EXTENDED/test27b	1
MCNPX_EXTENDED/test27c	1
MCNPX_EXTENDED/test27d	2
MCNPX_EXTENDED/test27e	5
MCNPX_EXTENDED/testdndg	1
MCNPX_EXTENDED/testincl	13
MCNPX_EXTENDED/testxold	1
MCNPX_EXTENDED/zrecoil	1

These results were used to create a set of gfortran templates with which to compare MPI execution results

CONFIG='gfortran plot openmpi'

Mode: Sequential

Sequential execution of the gfortran-MPI executable tracks the templates created with the sequential version except for minor differences in four PostScript plot files in the MCNPX_EXTENDED suites: two in class/; two in testmesh/.

Mode: Parallel

Parallel execution with NMPI=16 exhibits twenty (20) tests with tally-file differences from the sequential gfortran templates.

<u>Suite</u>	<u>mctal</u>
ELECTRONS	3
MAG_FIELDS	1
FEATURES/MODEL_DEV	1
MCNPX_EXTENDED/heavyions	1
MCNPX_EXTENDED/phys	2
MCNPX_EXTENDED/test27a	3
MCNPX_EXTENDED/test27c	1
MCNPX_EXTENDED/test27e	7
MCNPX_EXTENDED/testincl	1

Portland Group v12.10

All regression-like test suites were run for all builds and execution modes for this compiler. The compiler used is pgi/12.10. The system used for this testing was Moonlight.

CONFIG='portland plot seq'

Building the sequential executable succeeds.

Mode: Sequential

All tests ran to completion. Results are compared to the Moonlight/Linux templates.

Unclassified

Over all suites, one hundred thirty seven (137) tests exhibit tally file differences. The number of output and tally file differences from the Intel/Moonlight templates is given in the following table.

Suite	outp	mctal
REGRESSION	12	9
ELECTRONS	13	9
MAG_FIELDS	8	2
MUONS	5	5
PHTVR	51	48
FEATURES/DAWWG/DAWWG	21	17
FEATURES/UNC	1	1
FEATURES/MODEL_DEV	7	7
MCNPX_65	6	4
MCNPX_EXTENDED/avr	1	1
MCNPX_EXTENDED/classvar	5	0
MCNPX_EXTENDED/heavyions	5	5
MCNPX_EXTENDED/mbody	1	0
MCNPX_EXTENDED/phys	4	1
MCNPX_EXTENDED/push	2	1
MCNPX_EXTENDED/test27a	1	1
MCNPX_EXTENDED/test27b	2	1
MCNPX_EXTENDED/test27c	1	1
MCNPX_EXTENDED/test27d	3	2
MCNPX_EXTENDED/test27e	7	5
MCNPX_EXTENDED/testdndg	2	2
MCNPX_EXTENDED/testincl	13	13
MCNPX_EXTENDED/testxold	1	1
MCNPX_EXTENDED/zrecoil	1	1

A set of PGI templates was created from this mode's results.

CONFIG='portland plot openmpi'

Building with OpenMPI was successful. Most output files will exhibit differences when NMPI>2 with any compiler. Such differences are expected and not reported here.

Mode: Sequential (NMPI=1)

All tests ran to completion. Across all tests, four tests shows differences in PostScript plot files generated by the test. All other templates are fully tracked, compared to the PGI templates created from the sequential build.

Mode: Parallel (NMPI>2)

All scheduled tests (1223) ran to completion with 1, 4, 8 and 16 MPI processes. There are some tests that exhibit tally-file differences for NMPI>2. The suites with tally-file differences are listed below with the number of such differences for 4, 8 and 16 processes.

Suite	NMPI=4	NMPI=8	NMPI=16
ELECTRONS	3	3	3
MAG_FIELDS	1	1	1
FEATURES/MODEL_DEV	1	1	1
MCNPX_EXTENDED/heavyions	1	0	1

Unclassified

MCNPX_EXTENDED/phys	0	2	2
MCNPX_EXTENDED/test27a	3	3	3
MCNPX_EXTENDED/test27c	1	1	1
MCNPX_EXTENDED/test27e	9	9	10
MCNPX_EXTENDED/testdndg	7	7	8

PathScale v4.0.12.1

CONFIG='pathscale plot seq'

MCNP6 builds successfully on Moonlight after minor change to Linux.gcf. Compared to the Moonlight/Intel templates, there are one hundred thirty seven (137) tests that exhibit differences in the output and/or tally files. Results are compared to the Moonlight-tracking Intel templates.

Mode: Sequential

All regression-like suites run successfully. There are 103 tests with tally-file differences compared to the Intel templates (Godzilla and Moonlight). The following list provides the number of tests with output and/or tally file differences in each test suite, 138 and 103, respectively.

Suite	outp	mctal
REGRESSION	11	6
ELECTRONS	16	14
MAG_FIELDS	8	1
MUONS	4	4
PHTVR	29	27
FEATURES/DAWWG/DAWWG	10	10
FEATURES/UNC	1	0
FEATURES/MODEL_DEV	5	4
MCNPX_65	6	4
MCNPX_EXTENDED/avr	1	1
MCNPX_EXTENDED/classvar	5	0
MCNPX_EXTENDED/heavyions	5	4
MCNPX_EXTENDED/mbody	1	0
MCNPX_EXTENDED/phys	4	1
MCNPX_EXTENDED/push	2	1
MCNPX_EXTENDED/test27a	1	1
MCNPX_EXTENDED/test27b	2	1
MCNPX_EXTENDED/test27c	1	1
MCNPX_EXTENDED/test27d	3	2
MCNPX_EXTENDED/test27e	8	6
MCNPX_EXTENDED/testdndg	1	1
MCNPX_EXTENDED/testincl	12	12
MCNPX_EXTENDED/zrecoil	1	1

These results were used to create a set of PathScale templates for comparison to other PathScale builds and execution modes.

CONFIG='pathscale plot omp

Mode: Sequential

Out of 1227 regression tests overall, one fails to complete with the PathScale OpenMP executable: REGRESSION/inp125. This test hangs with the threaded executable even when run with a single thread.

Among the remaining (1226) tests, the OpenMP enabled version exhibits twenty-four (24) tests exhibiting output file differences compared to the sequential-build PathScale templates. Among these, there are eleven (11) that also exhibit tally file differences. The following list provides the number of tests with OUTP and/or MCTAL differences for each test suite in which differences occur.

Suite	outp	mctal
REGRESSION	6	0
ELECTRONS	3	3
PHTVR	2	0
FEATURES/MODEL_DEV	2	1
MCNPX_EXTENDED/classvar	1	0
MCNPX_EXTENDED/phys	1	0
MCNPX_EXTENDED/push	1	0
MCNPX_EXTENDED/test27c	1	1
MCNPX_EXTENDED/test27e	3	2
MCNPX_EXTENDED/testdndg	4	4

Mode: Threaded

This mode is not supported by the LANL MCNP developers.

The PathScale OpenMP executable does not successfully complete the regression-like test suites. The issues with threading and Pathscale are test dependent. It is worthwhile to note that many tests do run properly with threading, including the VALIDATION_CRITICALITY and VALIDATION_CRIT_EXPANDED test suites (150 tests overall). These two suites ran successfully and efficiently with sixteen (16) threads on Moonlight.

Windows

Testing on Windows was performed on a virtual system named Vickers. The actual operating system is named Windows Server 2008 R2 Standard – SP1. This system has a variable number of AMD-64bit CPUs. Compilation and execution was performed using the Cygwin Linux emulator. Templates were created on a system using Windows 7 and Intel 12.1.5.

Testing was performed on other Windows-OS versions (XP, Vista, 7) for basic installation from the distribution and execution of the regression tests. That testing is reported elsewhere.

Intel

The Intel compiler version used for testing on Vickers is Intel Parallel Studio XE 2011, The Intel FORTRAN version installed is 12.1.7.

CONFIG='intel msvc omp plot'

Mode: Sequential (the default for template creation)

This is the mode used to create the Windows templates and this execution mode is expected to track the templates exactly, which is found to be the case.

Mode: Threaded

Run with NTRD=2 (the number of cores available on Vickers), sixteen (16) tests exhibit tally file differences.

Suite	outp	mctal
FEATURES/DAWWG/DAWWG	1	1
FEATURES/MODEL_DEV	3	3
MCNPX_EXTENDED/heavyions	4	4
MCNPX_EXTENDED/test27b	1	1
MCNPX_EXTENDED/test27e	2	2
MCNPX_EXTENDED/testdndg	1	1
MCNPX_EXTENDED/testincl	4	4

gfortran

The GNU compiler version used for testing on Vickers is the GCC package 4.7.2, which provides both gfortran and gcc.

CONFIG='gfortran plot seq'

Mode: Sequential

Suite	outp	mctal
REGRESSION	32	16
ELECTRONS	19	13
MAG_FIELDS	7	2
MUONS	6	6
PHOTONS	4	4
PHTVR	86	70
FEATURES/DAWWG/DAWWG	14	13
FEATURES/UNC	7	7
FEATURES/MODEL_DEV	25	25
FEATURES/POINT_DETECTORS	29	29
FEATURES/FMESH_INC	5	5
MCNPX_65	18	12
MCNPX_EXTENDED/avr	3	2
MCNPX_EXTENDED/classgeom	4	4
MCNPX_EXTENDED/heavyions	6	6
MCNPX_EXTENDED/phys	6	3
MCNPX_EXTENDED/push	5	3
MCNPX_EXTENDED/test27a	1	1
MCNPX_EXTENDED/test27b	3	2

Unclassified

<u>Suite</u>	<u>outp</u>	<u>mctal</u>
MCNPX_EXTENDED/test27c	2	2
MCNPX_EXTENDED/test27d	10	10
MCNPX_EXTENDED/test27e	13	11
MCNPX_EXTENDED/testburn	6	3
MCNPX_EXTENDED/testdndg	2	2
MCNPX_EXTENDED/testincl	33	31
MCNPX_EXTENDED/testmcpn	2	1
MCNPX_EXTENDED/testmesh	4	2
MCNPX_EXTENDED/testxold	8	6
MCNPX_EXTENDED/zrecoil	3	1

MacOS (aka: Darwin)

Intel version 12.0.3

CONFIG='intel omp plot'

MCNP6 builds successfully with Intel/12.0.3 on the MacOS system, BlackJack. This is an eight-core system running MacOS 10.8.0. As the differences between the MacOS Intel version and the default (Linux) version are small, no set of templates was generated for comparison of modes on the MacOS.

Mode: Sequential

All regression-like test suites run successfully. There are eighteen (18) tests that exhibit tally-file differences from the Linux (Gozdzilla) template set. The suites with output or tally file differences are listed in the following table.

<u>Suite</u>	<u>outp</u>	<u>mctal</u>
REGRESSION	2	0
ELECTRONS	1	0
PHTVR	3	0
FEATURES/DAWWG/DAWWG	1	1
FEATURES/MODEL_DEV	3	3
MCNPX_65	1	0
MCNPX_EXTENDED/classvar	5	0
MCNPX_EXTENDED/heavyions	2	2
MCNPX_EXTENDED/test27b	1	1
MCNPX_EXTENDED/test27d	2	2
MCNPX_EXTENDED/test27e	4	4
MCNPX_EXTENDED/testdndg	1	1
MCNPX_EXTENDED/testincl	4	4

Mode: Threaded

When executed with four (4) threads, all 1227 tests run to completion. Overall, there are twenty-four (24) tally files that differ from the Linux (Godzilla) templates – six (6) more than the sequential run. The suites with tally file differences are listed in the following table.

<u>Suite</u>	<u>mctal</u>
ELECTRONS	3
FEATURES/DAWWG/DAWWG	1

Unclassified

FEATURES/MODEL_DEV	3
MCNPX_EXTENDED/avr	1
MCNPX_EXTENDED/heavyions	2
MCNPX_EXTENDED/phys	2
MCNPX_EXTENDED/test27b	1
MCNPX_EXTENDED/test27d	2
MCNPX_EXTENDED/test27e	4
MCNPX_EXTENDED/testdndg	1
MCNPX_EXTENDED/testincl	4

gfortran version 4.7.3

CONFIG='gfortran plot seq'

Mode: Sequential

The sequential gfortran build exhibits 142 tests with differences in output files from the Linux (Godzilla) templates. Of these, 114 show tally-file differences. The suites with output and/or tally differences are listed in the following table.

Suite	outp	mctal
REGRESSION	12	6
ELECTRONS	16	14
MAG_FIELDS	8	2
MUONS	4	4
PHTVR	29	27
FEATURES/DAWWG/DAWWG	11	11
FEATURES/UNC	1	1
FEATURES/MODEL_DEV	8	7
MCNPX_65	4	2
MCNPX_EXTENDED/avr	1	1
MCNPX_EXTENDED/classvar	5	0
MCNPX_EXTENDED/heavyions	5	5
MCNPX_EXTENDED/mbody	1	0
MCNPX_EXTENDED/phys	4	1
MCNPX_EXTENDED/push	2	1
MCNPX_EXTENDED/test27a	1	1
MCNPX_EXTENDED/test27b	1	1
MCNPX_EXTENDED/test27c	1	1
MCNPX_EXTENDED/test27d	5	4
MCNPX_EXTENDED/test27e	10	8
MCNPX_EXTENDED/testdndg	2	2
MCNPX_EXTENDED/testincl	13	13
MCNPX_EXTENDED/testxold	2	1
MCNPX_EXTENDED/zrecoil	1	1

References

- [1] “MCNP6 Release 1.0: Verification & Validation Testing (HTML),” L. J. Cox, LA-UR 13-00019, included in the MCNP6 Release 1.0 documentation collection on Disk 1.

Unclassified

- [2] “Tri-Lab Linux Capacity Cluster – Unifying Computing Across the NNSA Defense Complex”, https://asc.llnl.gov/computing_resources/tlcc/tlcc2_2012.pdf
- [3] Clustered High Availability Operating System (CHAOS) Linux, <https://computing.llnl.gov/linux/projects.html>