

LA-UR-13-22964

Approved for public release;
distribution is unlimited.

Title: MCNPX to MCNP6 Migration Notes

Author(s): Michael R. James

Intended for: MCNP6 Production Release Documentation
May 9, 2013



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 acceptance of this article, the publisher recognizes that the U.S. Government retains a 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.

MCNPX to MCNP6 Migration Notes
Michael James
05/09/2013

This document outlines the issues for users who are accustomed to MCNPX and are switching to MCNP6. MCNP6 operates like previous versions of MCNP5 and MCNPX, but users need to be aware of some minor differences. Input decks may require some changes for them to work as expected. This document is specific to the MCNP6 Version designated MCNP6_Release1 (MCNP6 1.0).

I. Overview.

Documentation. The MCNP distribution contains a large number of reports and papers. Users are encouraged to consult this document archive along with the MCNP6 Manual for a full understanding of the code.

Version numbering. MCNP6 Version identification is reported when the code is run **and** an input deck is processed. The 1st version of the production code is version MCNP6.1.

Speed. MCNP6 is in many cases slower than MCNPX (5-20%).

II. Build/Compiling.

Build system. MCNP6 follows the MCNP5 method for configuring and compiling the code, see the README file in the top-level source directory. You must use Cygwin to build on Windows machines.

LLNL Fission Multiplicity models. The LLNL Fission model consists of C++ routines which are compiled by default in MCNPX. In MCNP6, they are compiled by default on all systems except Windows. To compile this option on Windows machines it is necessary to use the MS Visual C compiler rather than the default Cygwin gcc compiler. (ex. make build CONFIG="intel plot msvc"). Overriding the compilation defaults can be done using the "llnl_fiss" and "no_llnl_fiss" keywords.

Integer4 and Integer8. When building MCNP6 it is not necessary to compile separate versions to gain the ability to run more than 2.1e9 source histories. MCNP6 has 8-byte integers pre-defined where needed.

III. Input Decks and Running.

MPHYS card. The use of physics models can now be controlled with an MPHYS card. MPHYS or MPHYS ON will turn on the use of the physics models, automatically transitioning to physics models above the highest energy data points available (if available) on a per isotope basis (ie. mix and match by isotope is automatic). MPHYS OFF will disable the use of model physics and will result in a fatal error if data libraries are not available for each isotope for each

particle type transported. For particles whose energy exceeds the highest cross section energy point, the highest available energy cross section data point will be used.

All mode n,p,e problems (and subsets) run by default as models off (MPHYS OFF). Any particle on mode card other than n,p, or e will automatically trigger physics models on (MPHYS ON).

Default physics: The CEM physics model is now the default overriding the Bertini model. The FLUKA physics model is not present in MCNP6 and its functions have been primarily taken over by the LAQGS model. Use lca 8j 0 to specify Bertini.

DBCN(29)=1. Beta versions of MCNP6 required the dbcn(29) flag to be set in order to run MCNPX features. This flag is now obsolete and no longer has any effect.

Other Important DBCN entries.

DBCN(18). This entry controls the energy indexing algorithm for electron transport related to bin interpolation. MCNPX used dbcn(18)=0 by default which was the “bin-centered” treatment. MCNP6 has implemented a better Landau treatment, dbcn(18)=2, which is superior. All electron transport problems will show differences due to this change.

DBCN(27). Antiparticle promotion. Use DBCN(27)=1 to lump particle and antiparticle pairs under one particle type (MCNPX behavior).

DBCN(38). Set = 1 to use older barpol.dat. By default, MCNP6 will use the newer barpol2001.dat data file.

DBCN(39). This entry controls the default S(a,b) smoothing behavior which was present in MCNPX but not in MCNP5. In MCNP6, the option is OFF by default. Use dbcn(39)=1 to turn this on.

DBCN(43). Interpolation of Form Factors. Set = 0 to use the linear interpolation from MCNPX. The default treatment is now logarithmic inversion or log-log.

DBCN(52). Set = 1 to use MCNPX functionality in emission of Auger electrons.

In summary, the line
DBCN 17j 0 8j 1 10j 1 1 3j 0 8j 1
will maximize similarity to MCNPX when using MCNP6.

Photon Doppler Broadening is ON by default in MCNP6. This is controlled is with the 5th phys:p entry (0 on, 1 off). The MCNPX default for PDB was OFF.

TOTNU is ON by default in MCNP6. The default in MCNPX was ON and in MCNP5 the default was OFF.

Input changes. Multiple phys:n card entries have been moved to accommodate differences between MCNP5 and MCNPX. Input decks which use PHYS:N 4th, 5th and 6th entries must be changed to run with MCNP6.

PHYS:N 4th entry: The delayed neutron and photon settings have been removed completely from the PHYS:N (4th entry) and PHYS:P (6th entry) cards, respectively. The ACT card controls all delayed physics settings in MCNP6.

PHYS:N 5th entry. The phys:n 5th entry (table-model transition energy) has been moved to phys:n 8th entry.

PHYS:N 6th entry. The fission multiplicity setting on the phys:n card (6th entry) has been moved to the FMULT card. The following table describes how the older settings are mapped to the new input format.

Table 1. FMULT Settings in MCNP6

MCNPX phys:n 6 th entry	MCNP6 FMULT setting
0	method=0, shift=0, data=0 [DEFAULT*]
-1, 1	method=3, shift=1, data=3
2	method=3, shift=2, data=3
3	method=3, shift=3, data=3
4	method=3, shift=4, data=3
5	method=5 [shift and data are unused]
MCNP5 phys:n 5 th entry	
0	method=0, shift=0, data=0 [DEFAULT*]
1	method=0, shift=0, data=1
2	method=0, shift=0, data=2

*Default if no method, data or shift keywords are specified. If any of these keywords appear the code will automatically assign values for the unspecified keywords. The default assignments are method=3, shift=1, data=3

The specific method, shift and data parameter combinations listed here are the only ones assured to work correctly. Other combinations (such as method=1, shift=1 data=2) are possible but have not been tested.

Error Checking for Lattices. In MCNP5 and MCNPX, the following cell card entry was allowed for a cell which was filled with a lattice, but all those lattice locations were filled by the material and density (the 2nd and 3rd entries) of that cell:

```
7 3 -2.7 -11 12 -13 14 -15 16          u=2 lat=1 imp:n=1
```

In effect, the fill card used by the lattice was provided internally to MCNP and given the value of 0. While this is a reasonable intermediate state for constructing geometries, this would occasionally lead to lost particles and the associated geometry error was difficult to debug, in part because the plotter would not indicate this error with red dashed lines. Alternatively, this would also allow a convenient way to get tally results in each of the lattice location, but that function is now much easier (and able to be plotted) with mesh tallies (FMESH). In MCNP6 this lack of the fill card is not allowed, and results in the error message:

```
fatal error. Lattice in cell      7 has no fill.
```

For these cases, the user needs to add an additional cell with the appropriate material and density, and an unused universe number and the corresponding fill. In this example, the above cell should be modified to:

```
7  0      -11 12 -13 14 -15 16 fill=5 u=2 lat=1 imp:n=1
8  3 -2.7 -18                u=5          imp:n=1
```

In the cases where the fill card is used to specify a range and then a finite number of filling locations, the zero fill entry is still valid. For example:

```
5  3 -1.0 -15  2 -18 17          imp:n=1 lat=1 u=2
      fill=0:1 0:3 0:0  4 4 4 4 0 5 4 4  $ 0 here is valid
```

Is a valid cell card description.

Also there are cases where options can be specified in two ways as older interface methods are phased out in favor of new approaches. MCNPX users should take note of RAND card/dbcn. Discussion of these interface differences can be found in the MCNP6 manual.

Additional data files. Some additional data files are necessary to run MCNP6. They are automatically installed when using the RSICC install disk. Also note that there has been significant rearrangement of data files for this production release!

These files are installed in the DATA directory and do not conflict with running MCNPX.

```
bert.dat
pht.dat
barpol2001.dat
delay_library_v2.dat
mollnix.tbl
pelxs.dat
```

Note that the bert.dat and pht.dat are ASCII files and are used in place of the binary bertin and phtlib files in MCNPX. Because they are ASCII, they are not platform and version specific (unlike their MCNPX equivalents).

The MCNP6 distribution has made significant changes to the directory structures which contain the nuclear data. However, these changes are all backwards compatible with MCNPX (and MCNP5). Note that MCNP6 uses a different xsdir file, named xsdir_mcnp6.1 by default. Users are encouraged to install the new nuclear data which includes the newest release ENDF7.1. To continue to use MCNPX, you must reinstall the bertin and phtlib files in your DATAPATH.

Different particle designations. Some particle symbols have changed in MCNP6 and a few particles from MCNPX have been dropped. Particle/antiparticle pairs have been separated into unique particle types. See the table in the Appendix for details.

Multiprocessing. When using multiprocessing with kcode runs, the MCNPX option for load balancing is not available.

Threading. Threading does not work with most MCNPX physics (with the exception of some burnup runs, see the Release notes). Users can request a problem to run threaded but when the use of MCNPX physics and/or features is detected, the code will automatically drop back to a single thread.

MCTAL compatibility. MCTAL files are mostly interoperable between all versions of MCNP5/6/X.

GRIDCONV and HTAPE3X. These programs are not packaged with MCNP6. However, the MCNPX versions should be able to read and process most of the output files from MCNP6.

FLUKA. The FLUKA physics module which was in MCNPX is not present in MCNP6. Its functionality has been replaced by LAQGSM.

IV. Bug/discrepancy list.

List of known issues with the MCNP6.1 version particularly with respect to MCNPX capabilities not otherwise called out in other documentation.

The mcplot arithmetic operations and tally 0 saving ability are not implemented in MCNP6.

In general, mctal files generated by previous versions of MCNP and MCNPX are interoperable with those generated by MCNP6. But because of the rearrangement of the transportable particle lists, some combinations of particle types on the mode card in will produce metal files that are not compatible between MCNP versions.

The HISTP option does not work correctly for MPI runs. The code tries to write (or overwrite) to the same HISTP file multiple times. The code will issue a fatal error when writing HISTP files with MPI.

The geometry plotter is not able to label and shade many of the cell-based properties which could be used in MCNPX (such as den, vol, imp, and mas). This will be fixed in future versions.

Appendix.
Particles in MCNPX and MCNP6

MCNPX				MCNP6			
ipt	char	Name	Comment	ipt	char	Name	Comment
1	n	neutron		1	n	neutron	
2	p	photon		2	p	photon	
3	e	electron		3	e	electron	
4		muon_-		4		mu_minus	
5	*	tau_-	removed	5	q	Aneutron	(anti: 1)
6	u	neutrino_e		6	u	nu_e	
7	v	neutrino_m		7	v	nu_m	
8	w	neutrino_t	removed	8	f	positron	(anti: 3)
9	h	proton		9	h	proton	
10	l	lambda_0		10	l	lambda0	
11	+	sigma_+		11	+	sigma_plus	
12	-	sigma_-		12	-	sigma_minus	
13	x	xi_0		13	x	xi0	
14	y	xi_-		14	y	xi_minus	
15	o	omega_-		15	o	omega_minus	
16	c	lambda_c+	removed	16	!	mu_plus	(anti: 4)
17	!	xi_c_+	removed	17	<	Anu_e	(anti: 6)
18	?	xi_c_0	removed	18	>	Anu_m	(anti: 7)
19	<	lambda_b_0	removed	19	g	Aproton	(anti: 9)
20	/	pi_+		20	/	pi_plus	
21	z	pi_0		21	z	pi_zero	
22	k	k_+		22	k	k_plus	
23	%	k_0_short		23	%	k0_short	
24	A	k_0_long		24	^	k0_long	
25	g	d_+	removed	25	b	Alambda0	(anti: 10)
26	@	d_0	removed	26	_	Asigma_plus	(anti: 11)
27	f	d_s_+	removed	27	~	Asigma_minus	(anti: 12)
28	>	b_+	removed	28	c	Axi0	(anti: 13)
29	b	b_0	removed	29	w	xi_plus	(anti: 14)
30	q	b_s_0	removed	30	@	Aomega_minus	(anti: 15)
31	d	deuteron		31	d	deuteron	
32	t	triton		32	t	triton	
33	s	helium3		33	s	helion	
34	a	alpha		34	a	alpha	
35	#	heavy_ion		35	*	pi_minus	
				36	?	k_minus	
				37	#	heavy_ion	