LA-UR- 11-01832

Approved for public release; distribution is unlimited.

Title: MCNPX to MCNP6 Migration Notes

Author(s): Michael R. James

Intended for: MCNP6 Release Documentation

MCNPX to MCNP6 Migration Notes
Michael James
3/23/2011

This document outlines the issues for users who are accustomed to MCNPX and are interested in using 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 Beta Release Version designated MCNP6_Beta_1.

I. Overview.

Documentation. A complete manual for MCNP6 is still being drafted. Use the MCNP5 and MCNPX manuals in combination with this document and others associated with this release.

Version numbering. MCNP6 Version identification is reported when the code is run **and** an input deck is processed.

Speed. MCNP6 is in most (but not all) cases slightly 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 defaults can be done using the "llnl_fiss" and "no_llnl_fiss" keywords.

Integer4 and Integer8. In 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.

DBCN(29)=1. Most migrating MCNPX users should choose this setting (and must do so explicitly, since dbcn(29)=1 is not the default). This flag is used to invoke MCNPX functionality where there might be conflicting behavior with MCNP5. For example most of the high-energy event generators (LAQGSM, INCL/ABLA, Bertini, etc.) and many features of MCNPX-style mesh tallies are only available when DBCN(29)=1. However, a number of features developed specifically for MCNP do not function unless DBCN(29)=0. For example, the MCNP5 method for variance reduction (including DXTRAN spheres) in the presence of pulse-height tallies, the explicit tracking of charged particles in magnetic fields, and the communication with LNK3DNT files only work when DBCN(29)=0. For full details, see the DBCN document included in this release.

Input changes. Some phys:n card entries have been moved to accommodate differences between MCNP5 and MCNPX. The phys:n 5rd entry (table-model transition energy) has been moved to phys:n 8th entry.
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 FMULT card/phys:n and RAND card/dbcn. Discussion of these interface differences can be found in the MCNP5 manual.

Additional data files. Some additional data files are necessary to run MCNP6. These can be added to your existing data directory and do not conflict with running MCNPX.
bert.dat
pht.dat
laqgsm_atab.dat
barpol2001.dat
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).

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 although when using the dbcn(29)=1 option default antiparticles are lumped together with their counterparts as in MCNPX. See the table in Appendix A and the DBCN document for details.

Multiprocessing. When using multiprocessing with kcode runs, load balancing can be invoked by using the "balance" command on the command line, instead of the "tasks" command with MCNPX. (ex. mpirun –np 12 mcnp6.mpi i=inp balance).

Threading. Threading does not work with most MCNPX physics and is disabled when dbcn(29)=1.

MCTAL compatibility. MCTAL files are mostly interchangeable 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. It expected to be restored in future versions of MCNP6 to provide backwards compatibility.

IV. Bug/discrepancy list.
List of known issues with the MCNP6_Beta_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.

FT8 RES tallies don't work well with other tallies. Problems that mix FT8 RES tallies with other types of tallies may give incorrect results.

The recently implemented M0 card has a limitation that requires library specifications to use the full library id with a leading period, e.g. M0 nlib=.24c.

The mcplot tally plotter appearance lacks some of the updates made in MCNPX. The ordinate axis numbering sometimes overlaps the axis and other small differences.

The LLNL Fission multiplicity option is not interoperable with some other code features. Not all of these combinations have the appropriate warnings and/or fatal errors associated with them.

When the "name" command-line option is used, the mdata files generated by MCNPX mesh tallies are not converted to files with a "named" convention but retain the name "mdata".

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 mctal files that are not compatible between MCNP versions.

The HISTP option does not work correctly for MPI runs on Windows. The code tries to write (or overwrite) to the same HISTP file multiple times.

Sampled lifetimes for muons that capture with a nucleus or decay after coming to rest are computed incorrectly.

DNDG yields for model-generated fission products are too low a significant factor (as much as a factor of two or more).

Results from burnup problems should not be trusted. Some significant bugs are known to be present in the implementation.

Appendix A.
Table 1. 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	I	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	0	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	