LA-UR-25-23548

Approved for public release; distribution is unlimited.

Title: MCNP® Code Version 6.3.1 Release Notes

Author(s): Rising, Michael Evan; Armstrong, Jerawan Chudoung; Bolding, Simon R.; Bull, Jeffrey S.; Clark, Alexander R.; Frederick, Cole Spencer; Giron, Jesse Frank; Grieve, Avery S.; Haeck, Wim; Jones, Fred Burke; Josey, Colin James; Kulesza, Joel A.; Lively, Michael Aaron; Swaminarayan, Sriram; Sweezy, Jeremy Ed; Weaver, Colin Andrew; Zukaitis, Anthony J.

Intended for: Report

Issued: 2025-04-28 (rev.1)

Los Alamos
NATIONAL LABORATORY

LA-UR-25-23548, Rev. 1

MCNP®

Code Version 6.3.1

Release Notes

Document compiled from `git` hash `754414f91f` on April 22, 2025.

This document cannot be made Section-508 compliant (accessible to persons with disabilities) at this time. If you need help accessing this document, please email mcnp_help@lanl.gov.

Los Alamos
NATIONAL LABORATORY

MCNP® Code Version 6.3.1 Release Notes

LA-UR-25-23548, Rev. 1

April 22, 2025

Los Alamos National Laboratory

Michael E. Rising
Jerawan C. Armstrong
Simon R. Bolding
Jeffrey S. Bull
Alexander R. Clark
Cole S. Frederick
Jesse F. Giron
Avery S. Grieve
Wim Haeck
Fred B. Jones
Colin Josey
Joel A. Kulesza
Michael A. Lively
Sriram Swaminarayan
Jeremy E. Sweezy
Colin A. Weaver
Anthony J. Zukaitis

Contents

The Monte Carlo N-Particle® (MCNP®) code is a general-purpose, continuous-energy, generalized-geometry, time-dependent, radiation transport code developed by the MCNP development team. MCNP calculations provide predictive capabilities that can replace expensive or impossible-to-perform experiments. Specific application problems include simulations of experimental diagnostics, intrinsic radiation, radiation detection and measurement, criticality safety, nuclear threat reduction and response, radiation health protection, nuclear weapons effects, and nuclear forensics. This MCNP code, version 6.3.1, follows the MCNP6.3.0 version [1].

Since the release of MCNP6.3.0, a variety of bug fixes and code enhancements have been completed for MCNP6.3.1. A few new features have also been added to this release to support both ongoing research and the release of the latest ENDF/B-VIII.1 nuclear data library.

The MCNP code, version 6.3.1, theory and user input information is documented in MCNP® Code Version 6.3.1 Theory & User Manual [2], the build guidance for various platforms is documented in MCNP® Code Version 6.3.1 Build Guide [3], and the verification and validation testing for various application benchmark test suites is documented in MCNP® Code Version 6.3.1 Verification & Validation Testing [4].

1 Deprecation and Removal of Old Features and Functionality

Features marked for deprecation are considered obsolete by the MCNP development team and are eligible for removal in future versions of the MCNP code. It is important to recognize, understand, and test new features that make old features deprecated. Deprecated features should not be relied upon because they may be removed in the next release of the code. Section 1.1 contains a listing of deprecated features. By removing these features in the future, the MCNP development team can reduce its maintenance burden and instead focus on providing new features (and code releases) more quickly.

While it is preferable to first mark features and capabilities as deprecated before they are fully removed from a future distribution, it is occasionally necessary to fully remove a feature without an immediate side-by-side modern replacement. Because of the desire to avoid this situation in as many circumstances as possible, few features are removed in this version of the MCNP code. Section 1.2 contains a listing of removed features. If there are questions or comments regarding removed features, please contact the MCNP team at mcnp_help@lanl.gov.

1.1 Deprecated Features

No new features have been deprecated beyond those listed in the MCNP6.3.0 version release notes [§1.1 of 1].

1.2 Removed Features

MCNP-53604	The `multitrack` keyword from the `EMBED` card is removed. This undocumented feature is not practical for real applications because of additional checking and subsequent computational slowdowns during transport. The variable for storing the `multitrack` value is removed from the restart file and the restart file version is updated.

Target	ZAID Format	Named Format
^{238}U	`92238`	`U-238`
$^{\text{nat}}$C	`6000`	`C-0`
$^{110\text{m}}$Ag	`47510`	`Ag-110m1`
$^{242\text{m}}$Am	`95242`	`Am-242m1`

Table 1: Example Comparisons of ZAIDs to Named Identifiers

2 New Features and Extensions

2.1 Density Type Required for LNK3DNT Embedded Meshes [MCNP-53464]

A new required `dentype` keyword is added to the `EMBED` card for LNK3DNT files. Densities in a LNK3DNT file can be mass or atom densities, but all MCNP6 versions prior to and including MCNP6.3.0 can only process a LNK3DNT file that contains mass densities. The `dentype` = `mass` and `dentype` = `atom` keyword-value choices are used to specify mass and atom densities in a LNK3DNT file in $\text{g} \cdot \text{cm}^{-3}$ and $\text{b}^{-1} \cdot \text{cm}^{-1}$, respectively. Note that densities in a LNK3DNT file may be mass or atom, but cannot be both. In previous code versions (prior to and including MCNP6.3.0), atom densities in mesh geometries are computed on-the-fly during the transport step and the mass densities are written into a restart file for a continue run. In the current code version, the mass densities read from a LNK3DNT file are converted to atom densities during the input processing step and atom densities are written into the restart file for a continue run.

2.2 Reworked Target Identifiers and Data Loading [MCNP-53601]

In previous releases, data library names were limited to 10-character strings with a 2-digit numeric library identifier. With the many decades of released data, the LANL nuclear data and MCNP development teams have run out of unique 2-digit identifiers to use for future libraries. In this release, this limitation is removed. Library names can now be up to 32 characters long. Numeric library identifiers are still supported without change, but alphanumeric library names are also allowed so long as the library and physics identifiers are first separated by a dot (e.g., `92238.8100c` and `92238.lib80x.c` are now both allowed).

Nuclide identifiers have also been reworked. Previously, the code only supported ZAIDs. Now, conventional named identifiers can be used. The ZAID format and the named format columns are now considered equivalent with examples shown in Table 1. Note that the new format does not swap ^{242}Am and $^{242\text{m}}$Am as has been done historically. Support for this new format has been propagated throughout the code, and with few exceptions noted in the manual (like input on `FU` cards used with `FT`), are available on all inputs.

In response to this change, the data loading method has been reworked. Previously, if `92238.00c` was listed on a material card, only an exact match of `92238.00c` would load. Now, the target identifier is decomposed into its Z, A, and metastable state S and matched by those values. As such, if `U-238` is listed on a material card, the first library that provides $(Z, A, S) = (92, 238, 0)$ in the **`xsdir`** is loaded, which could be `92238` or `U-238`. If `Am-242m1.80c` is listed, the library with $(Z, A, S) = (92, 242, 1)$ with library identifier `80` and physics identifier `c` is loaded, which could be `95242.80c` or `Am-242m1.80c`. Note that since `95242.80c` and `Am-242m1.80c` are considered identical, only one should be in the **`xsdir`** at any time. This approach allows users to continue to use ZAIDs with new data specified using names or to use names with old data specified with ZAIDs.

There was substantial change on both the terminal output and **outp** files. Tables that historically output ZAIDs will now output named identifiers. This includes `BURN` output and `KSEN` output. Tables that contain a data table name adjust column width to handle the extended library name format. This includes the event log, the physical constant table (Table 98), $S(\alpha, \beta)$ assignment (Table 102), and neutron/photon activity (Table 140). Table 100, which contains information on loaded cross-section libraries, is entirely reworked to better present available information. The A Compact ENDF (ACE) file formats have not been touched and will continue to contain ZAIDs.

In the process of implementing this feature, a pass was made at making the code's handling of metastable targets more consistent. This has fixed the following bugs:

- The random number sequence for unresolved resonances in neutron tabular data was shared between metastable and non-metastable targets.
- Tally tagging (`FT` `tag`) did not distinguish between metastable and non-metastable targets. Both, if present, were summed together.
- The `DBRC` card would not apply to metastable nuclides aside from $^{242\mathrm{m}}$Am.

2.3 Added SFC64 as Random Number Generator 8 [MCNP-53671]

A new generator is available when using `gen = 8` on `RAND` [5]. This generator has a substantially larger state space, evaluates faster, and emits better-distributed random bits than the current linear congruential generators. It also has two unique features relative to the current generators. First, it always provides a unique sequence of 2^{64} random numbers for each particle history, eliminating the need to set a `stride`. Second, when two simulations are run with different seeds, no two particles will ever share random number sequences, preventing correlations due to a poor choice of seed.

3 Code Enhancements and Notable Improvements

3.1 Updates to the Qt-based Plotter [MCNP-53270]

In MCNP6.3.0 a new Qt-based geometry, tally, and cross section plotter was released as a technology preview to replace the legacy X11 interface to the MCNP plotter. In this MCNP6.3.1 release the new Qt plotter is improved with a variety of upgrades to make the transition to the new plotter easier, including

- Support for Qt6
- Corrected PostScript font specifier
- Set onscreen font consistently
- Many minor tweaks to the GUI, including
  - font scaling
  - automatic font sizing
  - menu item positioning
  - per-OS font selection
  - window on-resize redraw enhancement

- Support for small pixel dimension screens, relevant for screens with magnification or in presentation mode
- Better scaling of the interface elements

It is essential that the community transition to and provide feedback to mcnp_help@lanl.gov on the new Qt-based plotter as this will be replacing the X11 plotter in the next release of the MCNP code.

3.2 Contributions to Next-Event Estimators from Neutron Inelastic Scattering [MCNP-53609]

An issue has been fixed where the lower quadratic root for the outgoing energy of neutrons from neutron inelastic scattering contributions to next-event estimators (point detectors, ring detectors, image detectors, and DXTRAN) had been previously ignored in all previous versions of MCNP and MCNPX software [6]. For next-event estimators, the outgoing neutron energy in the laboratory frame is determined from the quadratic equation,

$$E'_{\pm} = \frac{E}{(A+1)^2}\left[\mu_{lab} \pm \sqrt{\mu_{lab} + \frac{E'_{cm}}{E}(A+1)^2 - 1}\right]^2 .$$

Prior to this change, only the upper root of this quadratic equation was considerd on the advice from Carter and Cashwell [7] "... the lower root E'_{-} can usually be ignored without introducting appreciable error." However, ignoring the lower root causes next-event estimators to differ from non-deterministic estimators, such as track-length or surface crossing estimators. This change will change the random number sequence of problems containing next-event estimators. Changes to next-event estimators will be most significant for low-energy contributions to next-event estimators for neutron inelastic scattering on light nuclei.

3.3 Fixed Cinder Gamma-line File [MCNP-53667]

A newly processed and revised **cindergl.dat** file, named **cindergl_v3.dat**, is included to fix errors in the gamma lines loaded when using the `ACT` card. The previous version 2 of this file, although named **cindergl.dat** in the MCNP6.3.0 release, should be replaced with the new version 3 of this file when using the MCNP6.3.0 code. Note that the coding in MCNP6.3.1 has been updated to load the **cindergl_v3.dat**-named file. If a user would like to use a version of the code earlier than MCNP6.3.1, the **cindergl.dat** should be replaced by the **cindergl_v3.dat** data file in their data installation. Likewise, if a user would like to use MCNP6.3.1 with the version 2 file, the **cindergl_v3.dat** can be temporarily replaced by the **cindergl_v2.dat** file to identify simulation differences with the fixed data file.

4 Performance

The primary focus of this MCNP6.3.1 release has been on fixing bugs and improving upon the new MCNP6.3.0 code features. A few items listed in the code enhancements in §A.2, such as the modern C++ pseudo-random number generator implementation (see MCNP-53508) and the upgrades within unstructured mesh coding (see MCNP-53665), may provide improved calculation performance for certain applications.

5 Verification and Validation (V&V)

The verification and validation testing for various application benchmark test suites is documented in MCNP® Code Version 6.3.1 Verification & Validation Testing [4]. In this separate V&V document, the entirety of the MCNP6.3.1 simulation results are given alongside either the analytical or experimental results for the verification or validation test suites, respectively.

5.1 New Electron Stopping Power Validation

In comparison to V&V efforts associated with the previous MCNP6.3.0 code release, a new electron stopping power validation test suite is now included. This test suite includes comparisons of single event electron stopping powers, computed using `PTRAC` outputs, to semi-empirical data computed from experimental electron energy loss functions measurements. See the MCNP® Code Version 6.3.1 Verification & Validation Testing [4] and the original validation work by Lively et al. [8] for further information on this new validation test suite.

5.2 Summary of V&V Results using MCNP6.3.1

When comparing the calculated results of MCNP6.3.0 to those of MCNP6.3.1, there are some minor result updates for certain calculations. All of the changes in results are expected with justification given in the following sections.

In summary, the results indicate that MCNP6.3.1 is as accurate as MCNP6.3.0 for the range of problems tested. Both the MCNP6.3.0 results and the MCNP6.3.1 results are saved for reference within the **vnvstats** framework described further in §7.5.4. The few changes between MCNP6.3.0 and MCNP6.3.1 that caused differences in simulated V&V test results are discussed in the sections to follow.

5.2.1 Update to MCNP6.3.0 Reference Values

Several results published with the MCNP6.3.0 release were inadvertantly copied from the wrong source of result files. Fortunately, the impact is extremely minor and should not change the results of any existing analysis or usage of the previously published results. In this release, the following MCNP6.3.0 reference results are updated:

- validation/crit_expanded/references/MCNP630_ENDF71, case **u233-met-fast-005-case-1**
- verification/keff/references/MCNP630, cases **ce01**, **ce17**, **ce19**, **ce29**, **ce38**, **ce40**, **mg01**, **mg31**, and **mg40**

In all cases, only the reference k-effective standard deviations are updated, by less than 2×10^{-9} in all cases, to match results produced by the MCNP6.3.0 binaries. All of the mean values remain unchanged from the previously published results.

5.2.2 Changes based on fixes to MCNP6.3.1

The only bugfix in MCNP6.3.1 that has a direct impact on the V&V test results is the next-event estimator inelastic scattering sampling improvement discussed in §3.2. The changes made to the code to improve the inelastic scattering contributions to next-event estimators effect the pulsed sphere validation test suite. While this fix primarily impacts low-energy contributions to next-event estimators for neutron inelastic scattering on light nuclei, the existing pulsed sphere validation test results in **vnvstats** are different when using MCNP6.3.1 primarily because of a change in the random number sequence. In summary, all of the pulsed sphere results have changed, regardless of what nuclear data are being used, but all appear to only change within the statistical uncertainties of the tally mean values.

5.2.3 Changes based on calculational approach

In the time during the MCNP6.3.1 release cycle, a new high performance computing (HPC) platform at LANL was deployed to replace the old HPC platform in use during the MCNP6.3.0 release. To make use of the new machine to efficiently run the V&V test suites, the parallelization strategy is changed for some test suites run with MCNP6.3.1.

Even though the parallelization strategy used for most test suites is changed for this new HPC machine, the only test suite results impacted by the parallelism changes is validation/rossi. This test suite calculates the Rossi-α and other reactor kinetics parameters within a k-eigenvalue calculation for each benchmark problem. In MCNP6.3.0 the test problems were run with 36 OpenMP threads and in MCNP6.3.1 the test problems are run with 114 MPI ranks. The majority of the calculated Rossi-α mean values and standard deviations changed from version to version even though the calculated k-effective values remained the same. Fortunately, the change in the Rossi-α mean values and standard deviations are all within the statistical uncertainties of each other meaning the results from each calculation appear to be equally reliable. In the future, the robustness of the reactor kinetics parameter calculations will be investigated to make them less sensitive to the parallelization strategy used to run the code.

6 Significant Issues

Beyond the issues discussed in the MCNP6.3.0 version release notes [§5 of 1], no additional significant issues are identified in this release. New known issues can be found in §A.4.

7 Distribution and Installation

The MCNP6.3.1 code distribution package contains the source code, if requested (§7.1), the production and technology preview executables (see §7.2 and §7.3, respectively), various model physics data (§7.4), and several supplemental utilities and scripts (§7.5). Due to the significant similarities when compared to the MCNP6.3.0 code distribution package, each section herein describing the code, data, and miscellaneous utilities provides sufficient details on the updated content. In some instances it may be beneficial to refer to the MCNP6.3.0 release notes [1] for additional details or background information. Likewise, additional details for each individual component of the package are available in the multitude of **README** files included throughout the distribution.

Figure 1 shows the high-level layout of the MCNP6.3.1 code distribution. Many of the shown directories and files are referenced in the sections to follow.

7.1 MCNP6.3.1 Source Code

With the MCNP6.3.1 source code, it is possible to explore many build configurations with or without source code modifications. The most common reason to obtain the source code is to build an MPI-parallel version for use on multi-processor systems. While some production MPI executables are now distributed on each supported operating system, it may be necessary to obtain the source code

Table 2: Production MCNP6.3.1 Executable and Dependency Information

Executable Name	Operating System	Intel oneAPI Version	HDF5 Version	MPI Version
mcnp6	Linux	2023.2.0	1.14.1-2	*
mcnp6.ompi	Linux	2023.2.0	1.14.1-2	OpenMPI 4.1.5
mcnp6.mpi	Linux	2023.2.0	1.14.1-2	MPICH 4.1.1
mcnp6	macOS	2021.5.0	1.10.7	*
mcnp6.ompi	macOS	2021.5.0	1.10.7	OpenMPI 4.1.1
mcnp6.exe	Windows	2023.2.0	1.12.1	*
mcnp6.mpi.exe	Windows	2023.2.0	1.12.1	MS-MPI 10.1

Note: All executables configured with OpenMP enabled
* Default configuration with MPI disabled

to build an MPI-parallel version due to any unforeseen incompatibility issues with the distributed MPI versions. See the discussion on the production MPI executables in §7.2.

All details regarding building the code can be found in the MCNP® Code Version 6.3.1 Build Guide [3]. The compressed source code can be found in the **mcnp-src** directory of the distribution.

7.2 Production MCNP6.3.1

The executables bundled in the code distribution package are built for Linux, macOS, and Windows operating systems. In addition to the MCNP6.3.1 executables, the utilities described in Appendix E of the MCNP® Code Version 6.3.1 Theory & User Manual [2] are also built and packaged alongside the production executables. Table 2 includes all of the production executables by name for each operating system along with the versions of the Intel oneAPI Compilers, HDF5 library, and MPI library used to build each distributed executable. The **install_linux_mac.sh** and **install_windows.bat** scripts in the binaries directory of the distribution shown in Fig. 1 are the installers for Linux/macOS and Windows, respectively.

The production MPI builds are considered "best-effort builds" because ensuring portability of MPI applications is more complex than serial applications. To use the MPI executables, a compatible MPI library must be installed on the system. The binary installers will attempt to detect if a

Figure 1: General Layout of the MCNP6.3.1 Code Distribution

```
MCNP6.3.1
├── README.html ........................................ (file)
├── binaries ........................................... (directory)
│   ├── install_linux_mac.sh ........................... (file)
│   └── install_windows.bat ............................ (file)
├── data ............................................... (directory)
│   └── nd_manager ..................................... (directory)
├── docs ............................................... (directory)
├── licenses ........................................... (directory)
├── mcnp-src ............................... (optional) (directory)
└── utils .............................................. (directory)
    ├── isc ............................................ (directory)
    ├── um_pre_op ...................................... (directory)
    ├── um_pos_op ...................................... (directory)
    ├── vnvstats ....................................... (directory)
    └── whisper ........................................ (directory)
```

compatible MPI library is available on the system before installing the distributed MPI executables. The **README**(s) alongside the binary installers in the binaries directory of the code distribution package includes more detailed system and compatibility information.

Each production executable is packaged with a variety of operating-system-dependent dynamic libraries (**.so** on Linux, **.dylib** on macOS, and **.dll** on Windows) needed by the MCNP6.3.1 code. The libraries are installed when the executables are installed. All of the associated licenses related to the distributed third-party dynamic libraries are included in the top-level licenses directory within the distribution.

7.3 Qt-based Plotter Technology Preview

Alongside the production MCNP6.3.1 executables provided within the code distribution package, a separate collection of technology preview MCNP6.3.1 executables are provided that contain updates to the Qt-based plotter (see §3.1) first released with the MCNP6.3.0. Each of these executables, one each for Linux, macOS, and Windows operating systems, are **runtpe**-compatible with the production MCNP6.3.1 executables. That means that the results from calculations using the production executables can be visualized using either the production legacy plotter or the technology preview Qt plotter.

While the technology preview Qt plotter executables can also be used to do full MCNP simulations, including the Monte Carlo particle transport, these executables have not been rigorously verified and validated for performing full transport simulations. It is highly recommended to use the production MCNP6.3.1 executables for performing full transport simulations, and only using the Qt-based technology preview MCNP6.3.1 executables for geometry, tally, and cross section visualization purposes. Chapter 7 of the MCNP® Code Version 6.3.1 Theory & User Manual contains dedicated information on the new Qt-based plotter technology preview [2].

The Qt-based plotter technology preview executables are distributed with MPI disabled. One difference between the distributed Qt and production packages is the inclusion of a variety of dynamic Qt libraries specific to each operating system. These dynamic Qt libraries are distributed with the code such that Qt is not required to be installed by the user. However, if the user would prefer to install their own or use a system-installed version, the distributed Qt dynamic libraries can be removed in favor of a separately installed Qt library. Qt version 6.7.1 was used to build each of the technology preview executables, named **mcnp6.qt** and **mcnp6.qt.exe** for Linux and Windows, respectively. On macOS, Qt version 5.15.2 was used to build the technology preview executable, named **mcnp6.qt**.

Users are strongly encouraged to provide feedback on their experience with the Qt plotter to mcnp_help@lanl.gov as the MCNP Development Team continues to improve it and move toward it as the single plotting utility provided in a future release.

7.4 Nuclear, Atomic, and Model Physics Data

Much like MCNP6.3.0, the MCNP6.3.1 distribution is not bundled with all of the nuclear and atomic data needed to run the code for many applications. All of the ACE-formatted data is now publicly available on the https://nucleardata.lanl.gov website including all of the same data that was distributed with earlier versions of the MCNP code (see [§5.4.1 of 1] for more information). The non-ACE-formatted model physics data is included on the MCNP6.3.1 distribution (see §7.4.1).

The **nd_manager** utility, discussed further in §7.5.2 and [§5.5.3 of 1], is provided with the distribution to facilitate an easy and reliable installation.

7.4.1 Model Physics and Other Data

The model physics data is included locally within the MCNP6.3.1 distribution. It can be found in the **data/nd_manager/builtin** directory of the distribution. In this release, updates to the **cindergl.dat** file (see §3.3), addition of the ENDF/B-VIII.1-based DBRC data, and addition of the latest ISC 3.0.0 data [9] are all included in the **data/nd_manager/builtin/MCNP_6.3.1_DATA** directory.

7.5 Miscellaneous Utilities and Scripts

In addition to the utilities described in Appendix E of the MCNP® Code Version 6.3.1 Theory & User Manual [2], several separate utilities are distributed as part of the overall MCNP6.3.1 code package. Each separate utility is briefly described herein including some information on the whereabouts of the utilities within the distributed package.

7.5.1 Intrinsic Source Constructor (ISC)

A new version of the Intrinsic Source Constructor, ISC 3.0.0 [9], is included within the MCNP6.3.1 installation. The distribution includes:

- The complete ISC 3.0.0 source code and data packaged into a compressed archive within the **utils/isc** directory of the distribution. Decompress the **isc-3.0.0.zip** archive and get started by reading the **README**.
- A new (α, n) source capability and utility named **mesa** is now available with ISC 3.0.0. It is recommended to read the **mesa** user's guide [10] and verification report [11] to learn more about this new utility.
- New **misc**, **mattool**, and **mesa** binaries for Linux, macOS, and Windows. The MCNP code installation step of the installer handles installing these executables into the same path with the production MCNP6.3.1 executable.
- Updated ISC data files. The data installation step of the installer handles installing the ISC data through the **nd_manager** utility (see §7.5.2).

> **⚠ Caution**
>
> The **nd_manager**, invoked by the installer during the data installation step, does require the use of Python. If Python is not available, the ISC data archive **isc-data-3.0.tar.xz** can be decompressed in any preferred location on the filesystem. To use the data through the binaries or Python wheels, the **ISCDATA** environment variable must be set to the location of the installed data. On the distribution, the ISC data archive is located in the **data/nd_manager/builtin/MCNP_6.3.1_DATA/ISC_DATA-3.0** directory.

See [§5.5.1 of 1] for further information on ISC.

7.5.2 nd_manager Nuclear Data Downloader

To support the nuclear data installation and updating process, the Python-based **nd_manager** utility is provided with the distribution in the **data/nd_manager** directory. The primary functions that the **nd_manager** serves includes:

- Local updates based on remote (or local) database changes.

- Listing available libraries to download and install.
- Downloading all or user-specified libraries.
- Decompress and install downloaded libraries.
- Create and/or update **xsdir** files when libraries are (un)installed.

For more information on the functionality and example usage of the **nd_manager**, see the **README** included with the **nd_manager** utility. As this utility was first released with MCNP6.3.0, see [§5.5.3 of 1] for further information, which details the process the **nd_manager** applies to download and install data from the remote https://nucleardata.lanl.gov database of ACE-formatted libraries and the local **data/nd_manager/builtin** databases of non-ACE-formatted model physics libraries.

> **⚠ Caution**
>
> When the **nd_manager** is used within the MCNP6.3.1 installer, the `-all production` flag is passed to the `download`, `install`, and `create-xsdir` steps. As a result, the data in the **xsdir** files is organized in reverse chronological order of the individual library release date. In comparison to previous releases of the MCNP6 code, this changes the ordering within the **xsdir** file to default to the EPRDATA14 photon data rather than the MCPLIB84 photon data.

7.5.3 Unstructured Mesh (UM) Scripts

The UM-based utilities remain the same as those distributed with MCNP6.3.0. See [§5.5.4 of 1] for further information.

7.5.4 vnvstats Verification and Validation Testing Framework

The Python-based **vnvstats** framework is used to completely setup, execute, post-process, and document several suites of benchmark problems used for verification and validation of the MCNP code.

In this release, a new electron stopping validation test suite is included (see §5.1) in addition to all previously released verification and validation test suites.

The MCNP® Code Version 6.3.1 Verification & Validation Testing [4] document contains all of the comparisons between simulated results and the benchmark values within the **vnvstats** framework. A summary of the results is given in §5. See [§5.5.5 of 1] for further information on **vnvstats**.

7.5.5 Whisper

The Whisper-1.1 [12] package remains the same as was distributed with the MCNP6.3.0. See [§5.5.6 of 1] for further information on Whisper-1.1.

8 Software Quality Assurance

For all MCNP6 development, including source code changes, testing, documentation, and code releases, the MCNP Development Team follows a software quality assurance (SQA) plan defined by LANL [13]. While the SQA plan that the MCNP6 code is developed under is in part derived from other quality control and nuclear safety standards, it cannot be claimed that the MCNP6 code strictly follows any of these other standards.

At LANL, the code is categorized as non-safety commercially controlled software for all general applications. Therefore, the code should not be used for safety significant applications unless qualified to do so by individual users of the code for their specific areas of application. As part of the qualification of the MCNP6 code for specific applications, it is recommended, and may be required, that a suite of qualification tests be developed to cover the application areas of interest beyond those applications presented in the MCNP® Code Version 6.3.1 Verification & Validation Testing report [4].

Acknowledgements

Research and development support for this version of the MCNP code was provided by a number of organizations including the US Department Of Energy's NNSA Advanced Strategic Computing program, the Nuclear Survivability program, the Nuclear Criticality Safety program, and Los Alamos National Laboratory Directed Research and Development (LDRD) program, and the MCNP Site Support program. We thank these sponsors.

Additionally, we would like to thank the many practitioners using the MCNP code who have provided thoughtful and actionable feedback through the mcnp_help@lanl.gov resource, as well as those who have joined in the many community discussions on the new MCNP Forum. The code, data, and documentation for this MCNP6.3.1 release was made better through many of these interactions. We thank the user community for your continued engagement.

References

[Citing pages are listed after each reference.]

1. M. E. Rising, J. C. Armstrong, S. R. Bolding, F. B. Brown, J. S. Bull, T. P. Burke, A. R. Clark, D. A. Dixon, R. A. Forster, III, J. F. Giron, T. S. Grieve, H. G. Hughes, III, C. J. Josey, J. A. Kulesza, R. L. Martz, A. P. McCartney, G. W. McKinney, S. W. Mosher, E. J. Pearson, C. J. Solomon, Jr., S. Swaminarayan, J. E. Sweezy, S. C. Wilson, and A. J. Zukaitis, "MCNP® Code Version 6.3.0 Release Notes," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-22-33103, Rev. 1, Jan. 2023. DOI: 10.2172/1909545 [Pages 3, 8, 10, 11, and 12]

2. J. A. Kulesza, T. R. Adams, J. C. Armstrong, S. R. Bolding, F. B. Brown, J. S. Bull, T. P. Burke, A. R. Clark, R. A. Forster, III, J. F. Giron, A. S. Grieve, C. J. Josey, R. L. Martz, G. W. McKinney, E. J. Pearson, M. E. Rising, C. J. Solomon, Jr., S. Swaminarayan, T. J. Trahan, C. A. Weaver, S. C. Wilson, and A. J. Zukaitis, "MCNP® Code Version 6.3.1 Theory & User Manual," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-24-24602, Rev. 1, May 2024. DOI: 10.2172/2372634 [Pages 3, 9, 10, and 11]

3. J. S. Bull, C. J. Josey, J. A. Kulesza, M. E. Rising, and S. Swaminarayan, "MCNP® Code Version 6.3.1 Build Guide," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-25-22869, Rev. 1, Apr. 2025. [Pages 3 and 9]

4. C. J. Josey, A. R. Clark, J. A. Kulesza, M. A. Lively, E. J. Pearson, and M. E. Rising, "MCNP® Code Version 6.3.1 Verification & Validation Testing," Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-25-22398, Rev. 1, Mar. 2025. [Pages 3, 7, 12, and 13]

5. C. J. Josey, “Reassessing the MCNP Random Number Generator,” Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-23-25111, Rev. 1, Sep. 2023. DOI: 10.2172/1998091 [Page 5]

6. J. E. Sweezy, “Improvements to Contributions from Neutron Inelastic Scattering for Next-Event Estimators in MCNP® Software,” Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-23-31561, Rev. 1, Nov. 2023. DOI: 10.2172/2205018 [Pages 6 and 17]

7. L. L. Carter and E. D. Cashwell, “Particle Transport Simulation with the Monte Carlo Method,” Los Alamos Scientific Laboratory, Los Alamos, NM, USA, Tech. Rep. TID-26607, 1975. [Page 6]

8. M. A. Lively, D. Perez, B. P. Uberuaga, and X. Tang, “Validation of the Single-Event Method for Low-Energy Electron Transport Via Stopping Power Calculations with MCNP,” *Radiation Physics and Chemistry*, vol. 216, p. 111483, Mar. 2024, Los Alamos National Laboratory Tech. Rep. LA-UR-23-29803, Rev. 1. DOI: 10.1016/j.radphyschem.2023.111483 [Page 7]

9. C. J. Solomon, Jr., C. R. Bates, J. A. Kulesza, and M. J. Marcath, “The Intrinsic Source Constructor Package: Installation and Use,” Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-25-23815, Apr. 2025, ISC Version 3.0. [Page 11]

10. C. R. Bates, “MCNPs Easy Sources for (α,n) (MESA) 1.0: A User’s Guide,” Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-24-32000, Nov. 2024. DOI: 10.2172/2476821 [Page 11]

11. C. R. Bates, “MCNPs Easy Sources for (α,n) (MESA): Verification with Sources4c,” Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-24-31721, Oct. 2024. DOI: 10.2172/2475569 [Page 11]

12. F. B. Brown, M. E. Rising, and J. L. Alwin, “User Manual for Whisper-1.1,” Los Alamos National Laboratory, Los Alamos, NM, USA, Tech. Rep. LA-UR-17-20567, Jan. 2017. DOI: 10.2172/1341834 [Page 12]

13. M. E. Rising, F. B. Brown, J. R. Salazar, and J. E. Sweezy, “Overview of the MCNP6® SQA Plan and Requirements,” Los Alamos National Laboratory, Los Alamos, NM, USA, Memorandum XCP-3:20-008/LA-UR-20-26666, Aug. 2020. DOI: 10.2172/1657095 [Page 12]

A MCNP6.3.1 Specific Activities

A.1 New Features

MCNP-53464	A new `dentype` keyword is added to the `EMBED` card and is required to embed LNK3DNT files [§2.1].
MCNP-53671	A new random number generator is added [§2.3].

A.2 Code Enhancements

A.2.1 Data and Physics

MCNP-53601	Update the **xsdir** format and add named identifier support to the code [§2.2].
MCNP-53626	The tabular data **expung.F90** routine had a limit of reading 3000 reactions. This limitation is removed, allowing for arbitrary quantities of reactions.
MCNP-53627	The MARS physics routines had a hardcoded limit of allowed (Z, A) pairs in a simulation of 500. This limitation is removed.
MCNP-53844	The **dbrc_make_lib** utility, which generates data needed for the `DBRC` card, is modified to add in the necessary information to generate the **DBRC_endf81.txt** file.

A.2.2 Unstructured Mesh

MCNP-52874, MCNP-53593	Minor code cleanup and improvements.
MCNP-53604	The `multitrack` keyword from the `EMBED` card is removed [§1.2].
MCNP-53665	Performance of second order tetrahedral elements is improved by reducing the number of temporary arrays.
MCNP-53712	Performance of 1st order hexahedral elements is improved by refactoring the code used to determine if a point is inside a 1st order hexahedral element.

A.2.3 Modernization and Maintenance

MCNP-53118	Unused code is removed from **mcnp_storage.F90**.
MCNP-53270	Improvements to the new Qt-based plotter [§3.1].
MCNP-53508	Modern equivalent C++ versions replace the legacy random number generators.
MCNP-53522	Consistent styling is applied across the source code.
MCNP-53567	Excessive printing of "additional error messages on file" messages to screen is reduced.
MCNP-53594	Implicit type conversion for certain variables is removed.

MCNP-53773 More detailed printing of the specific feature(s) that cause the code to ignore the **tasks** command-line option and run the problem with only one thread is available.

MCNP-53786 Relocate **source** and **srcdx** routines into modules.

MCNP-53846 Laid groundwork for future support of the LLVM Flang compiler. While the code should build with LLVM Flang 20 or newer, it is not yet a supported compiler.

MCNP-53870 Linking against HDF5 dynamically as opposed to statically for improved build resiliency.

A.3 Fixed Bugs

MCNP-53079 In `BURN` calculations, isomeric branching ratios were computed partially using transport-computed 1-group data and CINDER 63-group data. This inconsistent data could result in negative absorption rates. Isomeric branching ratios are now computed solely from the 63-group data. These branching ratios will still multiply transport-computed (n,γ), $(\mathrm{n},2\mathrm{n})$, $(\mathrm{n},3\mathrm{n})$, (n,p), and (n,α) where needed.

MCNP-53188 When the number of entries on the `FMESH` card `xints` keywords, which includes `iints`, `jints`, `kints`, `eints`, and `tints`, is greater than the number of entries on the corresponding `xmesh` keywords, which includes `imesh`, `jmesh`, `kmesh`, `emesh`, and `tmesh`, an array out-of-bounds error would occur. Now, a fatal error is thrown if the number of entries on the `FMESH` `xmesh` and `xints` keywords are not the same.

MCNP-53236 The particle stack size is increased. This fixes a bug that caused the particle stack to overflow. This very rare bug could occur only if a particle was lost on its way to a detector.

MCNP-53408 Build issues with GCC 13.1 are fixed by adding a missing header include and work around a suspected compiler bug in GCC 14.1.

MCNP-53433, MCNP-53577
Cylindrical `FMESH` tally radial bins are now plotted in the correct location if the plot is orientated along the $r\theta$-plane and the plot is not centered on the mesh axis.

MCNP-53387, MCNP-53462, MCNP-53493, MCNP-53815, MCNP-53698
Various typos in comments and error messages are fixed.

MCNP-53548 The dual use of CGM and CGMF makes use of a common configuration control variable that is set every time CGM is used, but was only set the first time CGMF was used. Therefore, this caused a change in behavior in the CGMF simulations during the course of a simulation once CGM was used. This control variable is now set for every call to CGM and CGMF so they now work correctly together. The fix is implemented in CGMF 1.1.2.

MCNP-53571 Parsing the `C` or `T` options on the `FM` card when specified out of order is fixed.

MCNP-53576	When embedding a LNK3DNT structured mesh, a change in the order of the user-specified cell cards could cause incorrect material number selection when tracking within the mesh. This fix corrects this error such that the correct materials are now selected during both CSG and LNK3DNT tracking regardless of the cell ordering.
MCNP-53582	When the `KSEN` `cell` or `mat` keywords were used in a MPI-parallel calculation, the message buffer was undersized leading to incorrect behavior. The MPI buffer size is fixed to contain all of the tally information.
MCNP-53585	For electrons produced from pair production, sampling of the outgoing angular emission was inadvertently using the energy of the incident photon rather than the energy of the electron. Similarly, the angular emission for the positron was being calculated with the electron energy. This is now fixed such that the correct relativistic beta value is used for pair production. This fix is most relevant for electrons created from a low-energy photons ($1.022 - 5.0$ MeV). Though this effect is noticeable at high energies, the changes with respect to the outgoing electron/positron are minor.
MCNP-53609	An issue where the lower quadratic root for the outgoing energy of neutrons from neutron inelastic scattering contributions to next-event estimators (point detectors, ring detectors, image detectors, and DXTRAN) had been previously ignored [6] is fixed. This fix increases the number of low energy contributions to next-event estimators and will be most significant for light nuclei. Models with next-event estimators and containing deuterium are the most affected. This change modifies the random number sequence of problems with next-event estimators and using isotopes with inelastic nuclear data in the center-of-mass frame. This issue also fixes numerical issues when the cosine scattering angle in the center-of-mass frame was near -1.0 and 1.0. Also fixed a bug for Law 44 (Kalbach-87) where center-of-mass angles less than a cosine scattering angle less than -1.0 were incorrectly set to -1.0 [§3.2].
MCNP-53612	Incorrect reporting of number of random numbers used in parallel calculations is fixed.
MCNP-53615	Rare floating-point error when `SDEF` `par` = `sf` and the first history does not produce any particles to track is fixed.
MCNP-53621	Subroutine **`charged_particle_history`** is moved into a module file to fix build issues with Intel Classic 2023.0 and 2023.1.
MCNP-53628	If more than one knock-on electron is created at the end of a substep, only the first electron produced is assigned the knock-on electron tag. The remaining electrons keep the tag of the original electron. This bug is fixed.
MCNP-53641	Missing tally tags for 1) bremsstrahlung photons and 2) knock-on electrons during single electron transport, and for 3) fluorescence photons during the atomic relaxation process are added.
MCNP-53642	Undefined behavior caused by uninitialized variables in several portions of the code is fixed.

MCNP-53662	Missing space in memory allocation error message to prevent the word "array" from running into the array name itself is fixed.
MCNP-53667	An error in the formatting of **cindergl.dat** file was found such that the wrong gamma lines were loaded for some nuclides for the `ACT` card. The file is reprocessed and renamed **cindergl_v3.dat** with revised formatting. The content is otherwise identical to the previous version with the exception that the previously missing ^{252}Cf data was processed [§3.3]. Thanks to Olaf Schumann for reporting this issue.
MCNP-53691	A bug that could cause non-zero relative errors in the plotter and mctal file even though the tally results are zero is fixed.
MCNP-53708	A bug that causes MPI runs to hang if there's an error in the command line is fixed.
MCNP-53720	A bug for `PTRAC` input that incorrectly terminated with a fatal error if a `tally` and `event` filter were specified without a corresponding `value` input is fixed. The code now correctly allows for a default `value` of zero when using the `tally` and `event` filter together.
MCNP-53728	The time of the secondary particles from fission returned by CGMF is now fixed. The secondary particle time is based on the fission collision time in MCNP plus any time delays in CGMF based on late-prompt emission from isomeric states of the fission fragment. Previously, only the time delay from CGMF was returned meaning the secondary particles coming out of the fission event would be reset near or equal to zero time. The fix is implemented in CGMF 1.1.2.
MCNP-53737	MCNP6.3.0 added the ability to shade a cell by its importance in the X11 and Qt plotters. However, it was only implemented for non-void cells. This is fixed. Thanks to Tom Calverley, Rachel Smye, and Iona Webster for identifying this issue.
MCNP-53801	A bug that incorrectly caused surface-source read simulations with a void input to simulate and normalize tallies with 100,000 histories is fixed. The code now correctly uses the desired number of histories for these simulations.
MCNP-53821	The MCNP6.3.0 `FMESH` results when the source type is spontaneous fission (`par` = `sf`) do not scale correctly with the source weight (`wgt`) when specified on the `SDEF` card. MCNP6.2 and earlier versions do not encounter this issue. If using MCNP6.3.0 with the combination of the `FMESH`, spontaneous fission source, and a non-unity source weight, the tally results can be rescaled through the flux multiplier (`FM` card) or manually during post-processing. This issue is now fixed. Thanks to Jerome Verbeke for identifying this issue.
MCNP-53849	The combination of any adjoint-weighted tally (`KSEN`, or `kinetics` = `yes` and/or `precursor` = `yes` on `KOPTS`) with the fission matrix convergence testing on (`fmatconvrg` = `yes` on `KOPTS`), where the code automatically sets the number of inactive cycles (`ikz` on `KCODE`), would result in a stalled

calculation when running with MPI parallelism. This issue did not effect non-MPI parallel calculations. The issue is now fixed such that the combination of options work in all modes of parallelism. Thanks to Aidan Edens for identifying this issue.

MCNP-53869	Previously, the forced collision (`FCL`) and exponential transform (`EXT`) parameters and associated particle identifiers assigned in the cell-card block were unintentionally case sensitive and needed to be lowercase to be active. This bug is fixed and arbitrarily mixed case for these identifiers is permitted. Thanks to David Broughton for identifying this issue.
MCNP-53878	Charged particle ACE files that use ACELAW = 33 as the energy representation in the DLWH block set the LANDH locator to −1, which causes the code to look for a nonexistent angular distribution in the DLWH block. This causes MCNP to interpret memory incorrectly and can lead to an infinite loop as it tries to determine the emission cosine. The code now detects this erroneous information from the ACE file and sets the emission angle to 1.0, which is a good approximation for this highly forward type of scattering.
MCNP-53884	When tabulated (ACE) charged-particle files are in use along with photon transport, the logic in the code to handle photon production from charged-particle interactions was flawed for certain photon yield laws. This flaw results in incorrect memory access, which can present itself as a segmentation fault, a hanging calculation, or silent wrong answers. The impacted charged-particle ACE files include: **`3007.70r`**, **`3007.70s`**, and **`3007.70a`** from CP2011, and **`3007.00r`**, **`3007.00s`**, **`3006.00a`**, and **`3007.00a`** from CP2020. Photon production using these files is now fixed.
MCNP-53889	In MCNP6.3.0 `FMESH` results may not be normalized correctly when performing a `SSR` calculation. The `FMESH` tally results now use the proper weight normalization constant. However, the current results may not match those of previous versions, where this bug was not present in versions prior to MCNP6.3.0, due to a change in the random number sequence. Thanks to Ryan Archibald for identifying this issue.

A.4 Known Issues

MCNP-53230	The on-the-fly (OTF) Doppler broadening capability appears to be incompatible with ENDF/B-VIII.0, incorrectly processing U-238, and incompletely broadening U-235.
MCNP-53658	Tally tagging (`FT` `tag`) fails for cell identification numbers greater than 21,475.
MCNP-53695	Cerenkov photons are banked and given the same reaction number as fluorescence photons.
MCNP-53701	PARTISN cannot read some LNK3DNT files created by the MCNP code.
MCNP-53731	Reaction numbers for banked photons reported in `PTRAC` outputs can be incorrect.

MCNP-53778	Physical and numerical constants are slightly inconsistent with currently-recommended values.
MCNP-53803	Some uses of `ACT` `dg` = `lines` will lead to segmentation faults due to an array that is incorrectly sized.
MCNP-53807	The on-the-fly (OTF) Doppler broadening minimum and maximum temperatures are reported with incorrect units.
MCNP-53824	When performing charged particle transport on an unstructured mesh, a coding error can rarely cause the background cell's density to be used when performing straggling. If the background is a void, this will lead to an infinite loop.
MCNP-53858	Delta ray production (i.e., knock-on electrons from heavy charged particles) has a number of deficiencies related to handling of spin-dependence of incident projectiles.
MCNP-53886	When `swapb` instructions are provided on the `BURN` card for a time step with a power fraction `pfrac` of zero, the instructions are ignored.
MCNP-53887	When using `matmod` on the `BURN` card, material concentration adjustments are only applied to the given time step and then reverted to their original values for subsequent time steps. The adjusted concentrations should persist throughout the calculation.
MCNP-53913	The `WWGT` card operates inconsistent with how it is reported to in the MCNP6.3.1 manual and the `WWGE` card on which it is based. If it is included in the input file with no arguments, a single time group is used by the weight-window generator rather than a collection of default groups. To overcome this behavior, the equivalent input line to use is `wwgt:`$\mathscr{P}$ `1e-7 1e-6 1e-5 1e-4 1e-3 1e-2 1e-1 1e0 1e1 1e2`. Thanks to Tanner Hall for identifying this issue.
MCNP-53927	The CGM option (`ngam` = `2` on `PHYS`:`n`) includes approximations that make this option unsuitable for modeling the production of neutrons and/or gamma rays coming from neutron reactions. That is, unless the Q-value of a specific neutron reaction happens to be ~ 8.5 MeV, the resulting particle emissions from CGM will be inaccurate without the possibility of properly preserving the reaction-production correlations. Until the CGM code and its implementation within MCNP6 is improved, it is not recommended for production use.
MCNP-53932	The X11-based plotter will shade cells by temperature but the Qt-based plotter will not. Thank you to Alex Davis for identifying this issue.
MCNP-53940	The **`dbrc_make_lib`** utility does not create a file with metastable nuclides identified by the `SZZZAAA` format.
MCNP-53943	The `nps` option on the `STOP` card is read as an integer(4) instead of as an integer(8), limiting its maximum value to roughly 2.14 billion. This does not affect the `NPS` card.