

LA-UR-14-24330

Approved for public release; distribution is unlimited.

Title: MCNP(TM) Release 6.1.1 beta: Creating and Testing the Code
Distribution

Author(s): Cox, Lawrence J.
Casswell, Laura

Intended for: Distribution thru RSSIC with the MCNP6 Release 1.1 beta Package
Report
Web

Issued: 2014-06-12

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.

MCNP™ Release 6.1.1 beta: Creating and Testing the Code Distribution

¹L. J. Cox, ²L. F. Casswell

¹XCP-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™ 1.1 beta through RSICC at ORNL. It addresses tests on supported operating systems (Linux, MacOSX, Windows) with the supported compilers (Intel, Portland Group 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, MCNPX and MCNP6) and the collection of reference documents.

The MCNP6 1.1 beta Portion of the RSICC Distribution

MCNP6 1.1 beta is being prepared for release through the Radiation Safety Information Computational Center (RSICC). The RSICC distribution of MCNP6 1.1 beta will contain the following components:

- MCNP611: 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
- MCNP6: with or without source code

This paper addresses the generation and testing of the MCNP611 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 has continued on the main CVS trunk of the repository. Only reviewed and approved fixes and features made to the main CVS trunk were merged into the release branch as needed. The name of the branch is **MCNP6_Rel_6-1-1beta**, which is case-sensitive. There is a tag on the branch, **MCNP6_Rel_6-1-1beta** which points 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), some obsolete source code directories were removed from the release branch. Other than these changes, any source differences are due to changes made to the main CVS trunk that have not been approved and migrated to the release branch.

Exporting for Release

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

```
> cvs export -r MCNP6_Rel_6-1-1beta [-d <dirname>] MCNP6
> cd <dirname>
> Utilities/purge_export purge
```

Unclassified

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

Test Plan

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

Build Configurations and Operating Systems

MCNP6 1.1 beta is supported on Linux, MacOS and Windows. The compilers that are supported for building MCNP6 1.1 beta 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		
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 and test system (CBTS), which is run daily on the latest version from 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
PHVTR	Pulse-height tally variance reduction tests
FEATURES/ CGM COSMIC DAWWG DELAYED_PARTICLES FMESH_INC LIGHT_IONS MODEL_DEV PLOT_XS POINT_DETECTORS UNC	Ten (10) test suites for special features
MCNPX_65 MCNPX_EXTENDED	The MCNPX regression suite plus twenty-two (22) additional test suites 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 1408 problems overall. The REGRESSION suite by itself contains 167 problems. The listed V&V test suites contain 535 problems.

During MPI execution, there are four (of 1408) 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 compiler related issues in any execution mode on Linux with gfortran-built executables.

OS/Compiler Test Results

This section summarizes 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.

Currently only Intel and gfortran are currently supported. Although compilations with the Portland Group compiler have been performed on Linux, the test suite results for the various configurations of OpenMP and OpenMPI have not been performed. 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 are 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 compilers 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 1.1 beta 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 and gfortran compilers work well with OpenMPI. 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

Linux

The default Linux templates for the regression-like test suites are generated on the Moonlight Linux cluster using Intel Fortran 12.1.5. The default build configuration—`CONFIG='intel omp plot'`—is used for template generation. This version is executed sequentially (one thread) to create the templates. Moonlight is an ASC *Tri-Lab Capacity Computing Cluster* (TLCC2) [2] running the *Linux Chaos* [3] operating system.

For the non-Intel compilers, the results from a sequential build of each—`CONFIG='<compiler> plot'`—are compared to the default. 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 (Moonlight)

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

Execution Mode: Sequential (the default for template creation)

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

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 1.1 beta options force threading to be turned off.

With threading, most (1400 of 1408) of the output (OUTP) files show differences that consist of additional messages, round-off differences, etc. However, only seven (7) of the tests (~%0.5) 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* of non-tracking tally results remain constant for different numbers of threads. The number of non-tracking results and the differences themselves vary slightly with thread count. The test suites that show tally-file differences when run threaded include:

Suite	mctal
REGRESSION	1
MCNPX_65	1
MCNPX_EXTENDED/avr	1
MCNPX_EXTENDED/phys	2
MCNPX_EXTENDED/tstdng	2

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.

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.

Unclassified

Execution Mode: Sequential

All regression-like suites run to completion (1408 tests). Of these eighty one (81) output files and forty two (42) 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 two (2) tally results that exhibit differences from the Intel/Moonlight templates. Essentially all (1408) of the output files show differences, which is expected in parallel runs.

<u>Suite</u>	<u>outp</u>	<u>mctal</u>
MCNP_EXTENDED/testdng	16	2

Mode: MPI

When run with MPI only (NMPI=16, NTRD=1), there are seventeen (17) tally results that exhibit differences from the Intel/Moonlight templates.

<u>Suite</u>	<u>mctal</u>
ELECTRONS	4
FEATURES/MODEL_DEV	1
CGM	3
MCNPX_EXTENDED/phys	2
MCNPX_EXTENDED/test27e	5
MCNPX_EXTENDED/testdndg	1
MCNPX_EXTENDED/testincl	1

Mode: MPI with Threads

When run with MPI and threads, one MPI task is the master task and the remaining MPI tasks 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 fifteen (15) tally files that exhibit differences from the Intel/Moonlight templates.

Unclassified

Suite	mctal
ELECTRONS	4
FEATURES/MODEL_DEV	1
FEATURES/CGM	3
MCNPX_EXTENDED/test27e	4
MCNPX_EXTENDED/testdndg	3

This version was also run with NMPI=5/NTRD=4 (5x4) and NMPI=9/NTRD=2 (9x2) with similar results but with twenty one (21) tally files that exhibit differences from the Intel/Moonlight templates.

Suite	mctal
ELECTRONS	4
FEATURES/MODEL_DEV	1
FEATURES/CGM	4
FEATURES/COSMIC	1
MCNPX_EXTENDED/phys	2
MCNPX_EXTENDED/test27e	5
MCNPX_EXTENDED/testdndg	3
MCNPX_EXTENDED/testincl	1

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 thirty two (32) tally-file differences across twelve (12) suites.

Suite	mctal
REGRESSION	1
MAG_FIELDS	1
PHOTONS	2
FEATURES/DAWWG/DAWWG	2
FEATURES/MODEL_DEV	2
FEATURES/COSMIC	8
MCNPX_EXTENDED/heavyions	3
MCNPX_EXTENDED/phys	1
MCNPX_EXTENDED/push	2
MCNPX_EXTENDED/test27e	1
MCNPX_EXTENDED/testincl	8
MCNPX_EXTENDED/zrecoil	1

Unclassified

Mode: MPI

With sixteen MPI processes (NMPI=16), there are forty six (46) tally file differences.

Suite	mctal
REGRESSION	1
ELECTRONS	4
MAG_FIELDS	1
PHOTONS	2
FEATURES/DAWWG/DAWWG	2
FEATURES/MODEL_DEV	2
FEATURES/CGM	3
FEATURES/COSMIC	8
MCNPX_EXTENDED/heavyions	3
MCNPX_EXTENDED/phys	3
MCNPX_EXTENDED/push	1
MCNPX_EXTENDED/test27e	6
MCNPX_EXTENDED/testdndg	1
MCNPX_EXTENDED/testincl	8
MCNPX_EXTENDED/zrecoil	1

gfortran v4.7.0 (Moonlight)

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 (125) tally files (of 1408) that differ across the regression like suites.

Suite	mctal
REGRESSION	10
ELECTRONS	15
MAG_FIELDS	2
MUONS	4
PHOTONS	2
PHTVR	27
FEATURES/DAWWG/DAWWG	11
FEATURES/UNC	1
FEATURES/MODEL_DEV	13
FEATURES/LIGHT_IONS	3
MCNPX_65	2
MCNPX_EXTENDED/avr	1
MCNPX_EXTENDED/heavyions	5
MCNPX_EXTENDED/phys	1

Unclassified

<u>Suite</u>	<u>mctal</u>
MCNPX_EXTENDED/push	1
MCNPX_EXTENDED/test27a	1
MCNPX_EXTENDED/test27b	1
MCNPX_EXTENDED/test27c	1
MCNPX_EXTENDED/test27d	3
MCNPX_EXTENDED/test27e	4
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 one hundred thirty four (134) tests with tally-file differences from the sequential gfortran templates.

<u>Suite</u>	<u>mctal</u>
REGRESSION	10
ELECTRONS	15
MAG_FIELDS	2
MUONS	4
PHOTONS	2
PHTVR	27
FEATURES/DAWWG/DAWWG	4
FEATURES/MODEL_DEV	13
FEATURES/CGM	3
FEATURES/COSMIC	8
FEATURES/LIGHT_IONS	3
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	3
MCNPX_EXTENDED/test27e	4
MCNPX_EXTENDED/testdndg	1

Unclassified

<u>Suite</u>	<u>mctal</u>
MCNPX_EXTENDED/testincl	13
MCNPX_EXTENDED/testxold	1
MCNPX_EXTENDED/zrecoil	1

Portland Group v12.10 (Moonlight)

All regression-like test suites were run for a sequential build 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

1405 of 1408 tests ran to completion. Results are compared to the Moonlight/Linux templates.

Over all suites, one hundred fifty eight (158) tests exhibit tally file differences. The number of output and tally file differences from the Intel/Moonlight templates is given in the following table.

<u>Suite</u>	<u>outp</u>	<u>mctal</u>
REGRESSION	18	12
ELECTRONS	17	13
MAG_FIELDS	7	2
MUONS	5	5
PHOTONS	2	2
PHTVR	51	48
FEATURES/DAWWG/DAWWG	16	11
FEATURES/UNC	1	1
FEATURES/MODEL_DEV	13	13
FEATURES/CGM	1	1
FEATURES/COSMIC	8	8
FEATURES/PLOT_XS	17	0
FEATURES/LIGHT_IONS	3	3
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	4	2
MCNPX_EXTENDED/test27e	6	5
MCNPX_EXTENDED/testdndg	2	2
MCNPX_EXTENDED/testincl	13	13
MCNPX_EXTENDED/testxold	1	1
MCNPX_EXTENDED/zrecoil	1	1

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

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 1408 regression-like test suites run successfully. There are forty one (41) tests that exhibit tally-file differences from the Linux (Moonlight) template set. The suites with output or tally file differences are listed in the following table.

Suite	outp	mctal
REGRESSION	5	1
ELECTRONS	4	4

Unclassified

Suite	outp	mctal
MAG_FIELDS	3	1
PHOTONS	2	2
PHVTR	2	0
FEATURES/DAWWG/DAWWG	2	2
FEATURES/MODEL_DEV	2	2
FEATURES/COSMIC	8	8
FEATURES/PLOT_XS	14	0
MCNPX_65	1	0
MCNPX_EXTENDED/classvar	5	0
MCNPX_EXTENDED/heavyions	3	3
MCNPX_EXTENDED/mbody	1	0
MCNPX_EXTENDED/phys	2	1
MCNPX_EXTENDED/push	2	1
MCNPX_EXTENDED/test27d	2	2
MCNPX_EXTENDED/test27e	4	4
MCNPX_EXTENDED/testdndg	1	1
MCNPX_EXTENDED/testincl	8	8
MCNPX_EXTENDED/zrecoil	1	1

References

- [1] “Verification of MCNP6.1 and MCNP6.1.1 for Criticality Safety Applications,” F.B. Brown, B.C. Kiedrowski, J.S. Bull, LA-UR 14-22480, included in the MCNP6 Release 1.1beta documentation collection on Disk 4.
- [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>