NOTE: Use these buttons to print or save the form. DO NOT use the browser tool bar. SAVE PRINT

LA-UR-04-5921

Approved for public release; distribution is unlimited.

Title:	MCNP5_RSICC_1.30
Author(s):	MCNP Monte Carlo Team, X-5 Los Alamos National Laboratory
Submitted to:	

SAVE PRINT CLEAR FORM

Los Alamos NATIONAL LABORATORY

Patch to Update version MCNP5_RSICC_1.20 to version MCNP5_RSICC_1.30

X-5 Monte Carlo Team
Diagnostics Applications Group
Los Alamos National Laboratory

August 2004

This patch (patch-MCNP5_RSICC_1.20_to_1.30) updates the Radiation Safety Information Computational Center (RSICC) release of MCNP5 from version MCNP5_RSICC_1.20 to version MCNP5_1.30. The issues addressed by this patch as well as instructions on applying this patch are given below. This patch can be downloaded from:

www-xdiv.lanl.gov/x5/MCNP/mc/patch-MCNP5_RSICC_1.20_to_1.30

ISSUES

The following items list the new features, bug fixes and minor code improvements implemented in the new release. The T# or R# following the items are the tracking numbers for SQA purposes. Prior to being fixed, four of the following issues (bug fixes # 4, 9, 10, & 13) resulted in incorrect answers.

New Features:

1) Explicit 8 byte integer handling of nps and related variables. These modifications allow MCNP5 to read 8 byte integer values (.ie. from 2.4 billion to ~1E19) for certain input cards and to run these large number of histories. 8 Byte integer input is allowed for both entries on the NPS card, the ndp, ndm and dmmp entries on the PRDMP card, the seed, stride and hist entries on the RAND card, the nps and max keywords on the PTRAC card, the freq entry on the MPLOT card, and the DBUG keyword on the command line. T# 4904
2) Lattice Tally Enhancements. For very specific lattice geometry problems, where many tally features are not used, runtimes can be reduced by more than a factor of 100 by making simple assumptions about the tally routines. This functionality is most commonly used for voxel phantom geometries in problems which use F4 tally cards, DE DF cards and a single multiplier on the FM card to calculate dose or kerma throughout a voxel phantom. This new feature will automatically use the faster coding if the given criteria are met. Tally values will still track the slower version. This capability can be controlled with the SPDTL card. T# 5628
3) Improvements to the fmesh tally. Previously, the user had to specify the material on the FM card whose cross sections are used to calculate the mesh bin values. These cross sections were used for the entire mesh, even if the mesh covered several different materials in the geometry. Now, if you enter 0 as the material number for an FM card associated with the mesh tallies, MCNP will use the cross sections of the material in which the particle is traveling to calculate the bin values. Thus material dependent quantities that are computed with the use of the FM card (ie, neutron heating) can be calculated using mesh tallies that cover more than one material. T# 5588

4) Support for MPI on Mac OS X. The configuration file for Mac OS X systems (MCNP5/Source/config/Darwin.gcf) was enhanced to include options for compiling an MPI version of MCNP5 using LAM-MPI. T# 6015.

Bug Fixes:

1) Fixed bug in code sequence during cross-section interpolation. Does not affect regression test suite. Changes affect acecos.F90 R#
2) Fix mesh tally RUNTPE file read bug. Previous versions crash when using mesh tallies with a dose response function and limiting the number of dumps to 2. Changes affect src/fmesh_mod.F90. T# 5070
3) Decreased lower array index for IBNK to zero. Previous lower array index (1) caused MCNP5 to crash on some Windows PC systems. Reported by Ken Moy, DOE-NV. Changes affect src/imcn.F90, src/msgtsk.F90, and src/tpefil.F90 T# 5069
4) DXTRAN did not work properly if a DXTRAN sphere is located within repeated structure geometry (including lattices). Changes affect src/acegam.F90 src/brems.F90 src/collpn.F90 src/dxdiag.F90 src/dxtran.F90 src/dxtran_mod.F90 src/electr.F90 src/flaug.F90 src/hstory.F90 src/mgcoln.F90 src/startp.F90 src/summary.F90 src/ttbr.F90 src/ypbssp.F90 T# 4760
5) Change error message “CFS fetching no longer possible” to “cannot find xs library file specified in xsdir”. Changes affect ffetch.F90 T# 5141
6) Make changes in dotcomm communications library to allow use with LAMPI. Changes affect dotcomm/include/dotcomm.h dotcomm/src/internals/dotcommi_binary_op.c dotcomm/src/internals/dotcommi_convert.c dotcomm/src/internals/dotcommi_mpi.h dotcomm/src/internals/dotcommi_sizeof.c
7) Change ‘lock_var’ to type i8knd to allow CONFIG=’cheap omp’ on some platforms. Changes affect src/smmp.F90 T# 5160
8) Fix crash when transformations are applied to RPP macrobodies. Change to src/oldcd1.F90 T# 5785
9) Fix mesh tally material number bug. In some cases, the incorrect material was being identified internal to the code when an FM card was used to modify a mesh tally. This would occur if a) the material specified on the FM card was not used in the problem geometry, and b) the material index number (the order number in the INP file) did not match the material number. Changes affect stuff.F90 fmesh_mod.F90 T# 5629
10) Fix bug with lin-lin interpolation with DE DF cards and mesh tallies. Previously, if lin-lin interpolation was used on the DE DF cards which were modifying mesh tallies, the tally results would be all zero. Changes affect src/fmesh_mod.F90 and src/nextit.F90. T# 6002.
11) Fix bug in error message tally number for mesh tallies. Correct tally numbers for mesh tallies are not printed in warning messages. Changes affect fmesh_mod. #T 5999.
12) Fix bug in Doppler broadening routine for sampling of momenta for slightly glancing interactions. T# 4965.
13) Fixed bug that caused incorrect mesh tallies results when used with an FM card that specifies an attenuator or special multiplier set. T#6016.
14) Removed the erroneous warning message stating that a transformation is not used if it is used only for a mesh tally. T# 6087.

15) Fixed a bug in the geometry plotter that caused the theta bin boundaries of the weight-window mesh to be plotted in the incorrect location. T# 6100

Minor Coding Improvements

1) Changed GNU make ‘findstring’ conflict of omp and Compaq. Use GNU make command ‘filer’ instead. Changes affect config/Windows_NT.gcf. T# 4941
2) Removed outdated preprocessor macro GKSSIM. Changes affect all config/*.gcf files and src/ginst.F90 src/gkssim.F90 src/main.F90 and src/pconst.F90. T# 4860
3) Removed outdated preprocessor macro QWIN LAHEY PCDOS. LAHEY Winteracter graphics package is no longer supported, and all MCNP5 plotting executables on any platform must be compiled with an X11 graphics library. Changes affect src/FILE.list src/getexm.F90 src/getxst.F90 src/gxsub.F90 src/isourc.F90 src/ixsdir.F90 src/keypro.F90 src/mc.c src/mcplot_module.F90 src/newcd1.F90 src/newcrd.F90 src/plotg.F90 src/sread.F90 src/stuff.F90 src/tallyh.F90 src/tekdvr.F90 src/track.F90 src/ttyint.F90 src/volume.F90 src/xsgen.F90. T# 4850
4) Changed !$OMP ATOMIC directive to !$OMP CRITICAL. Changes affect src/bankit.F90 src/errprn.F90 src/prlost.F90. T# 4921
5) Added several named constants to mcnp_params.F90. T# 4900
6) Changed some format statements from 1peX.Y to esX.Y
7) Added _dknd to some numeric constants.
8) Changed the parameter huge to huge_real to allow the use of Fortran 90 intrinsic function huge(). T# 4900
9) Removed pack and unpack string routines in src/internals/mpi.
10) Added nullify statements for allocated arrays. T# 5020
11) Explicitly calculate maximum record length for the MESHTAL file. T# 5304
12) Change non-standard control edit descriptor used in format statement in fmesh_mod. T# 5942.
13) Expanded Table 32 to provide additional information regarding FM cards used with mesh tallies. T# 6024.
14) Support for Intel Fortran 95, version 7.1 for Windows PCs. T# 6026

INSTALLATION INSTRUCTIONS

For best results apply this patch to an unmodified copy for the November, 2003 RSICC release of MCNP5 (MCNP5_RSICC_1.20). This patch may not work with a modified version of MCNP5_RSICC_1.20. To verify the version number of your copy of the MCNP5 source see the file MCNP5/Source/config/VC_info.gcf”. This file should contain the following:

```
# --- Automatically Generated Version Control Information file
# --- Thread Name
THREAD = MCNP5_RSICC
# --- Thread Version Number
THD_VERS = 1.20
```

After verifying that you have an unmodified copy of MCNP5_RSICC_1.20 perform the following steps to install the patch.

1) Verify that you have the GNU patch utility installed by issuing the command “patch –v”. You should see output that looks similar to the output below. Note the version may be different.

```
$ patch -v
patch 2.5.8
Copyright (C) 1988 Larry Wall
Copyright (C) 2002 Free Software Foundation, Inc.

This program comes with NO WARRANTY, to the extent permitted by law.
You may redistribute copies of this program
under the terms of the GNU General Public License.
For more information about these matters, see the file named COPYING.

written by Larry Wall and Paul Eggert
```

2) Save the patch file “patch-MCNP5_RSICC_1.20_to _1.30” to the MCNP5 directory.

3) Change your working directory to the MCNP5 directory.

4) Apply the patch with the following command

```
$ patch –p1 < patch-MCNP5_RSICC_1.20_to_1.30
```

5) Rebuild the MCNP5 executable.

6) Run the Regression Test suite.

On Unix/Linux platforms, cd to the MCNP5/Testing/Regression directory, then

```
$ make test1
```

On Windows PCs, cd to the MCNP5/Installation directory, then

```
> cleanup
> runprob
```

7) Compare to expected answers. Since the expected answer file templates are stored in a .tar file, they cannot be updated with the patch utility. The following differences in the output files are expected. No differences in the mctal files should be present.

```
diff inp01o outp01
1987,1988c1987
<  binary file inp01p   written with          694 events
<                               from          100 histories.
---
>  binary file inp01p   written with     694 events from     100 histories.
```

```
diff inp02o outp02
2038,2039c2038
<  ascii  file inp02p   written with          194 events
<                                from          194 histories.
---
>  ascii  file inp02p   written with     194 events from     194 histories.

diff inp08o outp08
1258,1259c1258
<  ascii  file inp08p   written with            0 events
<                                from            0 histories.
---
>  ascii  file inp08p   written with       0 events from       0 histories.

Diff inp15o outp15
43,44d42
<  comment. lattice speed tally modifications will not be used.
<

diff inp16o outp16
109,110d108
<  comment. lattice speed tally modifications will not be used.
<

diff inp17o outp17
51,52d50
<  comment. lattice speed tally modifications will not be used.
<

diff inp18o outp18
1504,1505c1504
<  ascii  file inp18p   written with            3 events
<                                from            3 histories.
---
>  ascii  file inp18p   written with       3 events from       3 histories.

diff inp23o outp23
1261,1262c1261
<  ascii  file inp23p   written with          253 events
<                                from            6 histories.
---
>  ascii  file inp23p   written with     253 events from       6 histories.

diff inp24o outp24
81,82d80
<  comment. lattice speed tally modifications will not be used.
<

diff inp36o outp36
126c126
<      3        3  0.00000E+00   3.14159E+00
---
>      3        3  0.00000E+00   0.00000E+00   infinite
128c128
<      5        5  0.00000E+00   8.04248E+02
---
>      5        5  0.00000E+00   0.00000E+00   infinite

diff inp38o outp38
137,138d136
<  comment. lattice speed tally modifications will not be used.
<
573,575c571,573
<      1        1  0.00000E+00   1.25664E+05
<      2        2  0.00000E+00   0.00000E+00   asymmetric
<      3        3  0.00000E+00   0.00000E+00   asymmetric
---
>      1        1  0.00000E+00   1.88496E+05
>      2        2  0.00000E+00   4.80664E+04
>      3        3  0.00000E+00   4.80664E+04
585c583
```

```
<     13       13  0.00000E+00   0.00000E+00   asymmetric
---
>     13       13  0.00000E+00   0.00000E+00   infinite
```

```
# patch-MCNP5_RSICC_1.20_to_1.30 --- LA-UR-04-5921
#
# Patch to create MCNP5_RSICC_1.30 from MCNP5_RSICC_1.20
#
# USAGE
# ------------------------------------------------------------
# To apply this patch to an unmodified copy of the November, 2003 RSICC
# release of MCNP5 (CCC-710), MCNP5_RSICC_1.20, follow the directions
# below.
#
# 1) Verify that you have the GNU patch utility installed by issuing
#    the command "patch -v".  You should see output that looks similar
#    to the output below.  Note that the version may be different.
#
#    $ patch -v
#    patch 2.5.8
#    Copyright (C) 1988 Larry Wall
#    Copyright (C) 2002 Free Software Foundation, Inc.
#
#    This program comes with NO WARRANTY, to the extent permitted by law.
#    You may redistribute copies of this program
#    under the terms of the GNU General Public License.
#    For more information about these matters, see the file named COPYING.
#
#    written by Larry Wall and Paul Eggert
#
# 2) Make sure there are not carrage return (^M) characters in
#    Source/dotcomm/src/dotcomm_pack.F90 and
#    Source/dotcomm/src/internals/dotcommi_unpackstring.c
#    The ^M characters may make the patch fail on some platforms.
#    Simple editors such as vi or xemacs can be used to identify ^Ms and remove them.
#    Some simple programs, such as to_unix or dos2unix will also work.
#
# 3) Save the patch file "patch-MCNP5_RSICC_1.20_to_1.30" to the MCNP5
#    directory.
#
# 4) Change your working directory to the MCNP5 directory.
#
he# 5) Apply the patch with the following command:
#
#    $ patch -p1 < patch-MCNP5_RSICC_1.20_to_1.30
#
# 6) Recompile MCNP5.
#
# Note: This patch may fail if you have modified MCNP5.
# -------------------------------------------------------------
#
Prereq: 1.20
diff -Naurd MCNP5/Source/config/VC_info.gcf MCNP5_new/Source/config/VC_info.gcf
--- MCNP5/Source/config/VC_info.gcf  Wed Nov  5 17:23:12 2003
+++ MCNP5_new/Source/config/VC_info.gcf      Thu Jul 22 15:06:14 2004
@@ -2,4 +2,4 @@
 # --- Thread Name
 THREAD = MCNP5_RSICC
 # --- Thread Version Number
-THD_VERS = 1.20
+THD_VERS = 1.30
diff  -Naurd MCNP5/Source/config/AIX.gcf MCNP5_new/Source/config/AIX.gcf
--- MCNP5/Source/config/AIX.gcf      2003-04-30 20:08:44.000000000 -0600
+++ MCNP5_new/Source/config/AIX.gcf   2004-07-22 15:06:14.000000000 -0600
@@ -57,7 +57,7 @@

 # --- Plot option.
 ifeq (plot,$(findstring plot,$(CONFIG)))
-  DEF_PLOT =-DPLOT -DMCPLOT -DGKSSIM -DXLIB
+  DEF_PLOT =-DPLOT -DMCPLOT -DXLIB

   ifeq (,$(premake))
     PLOTLIBS = -lX11
diff  -Naurd MCNP5/Source/config/Darwin.gcf MCNP5_new/Source/config/Darwin.gcf
--- MCNP5/Source/config/Darwin.gcf   2003-11-05 17:23:12.000000000 -0700
```

```
+++ MCNP5_new/Source/config/Darwin.gcf       2004-07-22 15:06:14.000000000 -0600
@@ -11,26 +11,30 @@
 # Must have the Developers Toolkit + X11
 #   installed.
 #
-# As of 4/28/2003, this was only tested on a
-# PowerBook G4 running Mac OS X (10.2.4)
+#This was tested on a PowerBook G4 running
+#  * Mac OS X 10.2.4
+#  * Mac OS X 10.3
+#
+# For MPI version, use LAM-MPI  (not MPICH)
+# See instructions at the end of this file.
 #
 ##########################################

 # -------COMPILER OPTIONS------
-FPPmcnp  := $(shell pwd)/../config/fpp
+FPPmcnp  = $(CONFIG_DIR)/fpp

 # --- Configuration
 export CONFIG
 ifeq (,$(CONFIG))
-  CONFIG = cheap seq absoft gcc
+  CONFIG = cheap seq absoft gcc plot
 endif
 ifeq (default,$(CONFIG))
-  CONFIG = cheap seq absoft gcc
+  CONFIG = cheap seq absoft gcc plot
 endif

 # Compilers
-FCOMPILER = ABSOFT
-CCOMPILER = gcc
+  FCOMPILER = ABSOFT
+  CCOMPILER = gcc

 # --- path to cross-sections
 ifeq (,$(DATAPATH))
@@ -39,31 +43,62 @@

 # --- Plot option.
 ifeq (plot,$(findstring plot,$(CONFIG)))
-  DEF_PLOT = -DPLOT -DMCPLOT -DGKSSIM -DXLIB
+  DEF_PLOT = -DPLOT -DMCPLOT -DXLIB
+
+  ifeq (,$(premake))
+    PLOTLIBS = -L/usr/X11R6/lib -lX11
+  else
+    menuglib := $(subst lib,,$(menuglib))
+    menuglib := $(subst .so,,$(menuglib))
+    menuglib := $(subst .a,,$(menuglib))
+    PLOTLIBS = -L$(menugpath) -l$(menuglib)
+  endif
+
 endif

 #################### ABSOFT with gcc ####################

 ifeq (ABSOFT,$(findstring ABSOFT,$(FCOMPILER)))

-  ABSOFT    ?= '/Applications/Absoft'
-  CCOMPILER ?= gcc
+  ABSOFT   = /Applications/Absoft

   # --- Optimize/Debug options.
   ifeq (debug,$(findstring debug,$(CONFIG)))
-    FDEBUG = -g
-    CDEBUG = -g
-    FOPT   =
-    COPT   =
+    FDEBUG =
+    CDEBUG =
+    FOPT   = -g   -Z1643 -Z1644  -B18 +B51 -et
+    COPT   = -g   -fno-common -malign-natural
   else
     FDEBUG =
     CDEBUG =
-    FOPT   = -O1  -Z1643 -Z1644
-    COPT   =
+    FOPT   = -O1  -Z1643 -Z1644  -B18 +B51 -et
+    COPT   = -O1  -fno-common -malign-natural
   endif

-  INCLUDE_DIRS = -I'/usr/X11R6/include'
-  PLOTLIBS     = -L/usr/X11R6/lib  -lX11
+  # --- Distributed Multiprocessing options
+  # --- MPI option.
+  ifeq (mpi,$(findstring mpi,$(CONFIG)))
+    MPICH_ROOT       = /usr/local/lam
+    DOTCOMMROOT      = ../dotcomm
+    DOTCOMM_INTERNAL = mpi
+    DMMP_NAME        = -DDMMP_NAME=$Q$(EXEC)$Q
+    DEF_DMMP         = -DMULTP -DMPI -DMPICH
+    INC_DMMP_INTERNAL=
+    INC_DMMP         =
+    LIB_DMMP         = -L$(DOTCOMMROOT)/src -ldotcomm
+    LIBDOTCOMM       = $(DOTCOMMROOT)/src/libdotcomm.a
+    MOD_DMMP         = -p ../../src
+    MPIO             =
+    MPI              = mpich
+    MPICH            = -DMPICH
+    MPIFC=$(MPICH_ROOT)/bin/mpif77
+    MPICC=$(MPICH_ROOT)/bin/mpicc
+  else
+    MPIFC=$(FC)
+    MPICC=$(CC)
+  endif
+
+  INCLUDE_DIRS = -I/usr/X11R6/include

   DEF_MACH = -DMACOSX  -DABSOFT -DDEC -DCHEAP $(ROSSI)

@@ -71,9 +106,10 @@
   PREPROCESS := yes
   MISC= export ABSOFT; ABSOFT=$(ABSOFT);

-  FC       = $(MISC)  $(ABSOFT)/BIN/f90
+  FC       = $(MISC)  $(ABSOFT)/bin/f90
   FFLAGS   = $(FDEBUG) $(FOPT) $(SMMP) $(MPIO) \
-              -N11  -YEXT_NAMES=LCS  -YEXT_SFX=_ -YCFRL=1
+              -YEXT_NAMES=LCS  -YEXT_SFX=_ -YCFRL=1 \
+              -I../../src -I. -p .  -p ../dotcomm/src
   OBJF   = .o
   MOD_INC  = -I

@@ -82,12 +118,11 @@
   I      = .f95

   CPPFLAGS = $(DEFS)
-  CFLAGS   = -o $(<:.c=$(OBJC)) $(CDEBUG) $(COPT) $(WIN32)
+  CFLAGS   = $(CDEBUG) $(COPT)
   OBJC   = .o
   CC       = /usr/bin/gcc
-  WIN32    =

-  LD       = $(FC)
+  LD       = $(MPIFC)
   LDFLAGS  = $(FFLAGS) -lU77
   OUT      = -o  #

@@ -95,6 +130,8 @@
```

```
 endif

 # --- some other setup stuff
+AR      = /usr/bin/ar
+ARFLAGS = -rvus
 ECHO    = echo
 RM      = /bin/rm
 MV      = /bin/mv
@@ -104,3 +141,140 @@

 export
 # --- End of Make Include for Darwin
+
+#############################################################
+#
+# Instructions for installing LAM-MPI & MCNP5 for
+# parallel processing on a Mac OS X cluster
+#
+#      F Brown, 5/26/2004
+#
+# Notes:
+#
+# 1) You must make sure that the options used for compiling
+#    & installing LAM-MPI and MCNP5 are consistent. The
+#    instructions below are for installing the LAM-MPI package
+#    (NOT MPICH) using the Absoft Fortran-90 compiler & gcc.
+#
+# 2) I tried using MPICH from the ANL website. While it
+#    installed correctly & passed tests, and linked OK with
+#    MCNP5, there were SIGBUS error interrupts when trying
+#    to execute. This problem has not yet been resolved.
+#    For now, it is necessary to use LAM-MPI rather than
+#    MPICH for parallel MCNP5 on Mac OS X clusters.
+#
+# 3) Our convention for compiling mixed Fortran & C is to
+#    set options so that external names for the linker are
+#    lower case with an appended underscore. The Fortran
+#    options for doing this with Absoft F90 must be included
+#    with the FFLAGS environment variable used in compiling
+#    LAM-MPI.
+#
+# 4) LAM-MPI does not support F90 directly, just F77 & C.
+#    There are utilities called mpif77 and mpicc, but not
+#    mpif90. This is OK for use with MCNP5, however, since
+#    all MPI calls are made through the "dotcomm" routines,
+#    which use C at the lowest level.
+#
+# 5) We use the Absfoft F90 compiler, not the F77 compiler,
+#    for both LAM-MPI and MCNP5. For LAM-MPI, the FC environment
+#    variable is set to point to the Absoft Fortran-90 compiler.
+#    Then, invoking mpif77 really uses Fortran-90, not Fortran-77.
+#
+# 6) I installed LAM-MPI using the script shown below into
+#    the directory  /usr/local/lam.  Note that this directory
+#    is hardwired into the MCNP5 config information above in
+#    this file. If you install LAM-MPI in a different location,
+#    you must change the variable MPICH_ROOT above in this file.
+#
+# 7) After installing LAM-MPI, I "booted" LAM-MPI
+#    with the command "lamboot -v".  See the LAM-MPI documentation.
+#    Also, make sure that /usr/local/lam/bin is in your PATH.
+#
+# 8) These lines must be inserted into
+#    MCNP5/Source/dotcomm/include/dotcomm.h
+#    after line #162 (#include "mpi.h"):
+#
+#    #ifdef LAM_MPI
+#    /* -------------------------------------------------------- */
+#    /* if using LAM-MPI, must add defines for Fortran datatypes */
+#    /* -------------------------------------------------------- */
+#    #define MPI_2REAL            ((MPI_Datatype) &lam_mpi_double)
+#    #define MPI_2INTEGER    ((MPI_Datatype) &lam_mpi_long_long_int)
+#    #define MPI_REAL        ((MPI_Datatype) &lam_mpi_float)
+#    #define MPI_INTEGER     ((MPI_Datatype) &lam_mpi_long_int)
+#    #endif
+#
+# 9) Then, MCNP5 was compiled this way:
+#      cd MCNP5/Source
+#      make CONFIG='absoft gcc mpi plot cheap' NMPI=3 install
+#
+#    This compiles the code & runs the tests with 3 MPI processes
+#    (master + 2 slaves).  There will be lots of minor diffs in
+#    the output files, but mctal files will match exactly,
+#    except for problems 21, 28, 33, 34, 41. See the notes
+#    printed at the end.
+#
+#
+# ------------------------------------------------------------
+#   #!/bin/bash
+#   ######################################
+#   # Install LAM-MPI for use with MCNP5 #
+#   ######################################
+#
+#   #
+#   # location of tarfile for LAM-MPI,
+#   # downloaded from http://www.lam-mpi.org/7.0/download.php
+#   #
+#   LAMMPI_TGZ="/Users/fbrown/mcnp/LAM-MPI/lam-7.0.6.tar.gz"
+#
+#   #
+#   # directory for creating lam-7.06 dir & building LAM-MPI
+#   #
+#   LAMMPI_WORK="/Users/fbrown/mcnp/LAM-MPI"
+#
+#   #
+#   # target directory for installing LAM-MPI
+#   #
+#   #    Note: do the following BEFORE running this script:
+#   #       sudo mkdir /usr/local/lam
+#   #       sudo chown fbrown  /usr/local/lam
+#   #
+#   LAMMPI_TARGET="/usr/local/lam"
+#
+#   echo ".....un-tar lam-mpi....."
+#   cd $LAMMPI_WORK
+#   rm -rf lam-7.0.6
+#   tar xfz $LAMMPI_TGZ
+#
+#   cd lam-7.0.6
+#
+#   echo ".....set environment variables for lam-mpi....."
+#   export CC
+#   CC=gcc
+#
+#   export CFLAGS
+#   CFLAGS="-fno-common -malign-natural"
+#
+#   export CXX
+#   CXX=g++
+#
+#   export FC
+#   FC=/Applications/Absoft/bin/f90
+#
+#   export FFLAGS
+#   FFLAGS="-O1 -YEXT_NAMES=LCS -YEXT_SFX=_ -YCFRL=1 -B18 +B51 -et"
+#
+#   export PATH
+#   PATH="$LAMMPI_TARGET/bin:$PATH"
+#
+#   rm -rf $LAMMPI_TARGET/*
+#
+#   echo ".....configure, make, & install lam-mpi....."
```

```
+#
+#   ./configure --prefix=$LAMMPI_TARGET   --with-rsh="ssh -x"
+#
+#   make
+#
+#   make install
+#
diff  -Naurd MCNP5/Source/config/IRIX64.gcf MCNP5_new/Source/config/IRIX64.gcf
--- MCNP5/Source/config/IRIX64.gcf   2003-04-30 20:08:46.000000000 -0600
+++ MCNP5_new/Source/config/IRIX64.gcf       2004-07-22 15:06:14.000000000 -0600
@@ -55,7 +55,7 @@

 # --- Plot option.
 ifeq (plot,$(findstring plot,$(CONFIG)))
-  DEF_PLOT = -DPLOT -DMCPLOT -DGKSSIM -DXLIB
+  DEF_PLOT = -DPLOT -DMCPLOT -DXLIB

   ifeq (,$(premake))
     PLOTLIBS = -lX11
diff  -Naurd MCNP5/Source/config/Linux.gcf MCNP5_new/Source/config/Linux.gcf
--- MCNP5/Source/config/Linux.gcf    2003-11-05 17:23:12.000000000 -0700
+++ MCNP5_new/Source/config/Linux.gcf 2004-07-23 15:49:18.000000000 -0600
@@ -53,7 +53,7 @@

 # --- Plot option.
 ifeq (plot,$(findstring plot,$(CONFIG)))
-  DEF_PLOT = -DPLOT -DMCPLOT -DGKSSIM -DXLIB
+  DEF_PLOT = -DPLOT -DMCPLOT -DXLIB

   ifeq (,$(premake))
     PLOTLIBS = -L/usr/X11R6/lib -lX11
@@ -176,7 +176,7 @@

   ifeq (debug,$(findstring debug,$(CONFIG)))
     CDEBUG = -g
-    FDEBUG = -g
+    FDEBUG = -g
     FOPT = -O0
   else
     FOPT = -O1 -tp px
@@ -288,10 +288,14 @@

   ARCH:=$(shell uname -m)
   ifeq (,$(premake))
-    ifeq (ia64,$(findstring ia64,$(ARCH)))
-      FC=efc
+    ifeq (ifort,$(findstring ifort,$(shell which ifort)))
+      FC=ifort
     else
-      FC=ifc
+      ifeq (ia64,$(findstring ia64,$(ARCH)))
+        FC=efc
+      else
+        FC=ifc
+      endif
     endif
     CC=gcc
     ifeq (mpi,$(findstring mpi,$(CONFIG)))
@@ -303,11 +307,11 @@
   ifeq (ia64,$(findstring ia64,$(ARCH)))
     # Itanium
     FOPT = -O0
-    FFLAGS   = $(FDEBUG) $(FOPT) $(SMMP) -fpp2 -W0 -C90 -Vaxlib -static-libcxa $(I8R8)
$(MPIO)
+    FFLAGS   = $(FDEBUG) $(FOPT) $(SMMP) -fpp2 -W0 -Vaxlib -static-libcxa $(I8R8)
$(MPIO)
   else
     # i386 and up
     FOPT = -O0
-    FFLAGS   = $(FDEBUG) $(FOPT) $(SMMP) -fpp2 -pc64 -W0 -C90 -Vaxlib -static-libcxa
$(I8R8) $(MPIO)
```

```
+    FFLAGS   = $(FDEBUG) $(FOPT) $(SMMP) -fpp2 -pc64 -W0 -Vaxlib -static-libcxa $(I8R8)
$(MPIO)
   endif
   OBJF     = .o
   DEF_FCOMPILER= -DINTEL
diff  -Naurd MCNP5/Source/config/OSF1.gcf MCNP5_new/Source/config/OSF1.gcf
--- MCNP5/Source/config/OSF1.gcf      2003-11-05 17:23:12.000000000 -0700
+++ MCNP5_new/Source/config/OSF1.gcf  2004-07-22 15:06:14.000000000 -0600
@@ -52,7 +52,7 @@

 # --- Plot option.
 ifeq (plot,$(findstring plot,$(CONFIG)))
-  DEF_PLOT = -DPLOT -DMCPLOT -DGKSSIM -DXLIB
+  DEF_PLOT = -DPLOT -DMCPLOT -DXLIB
   PLOTLIBS = -lX11
 endif

diff  -Naurd MCNP5/Source/config/SunOS.gcf MCNP5_new/Source/config/SunOS.gcf
--- MCNP5/Source/config/SunOS.gcf     2003-04-30 20:08:50.000000000 -0600
+++ MCNP5_new/Source/config/SunOS.gcf 2004-07-22 15:06:14.000000000 -0600
@@ -56,7 +56,7 @@

 # --- Plot option.
 ifeq (plot,$(findstring plot,$(CONFIG)))
-  DEF_PLOT = -DPLOT -DMCPLOT -DGKSSIM -DXLIB
+  DEF_PLOT = -DPLOT -DMCPLOT -DXLIB
   PLOTLIBS = -lX11
 endif

diff  -Naurd MCNP5/Source/config/Windows_NT.gcf MCNP5_new/Source/config/Windows_NT.gcf
--- MCNP5/Source/config/Windows_NT.gcf       2003-04-30 20:08:52.000000000 -0600
+++ MCNP5_new/Source/config/Windows_NT.gcf   2004-07-22 15:06:14.000000000 -0600
@@ -3,14 +3,15 @@
 ##########################################
 #
 # Fortran 90 compilers supported:
-#  Compaq Fortran    (6.6b or higher)
-#  Lahey Fortran 95  (5.7b or higher)
-#  Absoft Fortran    (7.5  or higher)
+#  Compaq Fortran    (6.6b only) [sequential, plotting, mpi, pvm]
+#  Lahey Fortran 95  (5.7b or higher) [sequential, plotting]
+#  Absoft Fortran    (7.5  or higher) [sequential, plotting]
+#  Intel Fortran 95  (7.1 only)       [sequential, plotting]
 #
 # C Compilers supported
-#  Microsoft C++  (cl)
-#  GNU gcc
-#  Fujitsu C/C++ (with Lahey Fortran90)
+#  Microsoft C++  (cl)                  [sequential, plotting, mpi, pvm]
+#  GNU gcc                              [sequential, plotting]
+#  Fujitsu C/C++ (with Lahey Fortran90) [sequential, plotting]
 #
 ##########################################

@@ -36,7 +37,11 @@
     ifeq (absoft,$(findstring absoft,$(CONFIG)))
       FCOMPILER = ABSOFT
     else
-      FCOMPILER = CVF
+      ifeq (intel,$(findstring intel,$(CONFIG)))
+        FCOMPILER = INTEL
+      else
+        FCOMPILER = CVF
+      endif
     endif
   endif
 endif
@@ -96,7 +101,7 @@

 # --- Plot option.
 ifeq (plot,$(findstring plot,$(CONFIG)))
```

```
-  DEF_PLOT = -DPLOT -DMCPLOT -DGKSSIM -DXLIB
+  DEF_PLOT = -DPLOT -DMCPLOT -DXLIB
 endif

 #################### CVF with either gcc or cl ####################
@@ -162,7 +167,7 @@
   Q     = '

   # --- OpenMP threads option.
-  ifeq (omp,$(findstring omp,$(CONFIG)))
+  ifeq (omp,$(filter omp,$(CONFIG)))
     SMMP     =
     DEF_SMMP = -DMULTT
   endif
@@ -239,7 +244,7 @@

 ifeq (ABSOFT,$(findstring ABSOFT,$(FCOMPILER)))

-  ABSOFT    ?= 'c:\Absoft80'
+  ABSOFT    ?= '/cygdrive/c/Absoft80'
   CCOMPILER ?= gcc

   # --- Optimize/Debug options.
@@ -264,7 +269,8 @@
   PREPROCESS := yes
   MISC = export ABSOFT; ABSOFT=$(ABSOFT);

-  FC       = $(MISC)  $(ABSOFT)/BIN/f90
+  #FC       = $(MISC)  $(ABSOFT)/BIN/f90
+  FC       =  /cygdrive/c/Absoft80/bin/f90
   FFLAGS   = $(FDEBUG) $(FOPT) $(SMMP) $(MPIO)
   OBJF   = .obj
   MOD_INC  = -I
@@ -303,6 +309,94 @@

 endif

+#################### INTEL with either gcc or cl ####################
+
+ifeq (INTEL,$(findstring INTEL,$(FCOMPILER)))
+
+  # --- Optimize/Debug options.
+  ifeq (debug,$(findstring debug,$(CONFIG)))
+    FDEBUG = /Zi
+    CDEBUG = -g
+    FOPT  = /Od /Ob0 /Qfp_port /Qpc64 /0p
+    COPT  =
+  else
+    FDEBUG =
+    CDEBUG =
+    FOPT  = /Od /Ob0 /nologo   #/Ox will cause the regression test suite to fail!
+    COPT  =
+  endif
+
+  # --- Include and Module search directories.
+  INCLUDE_DIRS =  -I'..\X11R6\include'
+
+  # --- Location(s) for parts of the code in separate directories.
+  DEDXROOT ='..\dedx'
+  ifdef TESTING
+    DEDX_UNIT_TEST = -DDEDX_UNIT_TEST
+  endif
+  DEDX_UNIT_TEST_FLAG = $(DEDX_UNIT_TEST)
+  DEDX_LD_LIB_PATH = /LINK /LIBPATH:..\
+
+  # --- Plotting
+  PLOTLIBS = /LIBPATH:'..\X11R6\lib'   X11.lib
+
+  # --- Machine dependent CPP flags.
+  #DEF_MACH = -DCVF -DDEC -DCHEAP $(ROSSI) -DDIRACCESS_RECL_WORDS $(WIN32)
+
+  PREPROCESS := yes
+  PREPROCESS :=
+
+  ifeq (,$(PREPROCESS))
+    FPP       =
+    DEF_MACH = -DINTEL -DCHEAP $(ROSSI) -DDIRACCESS_RECL_WORDS -DWIN32 $(WIN32)
+    FPPFLAGS =  /Qfpp2 $(DEFS)
+    ALL_FPPFLAGS = $(FPPFLAGS)
+    Q    = \'#
+  else
+    FPP       = $(FPPmcnp)
+    DEF_MACH = -DINTEL -DCHEAP $(ROSSI) -DDIRACCESS_RECL_WORDS -DWIN32 $(WIN32)
+    FPPFLAGS =  $(DEFS)  -s $(I)
+    ALL_FPPFLAGS = $(FPPFLAGS)
+    I  = .for
+    DATAPATH := "$(shell echo "$(DATAPATH)" | sed -e 's/\\/\\\\/g')"
+    Q    = \'#
+  endif
+
+
+  CFLAGS  = $(CDEBUG) $(COPT)
+  CPPFLAGS= $(DEFS)
+  ifeq (gcc,$(findstring gcc,$(CCOMPILER)))
+    CC    = /usr/bin/gcc
+    OBJC  = .o
+    WIN32 =
+  endif
+  ifeq (cl,$(findstring cl,$(CCOMPILER)))
+    CC    = cl
+    OBJC  = .obj
+    WIN32 =
+  endif
+
+  # --- Compilers/Linker & options:
+  FC        = ifl
+  Q2        = '
+  FFLAGS    = $(FDEBUG) $(FOPT) $(SMMP) $(MOD_INC)$(Q2)$(DEDXROOT)$(Q2)
$(MOD_INC)$(Q2)..\src$(Q2) $(MPIO) /FR /w90 /w95 /4Yportlib
+  OBJF    = .obj
+  MOD_INC   = -I
+
+  LIB_SFX =.a
+
+  LD    = $(FC)
+  LDFLAGS = /nologo /link  /heap:83886080 /stack:41943040 /largeaddressaware
+### should make heap & stack options /nodefaultlib:libc
+  OUT   = /exe:# no trailing blanks
+
+  # --- OpenMP threads option.
+  ifeq (omp,$(filter omp,$(CONFIG)))
+    SMMP     = /MT /Qfpp /Qopenmp /Qopenmp_report2
+    DEF_SMMP = -DMULTT
+  endif
+endif
+
 # --- some other setup stuff for CYGWIN
 ECHO  = /usr/bin/echo
 RM    = /usr/bin/rm
diff  -Naurd MCNP5/Source/dotcomm/include/dotcomm.h
MCNP5_new/Source/dotcomm/include/dotcomm.h
--- MCNP5/Source/dotcomm/include/dotcomm.h   2003-04-30 20:08:54.000000000 -0600
+++ MCNP5_new/Source/dotcomm/include/dotcomm.h      2004-07-22 15:06:14.000000000 -0600
@@ -161,11 +161,22 @@

 #include "mpi.h"

-#define DOTCOMM_Datatype   MPI_Datatype
+#define DOTCOMM_Datatype   int

 #define DOTCOMM_Context    MPI_Comm
 #define DOTCOMM_Operation  MPI_Op
```

```
+#ifdef LAM_MPI
+/* ------------------------------------------------------ */
+/* if using LAM-MPI, must add defines for Fortran datatypes */
+/* ------------------------------------------------------ */
+#define MPI_2REAL            ((MPI_Datatype) &lam_mpi_double)
+#define MPI_2INTEGER         ((MPI_Datatype) &lam_mpi_long_long_int)
+#define MPI_REAL             ((MPI_Datatype) &lam_mpi_float)
+#define MPI_INTEGER          ((MPI_Datatype) &lam_mpi_long_int)
+#endif
+
+
   /* Note that Fortran 90 users don't need DOTCOMM datatypes
      since F90 can "discover" types. */

diff  -Naurd MCNP5/Source/dotcomm/src/dotcomm_bcast.F90
MCNP5_new/Source/dotcomm/src/dotcomm_bcast.F90
--- MCNP5/Source/dotcomm/src/dotcomm_bcast.F90      2003-04-30 20:09:00.000000000 -0600
+++ MCNP5_new/Source/dotcomm/src/dotcomm_bcast.F90  2004-07-22 15:06:14.000000000 -0600
@@ -127,7 +127,7 @@

     DOTCOMM_TYPE = DOTCOMM_CHARACTER
     CALL DOTCOMMI_BCAST(  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          ROOT_PE, &
@@ -207,7 +207,7 @@

     DOTCOMM_TYPE = DOTCOMM_INTEGER4
     CALL DOTCOMMI_BCAST(  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          ROOT_PE, &
@@ -287,7 +287,7 @@

     DOTCOMM_TYPE = DOTCOMM_INTEGER8
     CALL DOTCOMMI_BCAST(  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          ROOT_PE, &
@@ -367,7 +367,7 @@

     DOTCOMM_TYPE = DOTCOMM_REAL4
     CALL DOTCOMMI_BCAST(  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          ROOT_PE, &
@@ -447,7 +447,7 @@

     DOTCOMM_TYPE = DOTCOMM_REAL8
     CALL DOTCOMMI_BCAST(  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          ROOT_PE, &
diff  -Naurd MCNP5/Source/dotcomm/src/dotcomm_pack.F90
MCNP5_new/Source/dotcomm/src/dotcomm_pack.F90
--- MCNP5/Source/dotcomm/src/dotcomm_pack.F90       2003-04-30 20:09:06.000000000 -0600
+++ MCNP5_new/Source/dotcomm/src/dotcomm_pack.F90   2004-07-22 15:06:14.000000000 -0600
@@ -121,7 +121,7 @@
        SUBROUTINE DOTCOMMI_PACK( BUF, COUNT, DOTCOMM_TYPE, IERR)
         USE DOTCOMM_CONST_MOD
```

```
          !DEC$ ATTRIBUTES C, REFERENCE,ALIAS:'_DOTCOMMI_PACK' :: DOTCOMMI_PACK
-         CHARACTER(LEN=*), DIMENSION(:), INTENT(IN)  :: BUF
+         CHARACTER(LEN=*),               INTENT(IN)  :: BUF
          INTEGER(IDEF),                  INTENT(IN)  :: COUNT
          INTEGER(IDEF),                  INTENT(OUT) :: IERR
          INTEGER(IDEF) :: DOTCOMM_TYPE
@@ -148,7 +148,7 @@
     DOTCOMM_TYPE = DOTCOMM_CHARACTER

     CALL DOTCOMMI_PACK (  &
-         BUF,                 &
+         BUF(1),                 &
          COUNT, &
          DOTCOMM_TYPE, &
          IERR )
@@ -222,7 +222,7 @@

     DOTCOMM_TYPE = DOTCOMM_INTEGER4
     CALL DOTCOMMI_PACK (  &
-         BUF,                 &
+         BUF(1),                 &
          COUNT, &
          DOTCOMM_TYPE, &
          IERR )
@@ -296,7 +296,7 @@

     DOTCOMM_TYPE = DOTCOMM_INTEGER8
     CALL DOTCOMMI_PACK (  &
-         BUF,                 &
+         BUF(1),                 &
          COUNT, &
          DOTCOMM_TYPE, &
          IERR )
@@ -370,7 +370,7 @@

     DOTCOMM_TYPE = DOTCOMM_REAL4
     CALL DOTCOMMI_PACK (  &
-         BUF,                 &
+         BUF(1),                 &
          COUNT, &
          DOTCOMM_TYPE, &
          IERR )
@@ -444,7 +444,7 @@

     DOTCOMM_TYPE = DOTCOMM_REAL8
     CALL DOTCOMMI_PACK (  &
-         BUF,                 &
+         BUF(1),                 &
          COUNT, &
          DOTCOMM_TYPE, &
          IERR )
diff  -Naurd MCNP5/Source/dotcomm/src/dotcomm_recv.F90
MCNP5_new/Source/dotcomm/src/dotcomm_recv.F90
--- MCNP5/Source/dotcomm/src/dotcomm_recv.F90       2003-04-30 20:09:10.000000000 -0600
+++ MCNP5_new/Source/dotcomm/src/dotcomm_recv.F90   2004-07-22 15:06:14.000000000 -0600
@@ -139,7 +139,7 @@

     DOTCOMM_TYPE = DOTCOMM_CHARACTER
     CALL DOTCOMMI_RECV (  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          SRCPE, &
@@ -225,7 +225,7 @@

     DOTCOMM_TYPE = DOTCOMM_INTEGER4
     CALL DOTCOMMI_RECV (  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          SRCPE, &
@@ -397,7 +397,7 @@

     DOTCOMM_TYPE = DOTCOMM_REAL4
     CALL DOTCOMMI_RECV (  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          SRCPE, &
@@ -483,7 +483,7 @@

     DOTCOMM_TYPE = DOTCOMM_REAL8
     CALL DOTCOMMI_RECV (  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          SRCPE, &
diff  -Naurd MCNP5/Source/dotcomm/src/dotcomm_send.F90
MCNP5_new/Source/dotcomm/src/dotcomm_send.F90
--- MCNP5/Source/dotcomm/src/dotcomm_send.F90       2003-04-30 20:09:10.000000000 -0600
+++ MCNP5_new/Source/dotcomm/src/dotcomm_send.F90   2004-07-22 15:06:14.000000000 -0600
@@ -135,7 +135,7 @@

     DOTCOMM_TYPE = DOTCOMM_CHARACTER
     CALL DOTCOMMI_SEND (  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          DESTPE, &
@@ -221,7 +221,7 @@

     DOTCOMM_TYPE = DOTCOMM_INTEGER4
     CALL DOTCOMMI_SEND (  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          DESTPE, &
@@ -307,7 +307,7 @@

     DOTCOMM_TYPE = DOTCOMM_INTEGER8
     CALL DOTCOMMI_SEND (  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          DESTPE, &
@@ -393,7 +393,7 @@

     DOTCOMM_TYPE = DOTCOMM_REAL4
     CALL DOTCOMMI_SEND (  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          DESTPE, &
@@ -479,7 +479,7 @@

     DOTCOMM_TYPE = DOTCOMM_REAL8
     CALL DOTCOMMI_SEND (  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          DESTPE, &
diff  -Naurd MCNP5/Source/dotcomm/src/dotcomm_unpack.F90
MCNP5_new/Source/dotcomm/src/dotcomm_unpack.F90
```

```
--- MCNP5/Source/dotcomm/src/dotcomm_unpack.F90     2003-04-30 20:09:12.000000000 -0600
+++ MCNP5_new/Source/dotcomm/src/dotcomm_unpack.F90 2004-07-22 15:06:14.000000000 -0600
@@ -119,7 +119,7 @@
        SUBROUTINE DOTCOMMI_UNPACK (BUF, COUNT, DOTCOMM_TYPE, IERR)
         USE DOTCOMM_CONST_MOD
         !DEC$ ATTRIBUTES C, ALIAS:'_DOTCOMMI_UNPACK', REFERENCE :: DOTCOMMI_UNPACK
-        CHARACTER (LEN=*), DIMENSION(:), INTENT(INOUT) :: BUF
+        CHARACTER (LEN=*),               INTENT(INOUT) :: BUF
         INTEGER(IDEF),                  INTENT(IN) :: COUNT
         INTEGER(IDEF),                  INTENT(OUT) :: IERR
         INTEGER(IDEF)  :: DOTCOMM_TYPE
@@ -144,7 +144,7 @@

     DOTCOMM_TYPE = DOTCOMM_CHARACTER
     CALL DOTCOMMI_UNPACK (  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          IERR )
@@ -218,7 +218,7 @@

     DOTCOMM_TYPE = DOTCOMM_INTEGER4
     CALL DOTCOMMI_UNPACK (  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          IERR )
@@ -292,7 +292,7 @@

     DOTCOMM_TYPE = DOTCOMM_INTEGER8
     CALL DOTCOMMI_UNPACK (  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          IERR )
@@ -366,7 +366,7 @@

     DOTCOMM_TYPE = DOTCOMM_REAL4
     CALL DOTCOMMI_UNPACK (  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          IERR )
@@ -440,7 +440,7 @@

     DOTCOMM_TYPE = DOTCOMM_REAL8
     CALL DOTCOMMI_UNPACK (  &
-         BUF, &
+         BUF(1), &
          COUNT, &
          DOTCOMM_TYPE, &
          IERR )
diff  -Naurd MCNP5/Source/dotcomm/src/internals/mpi/dotcommpi_binary_op.c
MCNP5_new/Source/dotcomm/src/internals/mpi/dotcommpi_binary_op.c
--- MCNP5/Source/dotcomm/src/internals/mpi/dotcommpi_binary_op.c   2003-04-30
20:09:32.000000000 -0600
+++ MCNP5_new/Source/dotcomm/src/internals/mpi/dotcommpi_binary_op.c       2004-07-22
15:06:14.000000000 -0600
@@ -3,7 +3,11 @@
 /*                                                                  */
 /* Author: Richard Barrett; rbarrett@lanl.gov                       */
 /* Date: March 2002                                                 */
-
+/*  EDITED FOR LAMPI      SEPost  2-7-03  */
+/*    Switch on operation with switch on datatype within each operation
+      case changed to elseif blocks becasue with LAMPI MPI_Datatype and
+      MPI_Op  (and therefore DOTCOMM_Datatype and DOTCOMM_Operation) are
+      pointers instead of integers and cannot be basis for switches  */
 #include "dotcomm.h"

 /* ==========================================================================
@@ -37,11 +41,11 @@

    operation          Intent:       in
                       C       type: DOTCOMM_Operation
-                      The operation applied to the vector elements: \\
+                      The operation applied to the vector elements:

-                        \hspace*{.2in}{\tt DOTCOMM_OP_MAX}     \hspace*{.54in}  Maximum.
-                        \hspace*{.2in}{\tt DOTCOMM_OP_MIN}     \hspace*{.55in}  Minimum.
-                        \hspace*{.2in}{\tt DOTCOMM_OP_SUM}     \hspace*{.55in}  Sum. \\
+                        DOTCOMM_OP_MAX     Maximum.
+                        DOTCOMM_OP_MIN     Minimum.
+                        DOTCOMM_OP_SUM      Sum.

                       See the User Guide for details.

@@ -106,158 +110,167 @@
   /* ----------------------- */
   /* operate on the elements */
   /* ----------------------- */
-  switch ( operation )
-    {
-
-    case  DOTCOMM_OP_MAX:
-      switch( datatype )
+  /*switch ( operation ) */
+  if (operation == DOTCOMM_OP_MAX) {
+     /* switch( datatype )  */
+     if (datatype == DOTCOMM_REAL8)
        {
-       case DOTCOMM_REAL8:
          for ( i = 0; i < count; i++ )
            {
             ((double*)dest)[i] = DOTCOMM_MAX( ((double*)src_a)[i],
                                               ((double*)src_b)[i] );
           }
-         break;
+       }

-       case DOTCOMM_REAL4:
+    else if (datatype == DOTCOMM_REAL4)
+         {
         for ( i = 0; i < count; i++ )
           {
             ((float*)dest)[i] = DOTCOMM_MAX( ((float*)src_a)[i],
                                             ((float*)src_b)[i] );
           }
-         break;
+        }

-       case DOTCOMM_INTEGER8:
+    else if (datatype == DOTCOMM_INTEGER8)
+         {
         for ( i = 0; i < count; i++ )
           {
             ((long*)dest)[i] = DOTCOMM_MAX( ((long*)src_a)[i],
                                           ((long*)src_b)[i] );
           }
-         break;
+        }

-       case DOTCOMM_INTEGER4:
+     else if (datatype == DOTCOMM_INTEGER4)
+         {
         for ( i = 0; i < count; i++ )
           {
             ((int*)dest)[i] = DOTCOMM_MAX( ((int*)src_a)[i],
                                         ((int*)src_b)[i] );
           }
-        break;
+       }

-      case DOTCOMM_CHARACTER:
+    else if (datatype == DOTCOMM_CHARACTER)
+       {
         for ( i = 0; i < count; i++ )
           {
             ((char*)dest)[i] = DOTCOMM_MAX( ((char*)src_a)[i],
                                          ((char*)src_b)[i] );
           }
-        break;
+      }

-
-      default:
+     else
+       {
         ierr =  DOTCOMM_ERROR_INVALID_DATATYPE;
-        break;
+      }

-      } /* End datatype switch */
-     break;
+       /* End datatype switch */
+     }
+   else if (operation == DOTCOMM_OP_MIN) {

-   case  DOTCOMM_OP_MIN:
-     switch( datatype )
+     /* switch( datatype )*/
+       if (datatype == DOTCOMM_REAL8)
       {
-      case DOTCOMM_REAL8:
         for ( i = 0; i < count; i++ )
           {
             ((double*)dest)[i] = DOTCOMM_MIN( ((double*)src_a)[i],
                                             ((double*)src_b)[i] );
           }
-        break;
+        }

-      case DOTCOMM_REAL4:
+      else if (datatype ==  DOTCOMM_REAL4)
+        {
         for ( i = 0; i < count; i++ )
           {
             ((float*)dest)[i] = DOTCOMM_MIN( ((float*)src_a)[i],
                                            ((float*)src_b)[i] );
           }
-        break;
+       }

-      case DOTCOMM_INTEGER8:
+      else if (datatype == DOTCOMM_INTEGER8)
+          {
         for ( i = 0; i < count; i++ )
           {
             ((long*)dest)[i] = DOTCOMM_MIN( ((long*)src_a)[i],
                                          ((long*)src_b)[i] );
           }
-        break;
+        }

-      case DOTCOMM_INTEGER4:
+      else if (datatype == DOTCOMM_INTEGER4)
+          {
         for ( i = 0; i < count; i++ )
           {
             ((int*)dest)[i] = DOTCOMM_MIN( ((int*)src_a)[i],
                                          ((int*)src_b)[i] );
           }
-        break;
+        }

-       case DOTCOMM_CHARACTER:
+       else if (datatype == DOTCOMM_CHARACTER)
+          {
         for ( i = 0; i < count; i++ )
           {
             ((char*)dest)[i] = DOTCOMM_MIN( ((char*)src_a)[i],
                                           ((char*)src_b)[i] );
           }
-        break;
-
-
+        }

-       default:
+         else
+          {
         ierr =  DOTCOMM_ERROR_INVALID_DATATYPE;
-        break;
+        }

-       } /* End datatype switch */
-       break;
+        /* End datatype switch */
+       }

-     case  DOTCOMM_OP_SUM:
-       switch( datatype )
-       {
-       case DOTCOMM_REAL8:
+     else if (operation == DOTCOMM_OP_SUM) {
+
+       /*  switch( datatype )  */
+       if (datatype == DOTCOMM_REAL8)
+           {
         for ( i = 0; i < count; i++ )
           {
             ((double*)dest)[i] = ((double*)src_a)[i] + ((double*)src_b)[i];
           }
-        break;
+        }

-       case DOTCOMM_REAL4:
+       else if (datatype == DOTCOMM_REAL4)
+           {
         for ( i = 0; i < count; i++ )
           {
             ((float*)dest)[i] = ((float*)src_a)[i] + ((float*)src_b)[i];
           }
-        break;
+        }

-       case DOTCOMM_INTEGER8:
+       else if (datatype == DOTCOMM_INTEGER8)
+          {
         for ( i = 0; i < count; i++ )
           {
             ((long*)dest)[i] = ((long*)src_a)[i] + ((long*)src_b)[i];
           }
-        break;
+        }

-       case DOTCOMM_INTEGER4:
+       else if (datatype == DOTCOMM_INTEGER4)
+          {
         for ( i = 0; i < count; i++ )
```

```
            {
              ((int*)dest)[i] = ((int*)src_a)[i] + ((int*)src_b)[i];
            }
-         break;
+         }

-       case DOTCOMM_CHARACTER:
+       else if (datatype == DOTCOMM_CHARACTER)
+         {
          for ( i = 0; i < count; i++ )
            {
              ((char*)dest)[i] = ((char*)src_a)[i] + ((char*)src_b)[i];
            }
-         break;
+        }

-       default:
+        else
+         {
          return DOTCOMM_ERROR_INVALID_DATATYPE;
-
-       } /* End datatype switch */
-      break;
-
-
-    default:
+           }
+        /* End datatype switch */
+      }
+     else {

       return DOTCOMM_ERROR_INVALID_OPERATION;

diff  -Naurd MCNP5/Source/dotcomm/src/internals/mpi/dotcommpi_convert.c
MCNP5_new/Source/dotcomm/src/internals/mpi/dotcommpi_convert.c
--- MCNP5/Source/dotcomm/src/internals/mpi/dotcommpi_convert.c      2003-04-30
20:09:32.000000000 -0600
+++ MCNP5_new/Source/dotcomm/src/internals/mpi/dotcommpi_convert.c 2004-07-22
15:06:14.000000000 -0600
@@ -3,6 +3,10 @@
 /*                                                                    */
 /* Author: Richard Barrett; rbarrett@lanl.gov                         */
 /* Date: March 2002                                                   */
+/* EDITED FOR LAMPI SEPost   2-7-03
+   Switch on dotcomm_op changed to elseif because with LAMPI
+   MPI_Datatype and therefore DOTCOMM_Datatype are pointers instead
+   of integers.  Cannot switch on *dotcomm_op                 */

 #include "dotcomm.h"

@@ -34,7 +38,7 @@
    conversion           Intent:       in
                           C       type: const DOTCOMM_Conversions
                           The type of conversion.
-                          See DOTCOMM_Conversions (section \ref{DOTCOMM\\_Convert\_enum}
page \pageref{DOTCOMM\\_Convert\_enum})
+                          See DOTCOMM_Conversions
                           for a listing of the possible conversion types.

    Return Values
@@ -77,7 +81,7 @@
   DOTCOMM_Operation
     *protocol_op;           /* protocol operation    */

-  DOTCOMM_Datatype
+  MPI_Datatype
     *protocol_datatype;     /* protocol datatype     */

   DOTCOMM_Operation
@@ -102,7 +106,7 @@
     case DOTCOMM_TO_PROTOCOL_DATATYPE:
       dotcomm_datatype      = (DOTCOMM_Datatype*)in;
-      protocol_datatype = (DOTCOMM_Datatype*)out;
+      protocol_datatype = (MPI_Datatype*)out;

       if( *dotcomm_datatype == DOTCOMM_REAL8 )
         {
@@ -194,28 +198,25 @@
       dotcomm_op  = (DOTCOMM_Operation*)in;
       protocol_op = (DOTCOMM_Operation*)out;

-      switch( *dotcomm_op )
-       {
-       case DOTCOMM_OP_MAX:
+      /* switch( *dotcomm_op )   */
+
+     if (*dotcomm_op == DOTCOMM_OP_MAX)
         *protocol_op = MPI_MAX;
-        break;

-      case DOTCOMM_OP_MIN:
+     else if (*dotcomm_op == DOTCOMM_OP_MIN)
         *protocol_op = MPI_MIN;
-        break;

-      case DOTCOMM_OP_SUM:
+     else if (*dotcomm_op ==  DOTCOMM_OP_SUM)
         *protocol_op = MPI_SUM;
-        break;

-      default:
+     else
+       {
           ierr = DOTCOMM_ERROR_BAD_PARAMETER;
          DOTCOMMP_ASSERT ( (ierr == DOTCOMM_OK), "bad parameter",
                            DOTCOMM_ERROR_BAD_PARAMETER );
-        break;
         }
-      break;
-
+
+     break;
       /* --------------------------- */
       /* DONE: DOTCOMM to protocol op */
       /* --------------------------- */
diff  -Naurd MCNP5/Source/dotcomm/src/internals/mpi/dotcommpi_mpi.h
MCNP5_new/Source/dotcomm/src/internals/mpi/dotcommpi_mpi.h
--- MCNP5/Source/dotcomm/src/internals/mpi/dotcommpi_mpi.h  2003-04-30 20:09:32.000000000
-0600
+++ MCNP5_new/Source/dotcomm/src/internals/mpi/dotcommpi_mpi.h     2004-07-22
15:06:14.000000000 -0600
@@ -32,7 +32,7 @@
                        DOTCOMM_Conversions   conversion
                        );

-  int dotcommpi_sizeof ( DOTCOMM_Datatype datatype );
+  int dotcommpi_sizeof ( int datatype );

   /* ------------------------------------------------ */
   /* remove typesafe linkage if compiling under c++ */
diff  -Naurd MCNP5/Source/dotcomm/src/internals/mpi/dotcommpi_sizeof.c
MCNP5_new/Source/dotcomm/src/internals/mpi/dotcommpi_sizeof.c
--- MCNP5/Source/dotcomm/src/internals/mpi/dotcommpi_sizeof.c      2003-04-30
20:09:32.000000000 -0600
+++ MCNP5_new/Source/dotcomm/src/internals/mpi/dotcommpi_sizeof.c   2004-07-22
15:06:14.000000000 -0600
@@ -37,7 +37,7 @@

 #define DOTCOMM_LOCATION "dotcommpi_sizeof"

-int dotcommpi_sizeof ( DOTCOMM_Datatype datatype )
```

```
+int dotcommpi_sizeof ( int datatype )
 {

   /* ------------------ */
@@ -53,11 +53,11 @@

     if ( datatype == DOTCOMM_REAL8 )

-      sizeof_datatype = sizeof(double);
+      sizeof_datatype = 8;

     else if ( datatype == DOTCOMM_REAL4 )

-      sizeof_datatype = sizeof(float);
+      sizeof_datatype = 4;

     else if ( datatype == DOTCOMM_INTEGER8 )

@@ -66,11 +66,11 @@

     else if ( datatype == DOTCOMM_INTEGER4 )

-      sizeof_datatype = sizeof(int);
+      sizeof_datatype = 4;

     else if ( datatype == DOTCOMM_CHARACTER )

-      sizeof_datatype = sizeof(char);
+      sizeof_datatype = 1;

     else
       {
diff  -Naurd MCNP5/Source/install MCNP5_new/Source/install
--- MCNP5/Source/install      2003-11-05 17:23:12.000000000 -0700
+++ MCNP5_new/Source/install  2004-07-23 15:49:01.000000000 -0600
@@ -236,32 +236,53 @@
   if [ "$GCC" = '' ]
   then GCC=''
   fi
-  if [ "$PGI" != '' ]
-  then {
-    menuf90=portland
-    menuf90path="$PGI/linux86/bin/pgf90"
-    menuccpath="$GCC"
-  }
-  elif [ -d "/usr/pgi" ]
-  then {
-    menuf90=portland
-    PGI=/usr/pgi
-    menuf90path="$PGI/linux86/bin/pgf90"
-    menuccpath="$GCC"
-  }
-  elif [ -d "/usr/local/pgi" ]
-  then {
+
+  menuf90path=`which pgf90`
+
+  if [ "$menuf90path" = '' ]
+  then
+    if [ "$PGI" = '' ]
+    then
+      if [ -d   "/usr/pgi-5.2" ]
+        then PGI=/usr/pgi-5.2
+      elif [ -d "/usr/local/pgi-5.2" ]
+        then PGI=/usr/local/pgi-5.2
+      elif [ -d "/opt/pgi-5.2" ]
+        then PGI=/opt/pgi-5.2
+      elif [ -d "/usr/pgi-5.1" ]
+        then PGI=/usr/pgi-5.1
+      elif [ -d "/usr/local/pgi-5.1" ]
+        then PGI=/usr/local/pgi-5.1
+      elif [ -d "/opt/pgi-5.1" ]
+        then PGI=/opt/pgi-5.1
+      elif [ -d "/usr/pgi" ]
+        then PGI=/usr/pgi
+      elif [ -d "/usr/local/pgi" ]
+        then PGI=/usr/local/pgi
+      elif [ -d "/opt/pgi" ]
+        then PGI=/opt/pgi
+      fi
+    fi
+
+    if [ -d "$PGI/linux86/5.2/bin" ]
+    then
+      menuf90path="$PGI/linux86/5.2/bin/pgf90"
+    elif [ -d "$PGI/linux86/5.1/bin" ]
+    then
+      menuf90path="$PGI/linux86/5.1/bin/pgf90"
+    elif [ -d "$PGI/linux86/bin/pgf90" ]
+    then
+      menuf90path="$PGI/linux86/bin/pgf90"
+    fi
+  fi
+
+  if [ "$menuf90path" != '' ]
+  then
     menuf90=portland
-    PGI=/usr/local/pgi
-    menuf90path="$PGI/linux86/bin/pgf90"
-    menuccpath="$GCC"
-  }
-  else
-    menuf90path=`which pgf90`
     menuccpath="$GCC"
-    if [ "$menuf90path" != '' ] && [ "$menuccpath" != '' ]
-    then menuf90=portland
-    fi
+    export menuf90
+    export menuf90path
+    export menuccpath
   fi
 }
 #--------------------
@@ -336,43 +357,47 @@
 # Use standard locations to identify a Fortran 95 compiler.
 # Identify a C compiler to go with it.
 FindINTELCompiler () {
-  ARCH=`uname -m`
-  if [ "$ARCH" = 'ia64' ]
-  then {
-    ARCH="ia64"
-    INTEL_FC="efc"
-  }
-  else
-    ARCH="ia32"
-    INTEL_FC="ifc"
-  fi
-
   GCC=`which gcc`
   if [ "$GCC" = '' ]
   then GCC=''
   fi
-  if [ -d "/opt/intel/compiler70" ]
-  then {
-    menuf90=intel
-    menuf90path="/opt/intel/compiler70/$ARCH/bin/$INTEL_FC"
-    menuccpath="$GCC"
-  }
-  else
+
+  menuf90path=`which ifort`
+
+  if [ "$menuf90path" = '' ]
+  then
+    ARCH=`uname -m`
     if [ "$ARCH" = 'ia64' ]
-    then {
+    then
       menuf90path=`which efc`
-    }
+      ARCH="ia64"
+      INTEL_FC="efc"
     else
       menuf90path=`which ifc`
+      ARCH="ia32"
+      INTEL_FC="ifc"
     fi
-    export menuf90path
-    if [ "$menuf90path" != '' ]
-    then {
-      menuf90=intel
-      menuccpath="$GCC"
-    }
+  fi
+
+  if [ "$menuf90path" = '' ]
+  then
+    if [ -d "/opt/intel_fc_80/bin" ]
+    then
+      menuf90path="/opt/intel_fc_80/bin/ifort"
+    elif [ -d "/opt/intel/compiler70" ]
+    then
+      menuf90path="/opt/intel/compiler70/$ARCH/bin/$INTEL_FC"
     fi
   fi
+
+  if [ "$menuf90path" != '' ]
+  then
+    menuf90=intel
+    menuccpath="$GCC"
+    export menuf90
+    export menuf90path
+    export menuccpath
+  fi
 }
 #-------------------
 # FindCompaqCompiler
diff  -Naurd MCNP5/Source/src/acecas.F90 MCNP5_new/Source/src/acecas.F90
--- MCNP5/Source/src/acecas.F90      2003-04-30 20:10:00.000000000 -0600
+++ MCNP5_new/Source/src/acecas.F90  2004-07-22 15:14:40.000000000 -0600
@@ -48,7 +48,7 @@
   ! use the selected law to sample the energy (and possibly angle).
   ! if law samples without error, go to sample angle or coordinate
   ! transform as appropriate.
-  colout(1,ls) = -huge
+  colout(1,ls) = -huge_float
   lw = nint(xss(n))
   iw = id-1+nint(xss(n+1))

@@ -415,7 +415,7 @@
   ! ***********************************************************************
   ! print debug information for cross-section table errors.
 295 continue
-  colout(1,ls) = huge
+  colout(1,ls) = huge_float

 300 continue
   call zaid(2,ht,ixl(1,iex))
@@ -427,11 +427,11 @@
     & 5x, "law =",i3,5x,  "energy out =",1pe12.4)
   !$OMP END CRITICAL (PRINT_OUTPUT)

-  if(     colout(1,ls)==-huge ) then
```

```
+  if(     colout(1,ls)==-huge_float ) then
     call expirx(1,'acecas','an inappropriate or non-existent law was selected.')
   elseif( colout(1,ls)<0.      ) then
     call expirx(1,'acecas','emission energy was negative.')
-  elseif( colout(1,ls)==huge  ) then
+  elseif( colout(1,ls)==huge_float  ) then
     call expirx(1,'acecas','faulty cross-section data.')
   elseif( colout(1,ls)>erg    ) then
     call expirx(1,'acecas','emission energy exceeds incident energy.')
diff  -Naurd MCNP5/Source/src/acecos.F90 MCNP5_new/Source/src/acecos.F90
--- MCNP5/Source/src/acecos.F90      2003-04-30 20:10:00.000000000 -0600
+++ MCNP5_new/Source/src/acecos.F90  2004-07-22 15:14:40.000000000 -0600
@@ -20,7 +20,7 @@
   ! find the cosine table by binary search on the energy table.
   if( ka==0 ) then
     ! isotropic case
-    acecos = 2.*rang()-1.
+    acecos = 2.0_dknd*rang()-one
     ixcos = 0
     return
   endif
@@ -38,32 +38,32 @@
         ib = ih
       endif
     enddo
-    !
+    !
    ! sample between adjoining tables by interpolation fraction.
    if( rang()*(xss(ib)-xss(ic)) < erg-xss(ic) )  ic=ib
    lm = nint(xss(ic+n))
    if( lm==0 ) then
      ! isotropic case
-      acecos = 2.*rang()-1.
+      acecos = 2.0_dknd*rang()-one
      ixcos = 0
      return
    elseif( lm>0 ) then
      ! sample from table of 32 equiprobable cosine groups.
-      t1 = rang()*32.
+      t1 = rang()*32._dknd
      kr = t1
      ixcos = ia-1+lm
      acecos = xss(ixcos+kr)+(t1-kr)*(xss(ixcos+kr+1)-xss(ixcos+kr))
      return
    endif
  endif
-
+
  ! tabular probability angular distribution.
  k = ia-1-lm
  ixcos = -k
  jj = nint(xss(k))
  np = nint(xss(k+1))
  rn = rang()
-  !
+  !
  ! binary search of cumulative density function.
  ic = k+2*np+2
  ib = k+3*np+1
@@ -76,12 +76,17 @@
      ib = ih
    endif
  enddo
-  !
+  !
  fa = xss(ic-np)
  ca = xss(ic-2*np)
+  if ( jj /= 1 ) then
+    bb = (xss(ic-np+1)-fa)/(xss(ic-2*np+1)-ca)
+    if( bb /= zero ) then
+      acecos=ca+(sqrt(max(zero,fa**2+2.0_dknd*bb*(rn-xss(ic))))-fa)/bb
+      return
+    endif
+  endif
   acecos = ca+(rn-xss(ic))/fa
-  if( jj==1 )  return
-  bb = (xss(ic-np+1)-fa)/(xss(ic-2*np+1)-ca)
-  if( bb/=0. )  acecos=ca+(sqrt(max(zero,fa**2+2.*bb*(rn-xss(ic))))-fa)/bb
   return
+
 end function acecos
diff  -Naurd MCNP5/Source/src/acegam.F90 MCNP5_new/Source/src/acegam.F90
--- MCNP5/Source/src/acegam.F90       2003-04-30 20:10:04.000000000 -0600
+++ MCNP5_new/Source/src/acegam.F90   2004-07-22 15:14:40.000000000 -0600
@@ -4,6 +4,7 @@
 subroutine acegam
   ! generate and bank photons from a neutron collision.
   use mcnp_global
+  use dxtran_mod
  use mcnp_debug
   implicit real(dknd) (a-h,o-z)
   character(len=10) :: ht
@@ -85,7 +86,7 @@
   elseif( nww(2)==0 ) then
     t1 = max(gwt(icl),-gwt(icl)*wgt9(1))*fiml9(1,1)/(fiml(1)*sf)
   else
-    t1 = huge
+    t1 = huge_float
    sw = wtfasv              ! use neutron wtfasv if erg or erg/tme photon imp
    if ( iets(2)==0 ) then
      sf = 1.
@@ -143,11 +144,7 @@
   es = erg
   vel = slite
   ncp = 0
-  idx = 0
-  do i = 1,ndx(2)
-    if( (xxx-dxx(2,1,i))**2+(yyy-dxx(2,2,i))**2+(zzz-dxx(2,3,i))**2<&
-      & dxx(2,5,i))  idx = i
-  enddo
+  idx = inside_dxtran_sphere()
   st = totm
   if( mcal/=2 )  wtfasv = 1.

diff  -Naurd MCNP5/Source/src/acetot.F90 MCNP5_new/Source/src/acetot.F90
--- MCNP5/Source/src/acetot.F90       2003-04-30 20:10:06.000000000 -0600
+++ MCNP5_new/Source/src/acetot.F90   2004-07-22 15:14:40.000000000 -0600
@@ -6,6 +6,7 @@
   ! mm=1 for call from wtmult, otherwise zero.
   use mcnp_global
   use mcnp_debug
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/addtfc.F90 MCNP5_new/Source/src/addtfc.F90
--- MCNP5/Source/src/addtfc.F90       2003-04-30 20:10:06.000000000 -0600
+++ MCNP5_new/Source/src/addtfc.F90   2004-07-22 15:14:40.000000000 -0600
@@ -30,15 +30,15 @@

   ! calculate the entries in the new line.
   ft = 0.
-  t = max(1,nps)
+  t = max(1_i8knd,nps)
   if( nsr==71 .and. kcz>=ikz   )  t=nsrck*(kcz-ikz)
   call ra_kcheck( kcheck     )
   call ra_lcheck( lsav, ldif )
   if( kcheck>0 .and. kcz>=lsav )  t=nsrck*(kcz-lsav)
   if( kc8 < 0   )  t=t+nsrck-wt0*nsa
   if( knrm /= 0 )  t=pax(1,1,1)
-  if( nsr == 6.and.nrrs >= nrss )  t=max(1,np1)
-  if( nsr == 6.and.nrrs < nrss  )  t=max(1,npsr)
```

```
+  if( nsr == 6.and.nrrs >= nrss )  t=max(1_i8knd,np1)
+  if( nsr == 6.and.nrrs < nrss  )  t=max(1_i8knd,npsr)
   if( t > .5 )  ft=1./t
   npc(l)=nps

@@ -48,18 +48,18 @@
     DO_140_1: do ital_tmp=1,ntal
       ital = ital_tmp
       it = ital+iper*ntal
-      tfc(1:6,l,it) = 0.
-      nhsd(ln,it) = 0
+      tfc(1:6,l,it) = zero
+      nhsd(ln,it) = 0_i8knd
       k = jptal(7,ital)+iper*mxfp
       t = tal(k+mxf)
-      if( t==0. .or. ft==0. .or. tal(k+2*mxf)<=0. )  cycle DO_140_1
+      if( t==0. .or. ft==0. .or. tal(k+2*mxf)<=zero )  cycle DO_140_1
       tfc(2,l,it) = min(tal(k+2*mxf)/t**2-ft,one)
-      if( tfc(2,l,it) <= 1.e-8 )  tfc(2,l,it)=0.
+      if( tfc(2,l,it) <= 1.e-8_dknd )  tfc(2,l,it)=zero
       tfc(2,l,it) = sqrt(tfc(2,l,it))
       if( iptal(4,2,ital)/=0 )  t=t/tds(iptal(4,2,ital)+&
         & iptal(4,3,ital)*(jtf(1,ital)-1)+jtf(4,ital))
       if( (jptal(2,ital)==6.or.jptal(2,ital)==7) .and. &
-        &  jptal(4,ital)/=0 )  t=t*1.60219e-22
+        &  jptal(4,ital)/=0 )  t=t*1.60219e-22_dknd
       tfc(1,l,it) = t*ft
       tfc(3,l,it) = (cts-cpk)*tfc(2,l,it)**2
       ! calculate and store the variance of the variance for tfc bin.
@@ -74,12 +74,12 @@
           if( tfc(3,l,it)/=0. )  tfc(3,l,it)=60./tfc(3,l,it)
         endif
       endif
-      if( tfc(2,l,it)==0. )  tfc(3,l,it)=1.e30
+      if( tfc(2,l,it)==0. )  tfc(3,l,it)=1.e+30_dknd

      ! calculate slope (using a pareto) of tail of history tally pdf.
      ! the tail is defined as the extreme 5% with a minimum of 25
      ! history tallies for analysis to fit the pareto parameters.
-     if(nhsd(ln-3,it)-nhsd(1,it)-nhsd(ln-5,it) < 500)  cycle DO_140_1
+     if(nhsd(ln-3,it)-nhsd(1,it)-nhsd(ln-5,it) < 500_i8knd)  cycle DO_140_1

      ! check for the appearance of a bounded extreme tally value.
      if( shsd(lp,it)/shsd(lp-100,it) < 1.01 )  tfc(5,l,it)=10.
@@ -125,7 +125,7 @@
      ! store np and ts for later use in the prints.
 130   continue
      lk = ls+ks(ms)
-     nhsd(ln,it) = np
+     nhsd(ln,it) = int(np,i8knd)
      shsd(lp+2,it) = ts

      ! find pareto parameters that best fit the np extreme tallies.
diff  -Naurd MCNP5/Source/src/avrwgi.F90 MCNP5_new/Source/src/avrwgi.F90
--- MCNP5/Source/src/avrwgi.F90      2003-04-30 20:10:10.000000000 -0600
+++ MCNP5_new/Source/src/avrwgi.F90  2004-07-22 15:14:40.000000000 -0600
@@ -5,6 +5,7 @@
   ! initialize weight window generator.
   use mcnp_global
   use mcnp_debug
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

@@ -83,7 +84,7 @@
     if( ngww(i)>1  )  ig = 1
     if( wwg(9)/=0. )  cycle
     if( ngww(i)==1 )  ewwg(1+mgww(i)) = 100.
-    if( ngww(i)==1 .and. emx(i)<huge .and. emx(i)>100. )  ewwg(1+mgww(i)) = emx(i)
+    if( ngww(i)==1 .and. emx(i)<huge_float .and. emx(i)>100. )  ewwg(1+mgww(i)) = emx(i)
     if( ngww(i)==1 .and. wwg(8)/=0. )  ewwg(1+mgww(i)) = tco(i)
   enddo

diff  -Naurd MCNP5/Source/src/avrwwg.F90 MCNP5_new/Source/src/avrwwg.F90
--- MCNP5/Source/src/avrwwg.F90      2003-04-30 20:10:12.000000000 -0600
+++ MCNP5_new/Source/src/avrwwg.F90   2004-07-22 15:14:40.000000000 -0600
@@ -142,7 +142,7 @@
   if( nwgeoa==1 ) then
     do i = 1,3
       na = i
-      dr(i) = huge
+      dr(i) = huge_float
       if( dd(i)>0. )  dr(i) = (asm(na,mi(i))-dc(i))/vc(i)
       if( dd(i)<0. )  dr(i) = (asm(na,mi(i)-1)-dc(i))/vc(i)
     enddo
@@ -173,7 +173,7 @@
   ! radial (r)
   na = 1
   rv = vc(1)*dc(1)+vc(2)*dc(2)
-  dr(1) = huge
+  dr(1) = huge_float
   if( v/=0. ) then
     rs = dc(1)**2+dc(2)**2
     a = rv/v
@@ -194,7 +194,7 @@
 180 continue
   ! axial (z)
   na = 2
-  dr(2) = huge
+  dr(2) = huge_float
   if( vc(3)/=0. ) then
     if( vc(3)>=0. )  then
       dr(2) = (asm(na,mi(2))-dc(3))/vc(3)
@@ -207,7 +207,7 @@

   ! azimuthal (theta)
   ! if single azimuthal bin (ism(3)=2), point is always in it.
-  dr(3) = huge
+  dr(3) = huge_float
   if( ism(3)/=2 ) then
     na = 3
     rp = -vc(1)*dc(2)+vc(2)*dc(1)
@@ -219,7 +219,7 @@
       if( vn<0. ) then
         dr(3) = (st*dc(1)-ct*dc(2))/vn
        nf(3) = -1
-        if( dr(3)<=0. )  dr(3) = huge
+        if( dr(3)<=0. )  dr(3) = huge_float
       endif
     else
       t = asm(na,mi(3))
@@ -229,7 +229,7 @@
       if( vn>0. ) then
         dr(3) = (st*dc(1)-ct*dc(2))/vn
         nf(3) = 1
-        if( dr(3)<=0. )  dr(3) = huge
+        if( dr(3)<=0. )  dr(3) = huge_float
       endif
     endif
   endif
diff  -Naurd MCNP5/Source/src/avrxyz.F90 MCNP5_new/Source/src/avrxyz.F90
--- MCNP5/Source/src/avrxyz.F90      2003-04-30 20:10:12.000000000 -0600
+++ MCNP5_new/Source/src/avrxyz.F90   2004-07-22 15:14:40.000000000 -0600
@@ -9,6 +9,7 @@

   use mcnp_global
   use mcnp_debug
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/bankit.F90 MCNP5_new/Source/src/bankit.F90
```

```
--- MCNP5/Source/src/bankit.F90      2003-11-05 17:23:12.000000000 -0700
+++ MCNP5_new/Source/src/bankit.F90  2004-07-22 15:14:40.000000000 -0600
@@ -54,7 +54,7 @@
         write(iuo,30) npstc
         if( ntasks<=1 .and. ltasks<=1 ) then
           write(jtty,30) npstc
-30        format(  " bank is full.  bank backup file is being created.  nps =",i10)
+30        format(  " bank is full.  bank backup file is being created.  nps =",i12)
         endif
         open(iub+ktask,form='unformatted',status='scratch')
       endif
@@ -63,9 +63,9 @@
       if( lsb==0 ) then
         lsb = 1
 
-        !$OMP ATOMIC
+        !$OMP CRITICAL       (UPDATE_VARCOM)
         nbov = nbov+1
-
+        !$OMP END CRITICAL   (UPDATE_VARCOM)
       endif
 
       ! Write the first half (one block) of bank to the backup file.
diff  -Naurd MCNP5/Source/src/brems.F90 MCNP5_new/Source/src/brems.F90
--- MCNP5/Source/src/brems.F90       2003-04-30 20:10:18.000000000 -0600
+++ MCNP5_new/Source/src/brems.F90   2004-07-22 15:14:40.000000000 -0600
@@ -9,6 +9,7 @@
   ! ae = deflection cosine of electron over path segment d.
 
   use mcnp_global
+  use dxtran_mod
   use mcnp_debug
 
   implicit real(dknd) (a-h,o-z)
@@ -173,12 +174,8 @@
     ! bank the photon.
     vel = slite
     ncp = 0
-    idx = 0
+    idx = inside_dxtran_sphere()
    jsu = 0
-    do i=1,ndx(2)
-      if( (xxx-dxx(2,1,i))**2+(yyy-dxx(2,2,i))**2+&
-        & (zzz-dxx(2,3,i))**2 < dxx(2,5,i) )  idx=i
-    end do
    if( dbcn(20)/=0. ) then
      call bankit(16)
    endif
diff  -Naurd MCNP5/Source/src/calcva.F90 MCNP5_new/Source/src/calcva.F90
--- MCNP5/Source/src/calcva.F90      2003-11-05 17:23:12.000000000 -0700
+++ MCNP5_new/Source/src/calcva.F90  2004-07-22 15:14:40.000000000 -0600
@@ -32,6 +32,7 @@
    endif
  endif
 
+
  ! Do each appropriate surface of the current cell.
  vt = 0.
  ncrn = 0
diff  -Naurd MCNP5/Source/src/celnbr.F90 MCNP5_new/Source/src/celnbr.F90
--- MCNP5/Source/src/celnbr.F90      2003-04-30 20:10:20.000000000 -0600
+++ MCNP5_new/Source/src/celnbr.F90  2004-07-22 15:14:40.000000000 -0600
@@ -16,7 +16,7 @@
 
  implicit real(dknd) (a-h,o-z)
 
-  character(len=10) :: ha
+  character(len=80) :: ha
  integer :: ii(3)
  ! Statement function: moved to end
```

```
diff  -Naurd MCNP5/Source/src/celpar.F90 MCNP5_new/Source/src/celpar.F90
--- MCNP5/Source/src/celpar.F90       2003-04-30 20:10:20.000000000 -0600
+++ MCNP5_new/Source/src/celpar.F90   2004-07-22 15:14:40.000000000 -0600
@@ -7,6 +7,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/celsrf.F90 MCNP5_new/Source/src/celsrf.F90
--- MCNP5/Source/src/celsrf.F90       2003-04-30 20:10:22.000000000 -0600
+++ MCNP5_new/Source/src/celsrf.F90   2004-07-22 15:14:40.000000000 -0600
@@ -9,6 +9,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)
   character hj(2,3)*10, hl*5, hp*130, hq*2, ht*4
@@ -51,7 +52,7 @@
     endif
     do ic=1,mxa
       do j=1,ndx(i)
-        if( dxcp(j,i,ic)==huge )  dxcp(j,i,ic)=dxcp(0,i,ic)
+        if( dxcp(j,i,ic)==huge_float )  dxcp(j,i,ic)=dxcp(0,i,ic)
       end do
       do j=0,ndx(i)
         dxcp(j,i,ic) = min(max(zero,dxcp(j,i,ic)),one)
@@ -334,13 +335,13 @@
   if( mxt>1 )  go to 518

   ! If one time, check if all the cell temperatures are the same.
-  tp = -huge
+  tp = -huge_float
   do ic=1,mxa
     if( fim(1,ic)==0. .or. mat(ic)==0 )  cycle
-    if( tp==-huge )  tp=tmp(ic)
+    if( tp==-huge_float )  tp=tmp(ic)
     if( tp/=tmp(ic) )  go to 518
   end do
-  if( tp==-huge ) then
+  if( tp==-huge_float ) then
     write(iuo,512)
 512 format(/  "   all materials are in zero importance cells.")
     return
diff  -Naurd MCNP5/Source/src/chekcs.F90 MCNP5_new/Source/src/chekcs.F90
--- MCNP5/Source/src/chekcs.F90       2003-04-30 20:10:22.000000000 -0600
+++ MCNP5_new/Source/src/chekcs.F90   2004-07-22 15:14:40.000000000 -0600
@@ -6,6 +6,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)
   integer :: iu(0:mxlv), il(0:mxlv), mu(0:mxlv), jm(0:mxlv), nu(0:mxlv)
diff  -Naurd MCNP5/Source/src/chekit.F90 MCNP5_new/Source/src/chekit.F90
--- MCNP5/Source/src/chekit.F90       2003-04-30 20:10:22.000000000 -0600
+++ MCNP5_new/Source/src/chekit.F90   2004-07-22 15:14:40.000000000 -0600
@@ -11,6 +11,7 @@
   use mcnp_debug
   use mcnp_input
   use fmesh_mod
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)
   character(len=10) :: hs
@@ -655,7 +656,7 @@
```

```
   case( 60 )
     !  >>>>>  source particle cutoff number                               nps
-    if( nsr==71 .and. iitm>0 )  call erprnt(2,2,0,0,0,0,0,1,&
+    if( nsr==71 .and. i8itm>0_i8knd )  call erprnt(2,2,0,0,0,0,0,1,&
       & '  "nps card is ineffective in kcode problems."')

   case( 66 )
@@ -913,6 +914,7 @@
       if(hptr(m1c)=='event' .or. hptr(m1c)=='file' .or. hptr(m1c)=='type')  go to 9010
       if(hptr(m1c)/='value' .and. hptr(m1c)/='tally' .and. &
         &   hptr(m1c)/='filter' .and. hptr(m1c)/='max' .and. iitm<0)  go to 9010
+      if(hptr(m1c)=='max' .and. i8itm == 0_i8knd ) go to 9010
       if(hptr(m1c)=='cell' .and. namchg(1,iitm)==0)&
         & call erprnt(2,1,1,iitm,0,0,0,0,' "ptrac cell entry ",i5," is not a valid
cell."')
       if(hptr(m1c)=='surface' .and. namchg(2,iitm)==0) call erprnt(2,1,1,iitm,0,0,0,0,&
diff  -Naurd MCNP5/Source/src/chektr.F90 MCNP5_new/Source/src/chektr.F90
--- MCNP5/Source/src/chektr.F90       2003-04-30 20:10:24.000000000 -0600
+++ MCNP5_new/Source/src/chektr.F90   2004-07-22 15:14:40.000000000 -0600
@@ -7,6 +7,8 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod
+  use fmesh_mod

   implicit real(dknd) (a-h,o-z)

@@ -46,6 +48,11 @@
         if( mfl(3,j)==m .or. ktr(j)==i )  cycle DO_305
       end do
     endif
+
+    do j=1,nmesh
+      if( fm(j)%itr == i ) cycle DO_305
+    enddo
+
    call erprnt(1,2,1,i,0,0,0,0,' "tr",i3,  " card unused."')
   end do DO_305

diff  -Naurd MCNP5/Source/src/colinp.F90 MCNP5_new/Source/src/colinp.F90
--- MCNP5/Source/src/colinp.F90       2003-04-30 20:10:28.000000000 -0600
+++ MCNP5_new/Source/src/colinp.F90   2004-07-22 15:14:41.000000000 -0600
@@ -8,6 +8,8 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod
+

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/collpn.F90 MCNP5_new/Source/src/collpn.F90
--- MCNP5/Source/src/collpn.F90       2003-04-30 20:10:28.000000000 -0600
+++ MCNP5_new/Source/src/collpn.F90   2004-07-22 15:14:41.000000000 -0600
@@ -4,6 +4,7 @@
 subroutine collpn
   ! generate and bank particles from a photonuclear collision.
   use mcnp_global
+  use dxtran_mod
   use mcnp_debug

   implicit real(dknd) (a-h,o-z)
@@ -93,11 +94,7 @@
     wz = wg

     ! check if new particle is inside a dxtran sphere.
-    idx = 0
-    do nd=1,ndx(ipt)
-      if( (xxx-dxx(ipt,1,nd))**2+(yyy-dxx(ipt,2,nd))**2+&
-        & (zzz-dxx(ipt,3,nd))**2 < dxx(ipt,5,nd) )  idx=nd
-    end do
+    idx = inside_dxtran_sphere()

     ! the miminum weight is set by the dxtran weight cutoff.
     if( idx/=0 .and. wwp(ipt,5)>=0. ) then
@@ -120,7 +117,7 @@
       cycle DO_230
     endif
     ! minimum positive energy-dependent windows.
-    wl = huge
+    wl = huge_float
     wp = -1.
     do i=1,nww(ipt)
       wv = wwval(ipt,icl,i,1,ix)
diff  -Naurd MCNP5/Source/src/crit1_mod.F90 MCNP5_new/Source/src/crit1_mod.F90
--- MCNP5/Source/src/crit1_mod.F90   2003-04-30 20:10:32.000000000 -0600
+++ MCNP5_new/Source/src/crit1_mod.F90       2004-07-22 15:14:41.000000000 -0600
@@ -25,8 +25,9 @@
   use mcnp_input
   use ra1_mod
   use ra2_mod
-
-  implicit real(dknd) (a-h,o-z)
+  use erprnt_mod, only: erprnt
+
+ implicit real(dknd) (a-h,o-z)

 contains

@@ -326,8 +327,8 @@
     &   " read from the srctp file named ",a8, ".")

   ! print 3 lines of information about the keff calculation.
-  write(iuo,50) ikz,kct,nsrck,min(kcz,ikz),min(nps,nskk),&
-    & max(0,kcz-ikz),max(0,nps-nskk)
+  write(iuo,50) ikz,kct,nsrck,min(kcz,ikz),min(nps,int(nskk,i8knd)),&
+    & max(0,kcz-ikz),max(0_i8knd,nps-int(nskk,i8knd))
 50 format(  " the criticality problem was scheduled to skip",i4,&
     &   " cycles and run a total of",i5,  " cycles with nominally",i9,&
     &   " neutrons per cycle."/  " this problem has run",i4,&
@@ -543,8 +544,8 @@

   ! find largest and smallest of each active cycle keff estimator.
   do j = 1,3
-    sk = huge
-    bk = -huge
+    sk = huge_float
+    bk = -huge_float
     do i = ikz+1,mk
       if( rkpl(j,i)>bk ) then
         bk = rkpl(j,i)
diff  -Naurd MCNP5/Source/src/crit2_mod.F90 MCNP5_new/Source/src/crit2_mod.F90
--- MCNP5/Source/src/crit2_mod.F90   2003-04-30 20:10:32.000000000 -0600
+++ MCNP5_new/Source/src/crit2_mod.F90       2004-07-22 15:14:41.000000000 -0600
@@ -27,6 +27,7 @@
   use mcnp_debug
   use mcnp_input
   use ra2_mod
+  use erprnt_mod

  implicit real(dknd) (a-h,o-z)

@@ -558,6 +559,7 @@
   character(len=1)  :: hs
   real(dknd) :: ci(2,3,3),cg(3),ea(4),os(3),o2(3,3),za(4),zk(3,1)
   integer    :: nf(3)
+  integer(i8knd) :: ns

   if( nw/=2 ) then
     ! set up for calculation of the keffs with the largest values.
```

```
@@ -594,8 +596,8 @@
       enddo
     enddo
   enddo
-  ns = nps+nsa-nint(rkpl(18,kcz+1))
-  sd = huge
+  ns = nps+nsa-nint(rkpl(18,kcz+1),i8knd)
+  sd = huge_float

   ! print the header for the keff table by cycles skipped.
   write(iuo,70)
@@ -666,11 +668,11 @@
         o2(j,k) = o2(j,k)-rkpl(j,kcz+1-l)*rkpl(k,kcz+1-l)
       enddo
     enddo
-    ns = ns-nint(rkpl(18,kcz+1-l))
+    ns = ns-nint(rkpl(18,kcz+1-l),i8knd)
   enddo DO_190

   ! print the cycle number for the minimum combined keff deviation.
-  if( nc>0 )  write(iuo,200) kcz-nc,nc
+  if( nc>0 )  write(iuo,200) int(kcz,i8knd)-nc,nc
 200 format(2/," the minimum estimated standard deviation for the",&
     &   " col/abs/tl keff estimator occurs with",i4,  " inactive cycles",&
     &  " and",i5,  " active cycles.")
diff  -Naurd MCNP5/Source/src/crtcze.F90 MCNP5_new/Source/src/crtcze.F90
--- MCNP5/Source/src/crtcze.F90      2003-04-30 20:10:32.000000000 -0600
+++ MCNP5_new/Source/src/crtcze.F90  2004-07-22 15:14:41.000000000 -0600
@@ -8,6 +8,7 @@
   ! mm=1 to do generated weight windows.
   use mcnp_global
   use mcnp_debug
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

@@ -41,7 +42,7 @@
         i1 = ic
         i2 = ic
         w1 = 0.
-        w2 = huge
+        w2 = huge_float
         do js=abs(lca(ic)),abs(lca(ic+1))-1
           if( abs(lja(js)) > 1000000 )  cycle
           ij = 0
diff  -Naurd MCNP5/Source/src/dbmin.F90 MCNP5_new/Source/src/dbmin.F90
--- MCNP5/Source/src/dbmin.F90       2003-04-30 20:10:34.000000000 -0600
+++ MCNP5_new/Source/src/dbmin.F90   2004-07-22 15:14:41.000000000 -0600
@@ -12,7 +12,7 @@

   implicit real(dknd) (a-h,o-z)

-  dbmin = huge
+  dbmin = huge_float
   ic = icl
   x0 = xxx
   y0 = yyy
diff  -Naurd MCNP5/Source/src/dddiag.F90 MCNP5_new/Source/src/dddiag.F90
--- MCNP5/Source/src/dddiag.F90      2003-04-30 20:10:34.000000000 -0600
+++ MCNP5_new/Source/src/dddiag.F90  2004-07-22 15:14:41.000000000 -0600
@@ -36,7 +36,7 @@
     write(iuo,60) t4,int(ddn(j,id)),t1,ddn(j+9,id)*fpi,t2
   end do
   if( ddg(1,id)<0. ) then
-    write(iuo,60)huge,int(ddn(9,id)),1.,ddn(18,id)*fpi,1.
+    write(iuo,60)huge_float,int(ddn(9,id)),1.,ddn(18,id)*fpi,1.
 60  format(1pe19.5,i15,0pf20.5,1pe19.5,0pf15.5)
   else
    write(iuo,70)int(ddn(9,id)),1., ddn(18,id)*fpi,1.
@@ -47,7 +47,7 @@
   write(iuo,80) ddn(20,id)*fpi,ddn(21,id),t, int(ddn(23,id))
 80 format(/  " average tally per history =",1pe12.5,12x,"largest score =",e12.5/&
     &   " (largest score)/(average tally) =",e12.5,6x,&
-    &   "nps of largest score =",i9)
+    &   "nps of largest score =",i12)
   write(iuo,90)
 90 format(/  " score contributions by cell"/8x, "cell",4x, "misses",&
     & 6x, "hits",4x,  "tally per history",4x,  "weight per hit")
diff  -Naurd MCNP5/Source/src/den1.F90 MCNP5_new/Source/src/den1.F90
--- MCNP5/Source/src/den1.F90 2003-04-30 20:10:36.000000000 -0600
+++ MCNP5_new/Source/src/den1.F90     2004-07-22 15:14:41.000000000 -0600
@@ -12,6 +12,7 @@
   ! particles in various substances", phys rev b, 26, 6067(1982).
   use mcnp_global
   use mcnp_debug
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/den2.F90 MCNP5_new/Source/src/den2.F90
--- MCNP5/Source/src/den2.F90 2003-04-30 20:10:36.000000000 -0600
+++ MCNP5_new/Source/src/den2.F90     2004-07-22 15:14:41.000000000 -0600
@@ -8,6 +8,7 @@
   ! this routine solves eqs. (5) and (6) of sternheimer (1982).
   use mcnp_global
   use mcnp_debug
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/Depends MCNP5_new/Source/src/Depends
--- MCNP5/Source/src/Depends  2003-04-30 20:09:58.000000000 -0600
+++ MCNP5_new/Source/src/Depends      2004-07-22 15:14:40.000000000 -0600
@@ -9,10 +9,10 @@
 acedel$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 acefcn$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 acefpt$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) dynamic_arrays$(OBJF)
-acegam$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+acegam$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) dxtran_mod$(OBJF)
 acenu$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 acetbl$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-acetot$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+acetot$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 action$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 addtfc$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) ra2_mod$(OBJF)
 amatrx$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
@@ -21,9 +21,9 @@
 avrclc$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 avrnrm$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 avrout$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) rmc_mod$(OBJF)
-avrwgi$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+avrwgi$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 avrwwg$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-avrxyz$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+avrxyz$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 axis$(OBJF)   : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF)
 backup$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 bankit$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
@@ -32,18 +32,22 @@
 binval$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 brang$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 brem$(OBJF)   : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-brems$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+brems$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF) dxtran_mod$(OBJF)
 broadn$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 calcps$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 calcva$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF)
 celnbr$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) dynamic_arrays$(OBJF)
mcnp_plot$(OBJF)
-celpar$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF)
-celsrf$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF)
+celpar$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF) \
+              erprnt$(OBJF)
+celsrf$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF) \
+              erprnt$(OBJF)
 cgsdci$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-chekcs$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF)
+chekcs$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF) \
+              erprnt$(OBJF)
 chekit$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF) \
-              ra1_mod$(OBJF) fmesh_mod$(OBJF)
-chektr$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF)
+              ra1_mod$(OBJF) fmesh_mod$(OBJF)  erprnt$(OBJF)
+chektr$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF) \
+              erprnt$(OBJF) fmesh_mod$(OBJF)
 chkcel$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 chksrc$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 chqcel$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF)
@@ -51,48 +55,53 @@
 colidk$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) ra1_mod$(OBJF)
 colidn$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) ra1_mod$(OBJF)
 colidp$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-colinp$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF)
-collpn$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+colinp$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF) \
+              erprnt$(OBJF)
+collpn$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) dxtran_mod$(OBJF)
 confid$(OBJF) : mcnp_params$(OBJF) mcnp_debug$(OBJF)
 covar$(OBJF)  : mcnp_params$(OBJF) mcnp_debug$(OBJF)
 cprinp$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF)
 crit1_mod$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) crit2_mod$(OBJF) \
-              mcnp_input$(OBJF) ra1_mod$(OBJF) ra2_mod$(OBJF)
-crit2_mod$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) ra2_mod$(OBJF)
+              mcnp_input$(OBJF) ra1_mod$(OBJF) ra2_mod$(OBJF)  erprnt$(OBJF)
+crit2_mod$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) ra2_mod$(OBJF) \
+              erprnt$(OBJF)
 crspro$(OBJF) : mcnp_params$(OBJF) mcnp_debug$(OBJF)
-crtcze$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+crtcze$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 dbmin$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 dddet$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 dddiag$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 dddlev$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-den1$(OBJF)   : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-den2$(OBJF)   : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+den1$(OBJF)   : mcnp_global$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
+den2$(OBJF)   : mcnp_global$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 dmmp$(OBJF)   : $(LIBDOTCOMM) mcnp_params$(OBJF) mcnp_debug$(OBJF)
 dopplerp$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_random$(OBJF)
 dosef$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-dotrcl$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF)
+dotrcl$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF) \
+              erprnt$(OBJF)
 dunlev$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-dxdiag$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-dxtran$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+dxtran_mod$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 dynamic_arrays$(OBJF) : dmmp$(OBJF) mcnp_debug$(OBJF)
 echkcl$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-electr$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) fmesh_mod$(OBJF)
+electr$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) fmesh_mod$(OBJF) \
+              dxtran_mod$(OBJF)
 emaker$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 ephcom$(OBJF) : dmmp$(OBJF) mcnp_params$(OBJF)
-eqpbbn$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF)
+eqpbbn$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_input$(OBJF) \
+              erprnt$(OBJF)
 erf2$(OBJF)   : mcnp_params$(OBJF) mcnp_debug$(OBJF)
 ergimp$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 erprnt$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 errprn$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 escat$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 esloss$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) mcnp_landau$(OBJF)
```

```
-etsplt$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+etsplt$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 eventp$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) ra1_mod$(OBJF)
-exemes$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) dmmp$(OBJF)
+exemes$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) dmmp$(OBJF) \
+              erprnt$(OBJF)
 exmg$(OBJF)   : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 expire$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) dmmp$(OBJF) gxsub$(OBJF)
 expirx$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-expung$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+expung$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 extran$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 fastdr$(OBJF) : mcnp_params$(OBJF) mcnp_debug$(OBJF)
 ffetch$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
@@ -100,44 +109,56 @@
 findlv$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 finpht$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 fixcom$(OBJF) : dmmp$(OBJF) mcnp_params$(OBJF) racom$(OBJF)
-flaug$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+flaug$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF) dxtran_mod$(OBJF)
 fmesh_mod$(OBJF) : mcnp_global$(OBJF) messages$(OBJF) mcnp_params$(OBJF) \
-                 mcnp_iofiles$(OBJF) mcnp_data$(OBJF) mcnp_debug$(OBJF)
+              mcnp_iofiles$(OBJF) mcnp_data$(OBJF) mcnp_debug$(OBJF) \
+              erprnt$(OBJF)
 forcol$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 fshort$(OBJF) : mcnp_params$(OBJF) mcnp_debug$(OBJF)
 getexm$(OBJF) : mcnp_debug$(OBJF)
 getidt$(OBJF) : mcnp_params$(OBJF) mcnp_debug$(OBJF)
 getpar$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 getxs$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-getxst$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) dynamic_arrays$(OBJF)
+getxst$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) dynamic_arrays$(OBJF) \
+              erprnt$(OBJF)
 ginst$(OBJF)  : mcnp_debug$(OBJF)
 gmgww$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 gkssim$(OBJF) : mcnp_plot$(OBJF) mcnp_debug$(OBJF)
 gxsub$(OBJF)  : mcnp_global$(OBJF) mcnp_plot$(OBJF) gkssim$(OBJF) mcnp_debug$(OBJF)
 hpsort$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 hstory$(OBJF) : mcnp_global$(OBJF) smmp$(OBJF) ra1_mod$(OBJF) fmesh_mod$(OBJF) \
-              rmc_mod$(OBJF) mcnp_debug$(OBJF)
-igeom$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
+              rmc_mod$(OBJF) mcnp_debug$(OBJF) dxtran_mod$(OBJF)
+igeom$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) \
+              erprnt$(OBJF)
 imcn$(OBJF)   : mcnp_global$(OBJF) mcnp_input$(OBJF) crit1_mod$(OBJF) \
               ra1_mod$(OBJF) ra2_mod$(OBJF) \
               fmesh_mod$(OBJF) rmc_mod$(OBJF) mcnp_debug$(OBJF)
-inpert$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
+inpert$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) \
+              erprnt$(OBJF)
 inter$(OBJF)  : mcnp_global$(OBJF) mcnp_plot$(OBJF) mcnp_debug$(OBJF)
 intsec$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
-ipbc$(OBJF)   : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
+ipbc$(OBJF)   : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) \
+              erprnt$(OBJF)
 isheet$(OBJF) : mcnp_global$(OBJF) mcnp_plot$(OBJF) mcnp_debug$(OBJF)
 isos$(OBJF)   : mcnp_random$(OBJF) mcnp_params$(OBJF) mcnp_debug$(OBJF)
-isourc$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) crit1_mod$(OBJF) mcnp_debug$(OBJF)
-issrc$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
-itally$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) fmesh_mod$(OBJF)
+isourc$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) crit1_mod$(OBJF) mcnp_debug$(OBJF)
\
+              qttyin$(OBJF)
+issrc$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) \
+              erprnt$(OBJF)
+itally$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) \
+              fmesh_mod$(OBJF) erprnt$(OBJF)
 italpr$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
-items$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
-iwtwnd$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
-ixsdir$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
```

```
+items$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) \
+                erprnt$(OBJF)
+iwtwnd$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) \
+                erprnt$(OBJF)
+ixsdir$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) \
+                qttyin$(OBJF) erprnt$(OBJF)
 jbin$(OBJF)   : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 jdecod$(OBJF) : mcnp_global$(OBJF) mcplot_module$(OBJF) mcnp_plot$(OBJF) \
-                gxsub$(OBJF) mcnp_debug$(OBJF)
-jsourc$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
+                gxsub$(OBJF) mcnp_debug$(OBJF) qttyin$(OBJF)
+jsourc$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) \
+                erprnt$(OBJF)
 kdarg$(OBJF)  : mcnp_params$(OBJF) mcnp_debug$(OBJF)
 kdata$(OBJF)  : mcnp_debug$(OBJF)
 keypro$(OBJF) :
@@ -154,12 +175,15 @@
                 mcplot_module$(OBJF) crit1_mod$(OBJF) \
                 crit2_mod$(OBJF) ra1_mod$(OBJF) ra2_mod$(OBJF) gxsub$(OBJF) \
                 mcnp_debug$(OBJF)
-mapmaz$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-mbody$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
-mbodyo$(OBJF) : mcnp_params$(OBJF) mcnp_debug$(OBJF)
+mapmaz$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) qttyin$(OBJF) \
+                erprnt$(OBJF)
+mbody$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) \
+                erprnt$(OBJF)
+mbodyo$(OBJF) : mcnp_params$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF) \
+                erprnt$(OBJF)
 mbodyp$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
-mbodyr$(OBJF) : mcnp_params$(OBJF) mcnp_debug$(OBJF)
-mbodys$(OBJF) : mcnp_params$(OBJF) mcnp_debug$(OBJF)
+mbodyr$(OBJF) : mcnp_params$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF) erprnt$(OBJF)
+mbodys$(OBJF) : mcnp_params$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF) erprnt$(OBJF)
 mcnp_data$(OBJF)   : mcnp_params$(OBJF) mcnp_debug$(OBJF)
 mcnp_debug$(OBJF)  : mcnp_params$(OBJF)
 mcnp_global$(OBJF) : ephcom$(OBJF) fixcom$(OBJF) pblcom$(OBJF) tskcom$(OBJF)
varcom$(OBJF) \
@@ -176,37 +200,41 @@
 mctalw$(OBJF) : mcnp_global$(OBJF) ra1_mod$(OBJF) ra2_mod$(OBJF) mcnp_debug$(OBJF)
 messages$(OBJF) : dmmp$(OBJF) mcnp_debug$(OBJF)
 mgacol$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-mgcoln$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+mgcoln$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) dxtran_mod$(OBJF)
 mgcolp$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-mgimps$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-mgxspt$(OBJF) : mcnp_global$(OBJF) dynamic_arrays$(OBJF) mcnp_debug$(OBJF)
+mgimps$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
+mgxspt$(OBJF) : mcnp_global$(OBJF) dynamic_arrays$(OBJF) mcnp_debug$(OBJF) \
+                erprnt$(OBJF)
 midpnt$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
 movlat$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 msgcon$(OBJF) : mcnp_global$(OBJF) dmmp$(OBJF) messages$(OBJF) ra1_mod$(OBJF) \
-                racom$(OBJF) fmesh_mod$(OBJF) mcnp_debug$(OBJF)
+                racom$(OBJF) fmesh_mod$(OBJF) mcnp_debug$(OBJF) qttyin$(OBJF) \
+                erprnt$(OBJF)
 msgtsk$(OBJF) : mcnp_global$(OBJF) dmmp$(OBJF) messages$(OBJF) dynamic_arrays$(OBJF) \
                 smmp$(OBJF) ra1_mod$(OBJF) racom$(OBJF) fmesh_mod$(OBJF) mcnp_debug$(OBJF)
 namchg$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-namrsd$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+namrsd$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 newcd1$(OBJF) : mcnp_global$(OBJF) dynamic_arrays$(OBJF) mcnp_input$(OBJF) \
-                ra2_mod$(OBJF) fmesh_mod$(OBJF) mcnp_debug$(OBJF)
+                ra2_mod$(OBJF) fmesh_mod$(OBJF) mcnp_debug$(OBJF) qttyin$(OBJF) \
+                erprnt$(OBJF)
 newcel$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-newcrd$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) fmesh_mod$(OBJF) mcnp_debug$(OBJF)
+newcrd$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) fmesh_mod$(OBJF) mcnp_debug$(OBJF)
\
+                qttyin$(OBJF) erprnt$(OBJF)
 nextit$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) ra1_mod$(OBJF) ra2_mod$(OBJF) \
-              fmesh_mod$(OBJF) mcnp_debug$(OBJF)
-norma$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
-normh$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
+              fmesh_mod$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
+norma$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
+normh$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 nsf$(OBJF)    : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
-nsourc$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
+nsourc$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 nxtit1$(OBJF) : mcnp_global$(OBJF) dynamic_arrays$(OBJF) mcnp_input$(OBJF) \
               ra2_mod$(OBJF) fmesh_mod$(OBJF) mcnp_debug$(OBJF)
 nxtsym$(OBJF) : mcnp_debug$(OBJF)
 oldcd1$(OBJF) : mcnp_global$(OBJF) dynamic_arrays$(OBJF) mcnp_input$(OBJF) \
               fmesh_mod$(OBJF) mcnp_debug$(OBJF)
 oldcrd$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) ra1_mod$(OBJF) \
-              fmesh_mod$(OBJF) mcnp_debug$(OBJF)
+              fmesh_mod$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 output$(OBJF) : mcnp_global$(OBJF) mcplot_module$(OBJF) crit1_mod$(OBJF) \
-              ra1_mod$(OBJF) mcnp_debug$(OBJF)
+              ra1_mod$(OBJF) mcnp_debug$(OBJF) qttyin$(OBJF)
 pareto$(OBJF) : mcnp_params$(OBJF) mcnp_debug$(OBJF)
 pass1$(OBJF)  : mcnp_global$(OBJF) dynamic_arrays$(OBJF) mcnp_input$(OBJF)
mcnp_debug$(OBJF)
 pathmz$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
@@ -232,17 +260,19 @@
 pnctot$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 polhed$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
 prhpdf$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-prinv$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
+prinv$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) \
+              erprnt$(OBJF)
 prlost$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 prsdft$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 prsdst$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
 prssrj$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 prstat$(OBJF) : mcnp_global$(OBJF) ra2_mod$(OBJF) mcnp_debug$(OBJF)
-prtfcc$(OBJF) : mcnp_global$(OBJF) ra2_mod$(OBJF) mcnp_debug$(OBJF)
-psurf$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
-ptfc$(OBJF)   : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+prtfcc$(OBJF) : mcnp_global$(OBJF) ra2_mod$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
+psurf$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) \
+              erprnt$(OBJF)
+ptfc$(OBJF)   : mcnp_global$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 ptost$(OBJF)  : mcnp_global$(OBJF) mcnp_plot$(OBJF) mcnp_debug$(OBJF)
-ptrak$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+ptrak$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 pttyin$(OBJF) : mcnp_global$(OBJF) dmmp$(OBJF) mcnp_debug$(OBJF)
 putlbl$(OBJF) : mcnp_global$(OBJF) mcnp_plot$(OBJF) mcnp_debug$(OBJF)
 putnq$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
@@ -253,15 +283,17 @@
 quart$(OBJF)  : mcnp_params$(OBJF) mcnp_debug$(OBJF)
 ra1_mod$(OBJF): mcnp_global$(OBJF) dynamic_arrays$(OBJF) messages$(OBJF) \
               crit2_mod$(OBJF) ra2_mod$(OBJF) racom$(OBJF) mcnp_debug$(OBJF)
-ra2_mod$(OBJF): mcnp_global$(OBJF) mcnp_input$(OBJF) racom$(OBJF) mcnp_debug$(OBJF)
+ra2_mod$(OBJF): mcnp_global$(OBJF) mcnp_input$(OBJF) racom$(OBJF) mcnp_debug$(OBJF) \
+              erprnt$(OBJF)
 racom$(OBJF)  : mcnp_params$(OBJF) dmmp$(OBJF)
-rdprob$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
+rdprob$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) \
+              erprnt$(OBJF)
 reflec$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 refpbc$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-regula$(OBJF) : mcnp_global$(OBJF) mcnp_plot$(OBJF) mcnp_debug$(OBJF)
-rhoden$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
+regula$(OBJF) : mcnp_global$(OBJF) mcnp_plot$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
+rhoden$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 rmc_mod$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-ronge$(OBJF)  : mcnp_global$(OBJF) mcnp_landau$(OBJF) mcnp_debug$(OBJF)
+ronge$(OBJF)  : mcnp_global$(OBJF) mcnp_landau$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 rotas$(OBJF)  : mcnp_random$(OBJF) mcnp_params$(OBJF) mcnp_debug$(OBJF)
 rslmaz$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 runtpq$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
@@ -277,8 +309,9 @@
 setdas$(OBJF) : mcnp_global$(OBJF) messages$(OBJF) dynamic_arrays$(OBJF) \
               mcnp_input$(OBJF) mcnp_plot$(OBJF) \
               ra2_mod$(OBJF) fmesh_mod$(OBJF) mcnp_debug$(OBJF)
-sfiles$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
-shade$(OBJF)  : mcnp_global$(OBJF) mcnp_plot$(OBJF) gkssim$(OBJF) mcnp_debug$(OBJF)
+sfiles$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
+shade$(OBJF)  : mcnp_global$(OBJF) mcnp_plot$(OBJF) gkssim$(OBJF) mcnp_debug$(OBJF) \
+              erprnt$(OBJF)
 simint$(OBJF) : mcnp_params$(OBJF) mcnp_debug$(OBJF)
 simplx$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 sing$(OBJF)   : mcnp_global$(OBJF) mcnp_debug$(OBJF)
@@ -289,25 +322,26 @@
 sourck$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 spec$(OBJF)   : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_params$(OBJF)
mcnp_debug$(OBJF)
 spol$(OBJF)   : mcnp_params$(OBJF) mcnp_debug$(OBJF)
-sprob$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
+sprob$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 sqqint$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
 srcdx$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-srcsrf$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
-sread$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+srcsrf$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
+sread$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF) qttyin$(OBJF) erprnt$(OBJF)
 ssmsrc$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-startp$(OBJF) : mcnp_global$(OBJF) rmc_mod$(OBJF) mcnp_debug$(OBJF)
+startp$(OBJF) : mcnp_global$(OBJF) rmc_mod$(OBJF) mcnp_debug$(OBJF) dxtran_mod$(OBJF)
 sttop$(OBJF)  : mcnp_global$(OBJF) mcnp_plot$(OBJF) mcnp_debug$(OBJF)
-stuff$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) fmesh_mod$(OBJF)
+stuff$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) fmesh_mod$(OBJF)
\
+              qttyin$(OBJF) erprnt$(OBJF)
 sufwrt$(OBJF) : mcnp_global$(OBJF) smmp$(OBJF) ra2_mod$(OBJF) mcnp_debug$(OBJF)
 sumary$(OBJF) : mcnp_global$(OBJF) crit1_mod$(OBJF) ra2_mod$(OBJF) \
-              fmesh_mod$(OBJF) mcnp_debug$(OBJF)
+              fmesh_mod$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF) dxtran_mod$(OBJF)
 surfac$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 sursrc$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-tallmg$(OBJF) : mcnp_global$(OBJF) smmp$(OBJF) mcnp_debug$(OBJF)
-talloc$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
+tallmg$(OBJF) : mcnp_global$(OBJF) smmp$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
+talloc$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 tally$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 tallyd$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-tallyh$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+tallyh$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) qttyin$(OBJF)
 tallyp$(OBJF) : mcnp_global$(OBJF) ra2_mod$(OBJF) mcnp_debug$(OBJF)
 tallyq$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 tallyx$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
@@ -322,16 +356,16 @@
 torus$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 tpefil$(OBJF) : mcnp_global$(OBJF) dynamic_arrays$(OBJF) ra1_mod$(OBJF) \
               ra2_mod$(OBJF) mcnp_debug$(OBJF)
-track$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+track$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF) qttyin$(OBJF)
 transm$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-trfmat$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
-trfsrf$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
+trfmat$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
+trfsrf$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 trnspt$(OBJF) : mcnp_global$(OBJF) smmp$(OBJF) ra2_mod$(OBJF) mcnp_debug$(OBJF)
 tskcom$(OBJF) : fixcom$(OBJF) varcom$(OBJF) mcnp_params$(OBJF)
-ttbr$(OBJF)   : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+ttbr$(OBJF)   : mcnp_global$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF) dxtran_mod$(OBJF)
 ttyint$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-ufiles$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
-unimaz$(OBJF) : mcnp_global$(OBJF) dynamic_arrays$(OBJF) mcnp_debug$(OBJF)
+ufiles$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
+unimaz$(OBJF) : mcnp_global$(OBJF) dynamic_arrays$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
```

```
 unique$(OBJF) : mcnp_debug$(OBJF)
 uplev$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 uplpos$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
@@ -339,8 +373,9 @@
 utask$(OBJF)  : mcnp_global$(OBJF) smmp$(OBJF) ra1_mod$(OBJF) mcnp_debug$(OBJF)
 varcom$(OBJF) : dmmp$(OBJF) mcnp_params$(OBJF) racom$(OBJF)
 viewz$(OBJF)  : mcnp_global$(OBJF) mcnp_plot$(OBJF) mcnp_debug$(OBJF)
-voidcd$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
-volume$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
+voidcd$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
+volume$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF) qttyin$(OBJF) \
+               erprnt$(OBJF)
 vtask$(OBJF)  : mcnp_global$(OBJF) smmp$(OBJF) ra1_mod$(OBJF) \
                fmesh_mod$(OBJF) mcnp_debug$(OBJF)
 wgtul$(OBJF)  : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
@@ -348,12 +383,12 @@
 wtcalc$(OBJF) : mcnp_global$(OBJF) mcnp_input$(OBJF) mcnp_debug$(OBJF)
 wtmult$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 wtwndo$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-wwfile$(OBJF) : mcnp_global$(OBJF) rmc_mod$(OBJF) mcnp_debug$(OBJF)
+wwfile$(OBJF) : mcnp_global$(OBJF) rmc_mod$(OBJF) mcnp_debug$(OBJF) erprnt$(OBJF)
 wwval$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
 xact$(OBJF)   : mcnp_global$(OBJF) crit1_mod$(OBJF) mcnp_debug$(OBJF)
 xsec$(OBJF)   : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-xsgen$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF)
-ypbssp$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF)
+xsgen$(OBJF)  : mcnp_global$(OBJF) mcnp_debug$(OBJF) qttyin$(OBJF) erprnt$(OBJF)
+ypbssp$(OBJF) : mcnp_global$(OBJF) mcnp_debug$(OBJF) dxtran_mod$(OBJF)
 zaid$(OBJF)   : mcnp_debug$(OBJF)
 #
 # DOTCOMM objects
diff  -Naurd MCNP5/Source/src/dopplerp.F90 MCNP5_new/Source/src/dopplerp.F90
--- MCNP5/Source/src/dopplerp.F90     2003-04-30 20:10:38.000000000 -0600
+++ MCNP5_new/Source/src/dopplerp.F90 2004-07-22 15:14:41.000000000 -0600
@@ -25,14 +25,11 @@
   use mcnp_random, only : rang

   implicit real(dknd) (a-h,o-z)
-
-  real(dknd), parameter :: asc1 = 1.956934142   !1/rest mass electron
-  real(dknd), parameter :: asc2 = 137.03605     !fine struct. const.
  !----------------------------------------------------------------------

  ! For amu near unity
  if (amu > 0.999) then
-    esc = ene/(1+ene*asc1*(1-amu))
+    esc = ene/(one+ene/gpt(3)*(one-amu))
    return
  endif

@@ -68,28 +65,55 @@
    esgmax = ene-be
    if( esgmax<=zero ) cycle

-    ! Sample electron momentum pz from Compton profile cdf...
+    ! Calculate PZMAX
+    za = (be-ene*esgmax*(one-amu)/gpt(3))
+    zb = (two*ene*esgmax*(one-amu)+be**2)**0.5
+    pzmax = -fscon*za/zb
+
+    ! Find corresponding cdf for PZMAX
+      ! Index location of Compton Profile data
+      lp = jxs(10,iex)+nint(xss(jxs(9,iex)+ishell-1))-1   !beginning of CP data for
+                                                          !a given shell
+      it = nint(xss(lp))          !it = interpolation parameter
+      ne = nint(xss(lp+1))        !ne = number of momentum entries
+      jlp = lp+2                  !index of pz(1)
+      jup = lp+ne+1               !index of pz(ne)
+      jlc = lp+2*ne+2             !index of CDF(1)
+      juc = lp+3*ne+1             !index of CDF(ne)
+
+      ! binary search of momentum values, pz
+      do
+        if( jup-jlp<=1 )  exit
+        jmp = (jup+jlp)/2
+        if( pzmax<xss(jmp) ) then
+          jup = jmp
+        else
+          jlp = jmp
+        endif
+      enddo
+
+      cdfmax = xss(jlp+2*ne)      !Maximum value of CDF based on pzmax
+
+    ! Sample electron momentum pz from Compton profile cdf
+    ! Limits sampling of pzrn to  pzrn <= pzmax
+    rn = rang()*cdfmax         !Normalization to maximum value of CDF
+    if( rn > cdfmax) cycle
+
     ! binary search of Compton profile cdf
-    rn = rang()
-    lp = jxs(10,iex)+nint(xss(jxs(9,iex)+ishell-1))-1
-    it = nint(xss(lp))          !it = interpolation parameter
-    ne = nint(xss(lp+1))        !ne = number of momentum entries
-    jl = lp+2+2*ne              !index of CDF(1)
-    ju = lp+1+3*ne              !index of CDF(ne)
     do
-      if( ju-jl<=1 )  exit
-      jm = (ju+jl)/2
-      if( rn<xss(jm) ) then
-        ju = jm
+      if( juc-jlc<=1 )  exit
+      jmc = (juc+jlc)/2
+      if( rn<xss(jmc) ) then
+        juc = jmc
       else
-        jl = jm
+        jlc = jmc
       endif
     enddo
-    cdf1 = xss(jl)
-    pdf1 = xss(jl-ne)      !(lp+1+ne) + (jl-(lp+1+2ne))
-    p1   = xss(jl-2*ne)    !(lp+1) +  (jl-(lp+1+2ne))
-    pdf2 = xss(jl+1-ne)
-    p2   = xss(jl+1-2*ne)
+    cdf1 = xss(jlc)
+    pdf1 = xss(jlc-ne)      !(lp+1+ne) + (jlc-(lp+1+2ne))
+    p1   = xss(jlc-2*ne)    !(lp+1) +  (jlc-(lp+1+2ne))
+    pdf2 = xss(jlc+1-ne)
+    p2   = xss(jlc+1-2*ne)

     ! linear interpolation scheme
     if( p1 == p2 ) then
@@ -98,30 +122,26 @@
       pzrn = p1 + (rn - cdf1) / pdf1
     else
       bb = (pdf2 - pdf1) / (p2 - p1)
-      pzrn = p1 + (sqrt(max(zero,pdf1**2+2*bb*(rn-cdf1)))-pdf1)/bb
+      pzrn = p1 + (sqrt(max(zero,pdf1**2+two*bb*(rn-cdf1)))-pdf1)/bb
     endif

-    ! Calculate PZMAX; Reject if sampled PZRN is out of range
-    za = (be-ene*esgmax*(1-amu)*asc1)
-    zb = (2*ene*esgmax*(1-amu)+be**2)**0.5
-    pzmax = -asc2*za/zb
     if( pzrn<zero .or. pzmax<zero ) then
-      esc = ene/(1.+ene*asc1*(1.-amu))
+      esc = ene/(one+ene/gpt(3)*(one-amu))
       exit
     endif
     if( pzrn > pzmax ) cycle
     ! Calculate ESC from sampled pzrn
-    xa = (pzrn/asc2)**2-1.-((ene*(1.-amu)*asc1)**2)-2.*ene*(1.-amu)*asc1
-    xb = 2*ene+2*(ene**2)*(1.-amu)*asc1-2.*(pzrn/asc2)**2*ene*amu
-    xc = ((pzrn*ene/asc2)**2)-ene**2
-    xrad = xb**2-4.*xa*xc
+    xa = (pzrn/fscon)**2-one-((ene*(one-amu)/gpt(3))**2)-two*ene*(one-amu)/gpt(3)
+    xb = two*ene+two*(ene**2)*(one-amu)/gpt(3)-two*(pzrn/fscon)**2*ene*amu
+    xc = ((pzrn*ene/fscon)**2)-ene**2
+    xrad = xb**2-four*xa*xc
     if( xrad<zero ) cycle
     rn = rang()
-    if( rn<0.5 ) then
-      esc = .5*(-xb+sqrt(xrad))/xa
+    if( rn<half ) then
+      esc = half*(-xb+sqrt(xrad))/xa
     else
-      esc = .5*(-xb-sqrt(xrad))/xa
+      esc = half*(-xb-sqrt(xrad))/xa
     end if
     if( esc>0 .and. esc <= esgmax ) exit
   enddo
diff  -Naurd MCNP5/Source/src/dotrcl.F90 MCNP5_new/Source/src/dotrcl.F90
--- MCNP5/Source/src/dotrcl.F90      2003-04-30 20:10:40.000000000 -0600
+++ MCNP5_new/Source/src/dotrcl.F90  2004-07-22 15:14:41.000000000 -0600
@@ -7,6 +7,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/dxdiag.F90 MCNP5_new/Source/src/dxdiag.F90
--- MCNP5/Source/src/dxdiag.F90      2003-04-30 20:10:40.000000000 -0600
+++ MCNP5_new/Source/src/dxdiag.F90  2004-07-22 15:14:41.000000000 -0600
@@ -13,7 +13,7 @@
     if( ndx(ip)==0 )  cycle DO_140

     write(iuo,10) hnp(ip)(1:len_trim(hnp(ip)))//'s',nps
-10  format(  "1dxtran diagnostics -- ",a9,7x, "nps =",i9,51x,"print table 150")
+10  format(  "1dxtran diagnostics -- ",a9,7x, "nps =",i12,48x,"print table 150")

     ! print a table for each dxtran sphere.
     DO_130: do n=1,ndx(ip)
@@ -38,7 +38,7 @@
         if( ddx(ip,1,n)<0.  )  t4=-talb(j,2)*ddx(ip,1,n)
         write(iuo,50)t4,int(dxd(ip,j,n)),t1,fpi*dxd(ip,j+9,n),t2
       end do
-      if( ddx(ip,1,n)<0. )  write(iuo,50) huge,int(dxd(ip,9,n)),1.,fpi*dxd(ip,18,n),1.
+      if( ddx(ip,1,n)<0. )  write(iuo,50)
huge_float,int(dxd(ip,9,n)),1.,fpi*dxd(ip,18,n),1.
 50    format(1pe19.4,i15,0pf20.5,1pe20.5,0pf15.5)
       if( ddx(ip,1,n)>=0. )  write(iuo,60) int(dxd(ip,9,n)),1.,fpi*dxd(ip,18,n),1.
 60    format(  "  1st 200 histories",i15,0pf20.5,1pe20.5,0pf15.5)
@@ -46,7 +46,7 @@
       if( dxd(ip,20,n)/=0. )  t = dxd(ip,21,n)/(dxd(ip,20,n)*fpi)
       write(iuo,70) dxd(ip,20,n)*fpi,dxd(ip,21,n),t,int(dxd(ip,23,n))
 70    format(/  " average weight per history =",1pe12.5,7x,"largest weight =",e12.5/&
-        &   " largest/average =",e12.5,18x,  "nps of largest =",i9)
+        &   " largest/average =",e12.5,18x,  "nps of largest =",i12)
       write(iuo,80)
 80    format(2/," contributions by cell",2/,8x, "cell",4x, "misses",&
         & 6x, "hits",4x,  "weight per history",4x,  "weight per hit"/)
diff  -Naurd MCNP5/Source/src/dxtran.F90 MCNP5_new/Source/src/dxtran.F90
--- MCNP5/Source/src/dxtran.F90      2003-04-30 20:10:40.000000000 -0600
+++ MCNP5_new/Source/src/dxtran.F90  1969-12-31 17:00:00.000000000 -0700
@@ -1,299 +0,0 @@
-!+ $Id$
-! Copyright LANL/UC/DOE - see file COPYRIGHT_INFO
-
-subroutine dxtran
```

```
-  ! create particles on the dxtran spheres.
-  use mcnp_global
-  use mcnp_debug
-  implicit real(dknd) (a-h,o-z)
-
-  ! return if kcode problem is not settled.  initialize.
-  if( kc8>0 )  return
-  tw = wgt
-  if( ipsc==10 )  tw = tw*cbwf
-  te = erg
-
-  ! adjust the weight according to the cell probability.
-  if( dxcp(0,ipt,icl)==0. ) go to 110
-  if( dxcp(0,ipt,icl)<1. ) then
-    if( rang()>dxcp(0,ipt,icl) ) go to 110
-    tw = tw/dxcp(0,ipt,icl)
-  endif
-
-  ! pre-processing for special cases.
-  select case( ipsc )
-
-    case( 4 )
-      !  >>>>>  ipsc=4 -- neutron from collision with moving target.
-      tw = cmult*tw
-      if( ntyn<0 ) then
-        f = awn(iex)/ycn
-        tpd(7) = f*ssr
-        tpd(1) = tpd(7)
-        if( ntyn/=-99 ) tpd(1) = (1.+awn(iex))*sqrt(ergace/eg0)
-        tpd(2) = eg0/(1.+awn(iex))**2
-        tpd(4) = uold(1)+f*vtr(1)
-        tpd(5) = uold(2)+f*vtr(2)
-        tpd(6) = uold(3)+f*vtr(3)
-      else
-        g = sqrt(2.*ttn/awn(iex))
-        tpd(7) = g*ssr
-        tpd(1) = sqrt(2.*ergace)
-        tpd(2) = .5d+0
-        tpd(4) = g*vtr(1)
-        tpd(5) = g*vtr(2)
-        tpd(6) = g*vtr(3)
-      endif
-      tpd(3) = tpd(4)**2+tpd(5)**2+tpd(6)**2
-      if( tpd(1)**2<tpd(3) ) then
-        tw = tw*2.
-        if( rang()>.5d+0 )  tpd(3) = -tpd(3)
-      endif
-      go to 130
-
-    case( 5, 14:16, 104 )
-      !  >>>>>  ipsc=5 -- neutron from collision with stationary target and
-      ! ipsc=14 -- neutron from kalbach-87 (law 44) collision and
-      ! ipsc=15 -- neutron from law 67 (endf/b-vi law 7) collision and
-      ! ipsc=16 -- neutron from law 61 (tabulated energy / angle) and
-      ! ipsc=104 -- particle from isotropic multigroup collision.
-      tw = cmult*tw
-      go to 130
-
-    case( 103 )
-      !  >>>>>  ipsc=103 -- photon from double fluorescence.
-      if( rang()>.5d+0 )  te = xss(jxs(4,iex)+3*nxs(4,iex)+1)
-      tw = 2.*tw
-      if( te>elc(2) ) go to 130
-      go to 110
-
-    case( 105 )
-      !  >>>>>  ipsc=105 -- photon from pair production.
-      tw = 2.*tw
-      go to 130
-
-    case( 3, 6:13, 17:18, 101:102, 106 )
-      go to 130
-
-    case default
-      call expirx(1,'dxtran','illegal value for ipsc.')
-      return
-
-  end select
-
-110 continue
-  ! increment nziytc for particles that failed.
-  l = icl+mxa*(ndx(1)*(ipt-1)-1)
-  do ix = 1,ndx(ipt)
-    if( idx/=ix ) then
-      nziytc(1,ix,ipt) = nziytc(1,ix,ipt)+1
-      dxc(kdxc+3,l+mxa*ix) = dxc(kdxc+3,l+mxa*ix)+1.
-    endif
-  enddo
-  return
-
-130 continue
-  ! do all of the spheres for the current kind of particle.
-  call savpar(0,2)
-  nb = npb
-  DO_370: do ix = 1,ndx(ipt)
-    if( idx==ix )  cycle DO_370
-
-    ld = icl9(npb)+mxa*(ix-1+(ipt-1)*ndx(1))
-    dxc(kdxc+3,ld) = dxc(kdxc+3,ld)+1.
-    wgt = tw
-    if( dxcp(ix,ipt,icl)==0. ) go to 280
-    if( dxcp(ix,ipt,icl)/=1. ) then
-      if( rang()>dxcp(ix,ipt,icl) ) go to 280
-      wgt = wgt/dxcp(ix,ipt,icl)
-    endif
-    idx = ix
-
-    ! calculate the distance to the center of the sphere.
-    if( lev==0 ) then
-      f = dxx(ipt,1,ix)-xxx
-      g = dxx(ipt,2,ix)-yyy
-      h = dxx(ipt,3,ix)-zzz
-    else
-      f = dxx(ipt,1,ix)-udt(1,0)
-      g = dxx(ipt,2,ix)-udt(2,0)
-      h = dxx(ipt,3,ix)-udt(3,0)
-    endif
-    ds = f**2+g**2+h**2
-    rd = 1./sqrt(ds)
-
-    ! sample the direction to the point on the dxtran sphere.
-    pd = 1.
-    erg = te
-    psc = .5d+0
-    if( ipsc/=11 ) then
-      if( dxx(ipt,5,ix)<=1.d-5*ds ) then
-        d4 = dxx(ipt,4,ix)
-        d5 = dxx(ipt,5,ix)
-        if( d4<=1.d-10*ds )  d4 = d5
-        if( rang()*(4.*d4+d5)>=d5-d4 ) then
-          cf = (ds-rang()*.5d+0*d4)/ds
-          pd = .1d+0*(4.*d4+d5)/ds
-        else
-          cf = (ds-.5d+0*(d5-rang()*(d5-d4)))/ds
-          pd = .5d+0*(4.*d4+d5)/ds
-        endif
-      else
-        ci = sqrt(ds-dxx(ipt,4,ix))*rd
-        co = sqrt(ds-dxx(ipt,5,ix))*rd
-        pd = 5.-4.*ci-co
-        if( rang()*pd>=ci-co ) then
-          cf = (ci+rang()*(1.-ci))
-          pd = .2d+0*pd
-        else
-          cf = (co+rang()*(ci-co))
-        endif
-      endif
-      tpp(1) = f*rd
-      tpp(2) = g*rd
-      tpp(3) = h*rd
-      call rotas(cf,tpp,uvw,0,irt)
-      if( lev/=0 ) call dddlev
-
-      ! calculate the scattering probability and energy.
-      if( ipsc<=100 ) then
-        call calcps
-        if( kdb/=0  ) go to 380
-        if( psc==0. ) go to 290
-        if( erg<elc(ipt) ) go to 310
-        if( ipt/=2  ) vel = slite*sqrt(erg*(erg+2.*gpt(ipt)))/(erg+gpt(ipt))
-      endif
-    endif
-
-    ! calculate the transmission to the new location.
-    if( lev==0 ) then
-      q = f*uuu+g*vvv+h*www
-    else
-      q = f*udt(4,0)+g*udt(5,0)+h*udt(6,0)
-    endif
-    if( q<0.  ) go to 290
-    dl = dxx(ipt,5,ix)-ds+q**2
-    if( dl<0. ) go to 290
-    dd = q-sqrt(dl)
-    dd = dd-dd*1d-10
-    if( dd>=vel*(tco(ipt)-tme)) go to 300
-    s = 0.
-    if( ddx(ipt,1,ix)<0. )  then
-      s = -ddx(ipt,1,ix)/(wgt*psc*pd)
-    elseif( ddx(ipt,1,ix)>0. .and. npstc>200 ) then
-      s = ddx(ipt,1,ix)*dxd(ipt,24,ix)/(wgt*psc*pd)
-    endif
-    call transm(dd,s,iz)
-    if( iz/=0 )  go to 295
-
-    ! set up and bank the new particle.
-    jsu = 0
-    wgt = wgt*psc*pd*exp(-amfp)
-    if( iets(ipt)>0 ) then
-      call ergimp(0)
-    endif
-    do i = 1,mipt
-      elc(i) = elp(i,icl)
-      fiml(i) = fim(i,icl)
-      do l = 0,lev-1
-        elc(i)  = max( elc(i), elp(i,int(udt(7,l))) )
-        fiml(i) = fiml(i)*fim(i,int(udt(7,l)))
-      enddo
-    enddo
-    sf = wtfasv
-    if( dbcn(20)/=0. )  sf = 1.
-    if( wgt*fiml(1)*sf<=fiml9(1,1)*dxw(ipt,2) ) then
-      if( wgt*fiml(1)*sf<fiml9(1,1)*dxw(ipt,1)*rang() )  go to 320
-      wgt = fiml9(1,1)*dxw(ipt,1)/(fiml(1)*sf)
-    endif
-    if( erg<elc(ipt) )  go to 310
-    npa = 1
-    iex = iex9(npb)
-    paxtc(1,8,ipt) = paxtc(1,8,ipt)+1.
-    paxtc(2,8,ipt) = paxtc(2,8,ipt)+wgt
-    paxtc(3,8,ipt) = paxtc(3,8,ipt)+wgt*erg
-    pwb(kpwb+ipt,10,icl) = pwb(kpwb+ipt,10,icl)+wgt
-    call bankit(1)
-
```

```
-    ! accumulate data for dxtran diagnostics.
-    dxd(kdxd+ipt,20,ix) = dxd(kdxd+ipt,20,ix)+wgt
-    j = 0
-    if( ddx(ipt,1,ix)<0. ) then
-      if( wgt>=-ddx(ipt,1,ix)*ddx(ipt,2,ix) )  j = 1
-      do i = 1,8
-        if( wgt<=-talb(i,2)*ddx(ipt,1,ix) ) exit
-      enddo
-    else
-      i = 9
-      if( npstc>200 ) then
-        if( wgt>=ddx(ipt,2,ix)*dxd(ipt,24,ix) )  j = 1
-        i = 1
-        if( wgt>ddx(ipt,1,ix)*dxd(ipt,24,ix) ) then
-          do i = 2,7
-            if( wgt<=talb(i,1)*dxd(ipt,24,ix) ) exit
-          enddo
-        endif
-      endif
-    endif
-    dxd(kdxd+ipt,i,ix)   = dxd(kdxd+ipt,i,ix)+1.
-    dxd(kdxd+ipt,i+9,ix) = dxd(kdxd+ipt,i+9,ix)+wgt
-    if( wgt>dxd(kdxd+ipt,21,ix) )  dxd(kdxd+ipt,23,ix) = npstc
-    dxd(kdxd+ipt,21,ix)  = max(dxd(kdxd+ipt,21,ix),wgt)
-    dxc(kdxc+1,ld) = dxc(kdxc+1,ld)+1.
-    dxc(kdxc+2,ld) = dxc(kdxc+2,ld)+wgt
-
-    ! print a line for especially heavy created particles.
-    if( j==0 ) go to 330
-
-    !$OMP CRITICAL (PRINT_OUTPUT)
-    if( dxd(ipt,22,ix)<100. ) then
-      dxd(ipt,22,ix) = dxd(ipt,22,ix)+1.
-      if( mct>=-1 ) then
-        write(iuo,260) hnp(ipt),idx,wgt,wgt9(npb),psc,amfp,&
-          & dd,erg,ncl(icl9(npb)),npstc,ncp
-260     format(/ " idx",4x, "dx wgt",4x, "col wgt",5x, "psc",7x, "amfp",8x,&
-          &  "dd",9x, "erg",5x, "cell",8x, "nps  ncp",13x,a1,&
-          & /i3,1p,6e11.4,i6,i11,i5)
-      endif
-    endif
-    !$OMP END CRITICAL (PRINT_OUTPUT)
-
-    go to 330
-
-    ! increment nziytc for particles that failed.
-280 continue
-    nziytc(1,ix,ipt) = nziytc(1,ix,ipt)+1
-    go to 330
-290 continue
-    nziytc(2,ix,ipt) = nziytc(2,ix,ipt)+1
-    go to 330
-295 continue
-    if( kdb/=0 )  go to 380
-    nziytc(iz,ix,ipt) = nziytc(iz,ix,ipt)+1
-    if( iz/=5  )  go to 330
-    j = nziy(5,ix,ipt)+nziytc(5,ix,ipt)-1
-    call errprn(1,j,2,zero+ix,zero+ncl(icl),'dxt','cel',&
-      & 'contribution to dxtran sphere prevented by zero importance.')
-    go to 330
-300 continue
-    nziytc(6,ix,ipt) = nziytc(6,ix,ipt)+1
-    go to 330
-310 continue
-    nziytc(7,ix,ipt) = nziytc(7,ix,ipt)+1
-    go to 330
-320 continue
-    nziytc(8,ix,ipt) = nziytc(8,ix,ipt)+1
-
-330 continue
-    ! restore the particle description.
-    call getpar(nb,2)
-  enddo DO_370
-  npb = nb-1
-  return
-
-380 continue
-  ! return with kdb=5 or 6 if the particle gets lost.
-  kdb = kdb+4
-  npb = nb-1
-  return
-end subroutine dxtran
diff  -Naurd MCNP5/Source/src/dxtran_mod.F90 MCNP5_new/Source/src/dxtran_mod.F90
--- MCNP5/Source/src/dxtran_mod.F90   1969-12-31 17:00:00.000000000 -0700
+++ MCNP5_new/Source/src/dxtran_mod.F90      2004-07-22 15:14:41.000000000 -0600
@@ -0,0 +1,483 @@
+!+ $Id: dxtran_mod.F90,v 1.1 2004/04/07 16:19:47 jgoorley Exp $
+! Copyright LANL/UC/DOE - see file COPYRIGHT_INFO
+
+module dxtran_mod
+  !-------------------------------------------------------------------
+  ! This module contains:
+  !    (1) dxtran - perform dxtran calculations & tallies
+  !    (2) dxdiag - print dxtran diagnostics, called from sumary
+  !    (3) inside_dxtran_sphere - returns the number of the dxtran
+  !                 sphere containing particle, 0 if not inside sphere
+  !    (4) dist_dxtran_sphere   - returns distance to nearest dxtran
+  !                 sphere
+  !-------------------------------------------------------------------
+  use mcnp_global
+  use mcnp_debug
+  !save
+
+  ! varcom:
+  !  integer   :: nziy(   8, mxdx, mipt )       != DXTRANs lost to zero importance
+  !  real(dknd):: dxcp( 0:mxdx*min(1,nxnx), mipt, mxa )  != DXTRAN cell probabilities
+  !  real(dknd):: dxc( 3, mxa*nxnx*mt)          != DXTRAN contributions by cell
+  !  real(dknd):: dxd( mipt, 24, mxdx*(mt+1) ) != DXTRAN diagnostics
+  ! fixcom:
+  !  real(dknd):: ddx( mipt, 2, mxdx )  != controls for DXTRAN diagnostics
+  !  real(dknd):: dxx( mipt, 5, mxdx )  != DXTRAN sphere parameters
+  !  real(dknd):: dxw( mipt, 3 )        != DXTRAN weight cutoffs
+  !  integer   :: ndx(mipt)             != Number of neutron & photon DXTRAN spheres
+  !  integer   :: nxnx                  != Number of DXTRAN spheres in problem
+  ! tskcom:
+  !  integer   :: nziytc( 8, mxdx, mipt ) != Task copy of nziy
+  !common               /dxtran_com/     nziytc
+  !xxOMP THREADPRIVATE( /dxtran_com/ )
+
+CONTAINS
+
+
+!--------------------------------------------------------------------
+subroutine dxtran
+  ! create particles on the dxtran spheres - called during histories
+  implicit none
+  real(dknd) :: ci, co, cf, d4, d5, dd, dl, ds
+  real(dknd) ::  f,  g,  h, pd,  q, rd, rg,  s, sf, te, tw
+  integer    ::  i, ix, iz,  j,  l, ld, nb
+
+  ! return if kcode problem is not settled.  initialize.
+  if( kc8>0 )  return
+  tw = wgt
+  if( ipsc==10 )  tw = tw*cbwf
+  te = erg
+
+  ! adjust the weight according to the cell probability.
+  if( dxcp(0,ipt,icl)==zero ) go to 110
+  if( dxcp(0,ipt,icl)<one ) then
+    if( rang()>dxcp(0,ipt,icl) ) go to 110
+    tw = tw/dxcp(0,ipt,icl)
+  endif
```

```
+
+  ! pre-processing for special cases.
+  select case( ipsc )
+
+    case( 4 )
+      !  >>>>>  ipsc=4 -- neutron from collision with moving target.
+      tw = cmult*tw
+      if( ntyn<0 ) then
+        f = awn(iex)/ycn
+        tpd(7) = f*ssr
+        tpd(1) = tpd(7)
+        if( ntyn/=-99 ) tpd(1) = (one+awn(iex))*sqrt(ergace/eg0)
+        tpd(2) = eg0/(one+awn(iex))**2
+        tpd(4:6) = uold(1:3)+f*vtr(1:3)
+      else
+        g = sqrt(two*ttn/awn(iex))
+        tpd(7) = g*ssr
+        tpd(1) = sqrt(two*ergace)
+        tpd(2) = half
+        tpd(4:6) = g*vtr(1:3)
+      endif
+      tpd(3) = tpd(4)**2+tpd(5)**2+tpd(6)**2
+      if( tpd(1)**2<tpd(3) ) then
+        tw = tw*two
+        if( rang()>half )  tpd(3) = -tpd(3)
+      endif
+      go to 130
+
+    case( 5, 14:16, 104 )
+      !  >>>>>  ipsc=5 -- neutron from collision with stationary target and
+      ! ipsc=14 -- neutron from kalbach-87 (law 44) collision and
+      ! ipsc=15 -- neutron from law 67 (endf/b-vi law 7) collision and
+      ! ipsc=16 -- neutron from law 61 (tabulated energy / angle) and
+      ! ipsc=104 -- particle from isotropic multigroup collision.
+      tw = cmult*tw
+      go to 130
+
+    case( 103 )
+      !  >>>>>  ipsc=103 -- photon from double fluorescence.
+      if( rang()>half )  te = xss(jxs(4,iex)+3*nxs(4,iex)+1)
+      tw = two*tw
+      if( te>elc(2) ) go to 130
+      go to 110
+
+    case( 105 )
+      !  >>>>>  ipsc=105 -- photon from pair production.
+      tw = two*tw
+      go to 130
+
+    case( 3, 6:13, 17:18, 101:102, 106 )
+      go to 130
+
+    case default
+      call expirx(1,'dxtran','illegal value for ipsc.')
+      return
+
+  end select
+
+110 continue
+  ! increment nziytc for particles that failed.
+  l = icl+mxa*(ndx(1)*(ipt-1)-1)
+  do ix = 1,ndx(ipt)
+    if( idx/=ix ) then
+      nziytc(1,ix,ipt) = nziytc(1,ix,ipt)+1
+      dxc(kdxc+3,l+mxa*ix) = dxc(kdxc+3,l+mxa*ix)+one
+    endif
+  enddo
+  return
+
+130 continue
+  ! do all of the spheres for the current kind of particle.
+  call savpar(0,2)
+  nb = npb
+  DO_370: do ix = 1,ndx(ipt)
+    if( idx==ix )  cycle DO_370
+
+    ld = icl9(npb)+mxa*(ix-1+(ipt-1)*ndx(1))
+    dxc(kdxc+3,ld) = dxc(kdxc+3,ld)+one
+    wgt = tw
+    if( dxcp(ix,ipt,icl)==zero ) go to 280
+    if( dxcp(ix,ipt,icl)/=one ) then
+      if( rang()>dxcp(ix,ipt,icl) ) go to 280
+      wgt = wgt/dxcp(ix,ipt,icl)
+    endif
+    idx = ix
+
+    ! calculate the distance to the center of the sphere.
+    if( lev==0 ) then
+      f = dxx(ipt,1,ix)-xxx
+      g = dxx(ipt,2,ix)-yyy
+      h = dxx(ipt,3,ix)-zzz
+    else
+      f = dxx(ipt,1,ix)-udt(1,0)
+      g = dxx(ipt,2,ix)-udt(2,0)
+      h = dxx(ipt,3,ix)-udt(3,0)
+    endif
+    ds = f**2+g**2+h**2
+    rd = one/sqrt(ds)
+
+    ! sample the direction to the point on the dxtran sphere.
+    pd = one
+    erg = te
+    psc = half
+    if( ipsc/=11 ) then
+      if( dxx(ipt,5,ix)<=1.e-5_dknd*ds ) then
+        d4 = dxx(ipt,4,ix)
+        d5 = dxx(ipt,5,ix)
+        if( d4<=1.e-10_dknd*ds )  d4 = d5
+        if( rang()*(four*d4+d5)>=d5-d4 ) then
+          cf = (ds-rang()*half*d4)/ds
+          pd = tenth*(four*d4+d5)/ds
+        else
+          cf = (ds-half*(d5-rang()*(d5-d4)))/ds
+          pd = half*(four*d4+d5)/ds
+        endif
+      else
+        ci = sqrt(ds-dxx(ipt,4,ix))*rd
+        co = sqrt(ds-dxx(ipt,5,ix))*rd
+        pd = five-four*ci-co
+        if( rang()*pd>=ci-co ) then
+          cf = (ci+rang()*(one-ci))
+          pd = .2_dknd*pd
+        else
+          cf = (co+rang()*(ci-co))
+        endif
+      endif
+      tpp(1) = f*rd
+      tpp(2) = g*rd
+      tpp(3) = h*rd
+      call rotas(cf,tpp,uvw,0,irt)
+      if( lev/=0 ) call dddlev
+
+      ! calculate the scattering probability and energy.
+      if( ipsc<=100 ) then
+        call calcps
+        if( kdb/=0  ) go to 380
+        if( psc==zero ) go to 290
+        if( erg<elc(ipt) ) go to 310
+        if( gpt(ipt)/=zero )  then
+          vel = slite*sqrt(erg*(erg+two*gpt(ipt)))/(erg+gpt(ipt))
+        else
+          vel = slite
+        endif
+      endif
+    endif
+
+    ! calculate the transmission to the new location.
+    if( lev==0 ) then
+      q = f*uuu+g*vvv+h*www
+    else
+      q = f*udt(4,0)+g*udt(5,0)+h*udt(6,0)
+    endif
+    if( q<zero  ) go to 290
+    dl = dxx(ipt,5,ix)-ds+q**2
+    if( dl<zero ) go to 290
+    dd = q-sqrt(dl)
+    dd = dd-dd*1.0e-10_dknd
+    if( dd>=vel*(tco(ipt)-tme)) go to 300
+    s = zero
+    if( ddx(ipt,1,ix)<zero )  then
+      s = -ddx(ipt,1,ix)/(wgt*psc*pd)
+    elseif( ddx(ipt,1,ix)>zero .and. npstc>200_i8knd ) then
+      s = ddx(ipt,1,ix)*dxd(ipt,24,ix)/(wgt*psc*pd)
+    endif
+    call transm(dd,s,iz)
+    if( iz/=0 )  go to 295
+
+    ! set up and bank the new particle.
+    jsu = 0
+    wgt = wgt*psc*pd*exp(-amfp)
+    if( iets(ipt)>0 ) then
+      call ergimp(0)
+    endif
+    do i = 1,mipt
+      elc(i) = elp(i,icl)
+      fiml(i) = fim(i,icl)
+      do l = 0,lev-1
+        elc(i)  = max( elc(i), elp(i,int(udt(7,l))) )
+        fiml(i) = fiml(i)*fim(i,int(udt(7,l)))
+      enddo
+    enddo
+    sf = wtfasv
+    if( wgt*fiml(1)*sf<=fiml9(1,1)*dxw(ipt,2) ) then
+      if( wgt*fiml(1)*sf<fiml9(1,1)*dxw(ipt,1)*rang() )  go to 320
+      wgt = fiml9(1,1)*dxw(ipt,1)/(fiml(1)*sf)
+    endif
+    if( erg<elc(ipt) )  go to 310
+    npa = 1
+    iex = iex9(npb)
+    paxtc(1,8,ipt) = paxtc(1,8,ipt)+1.
+    paxtc(2,8,ipt) = paxtc(2,8,ipt)+wgt
+    paxtc(3,8,ipt) = paxtc(3,8,ipt)+wgt*erg
+    pwb(kpwb+ipt,10,icl) = pwb(kpwb+ipt,10,icl)+wgt
+    call bankit(1)
+
+    ! accumulate data for dxtran diagnostics.
+    dxd(kdxd+ipt,20,ix) = dxd(kdxd+ipt,20,ix)+wgt
+    j = 0
+    if( ddx(ipt,1,ix)<zero ) then
+      if( wgt>=-ddx(ipt,1,ix)*ddx(ipt,2,ix) )  j = 1
+      do i = 1,8
+        if( wgt<=-talb(i,2)*ddx(ipt,1,ix) ) exit
+      enddo
+    else
+      i = 9
+      if( npstc>200_i8knd ) then
+        if( wgt>=ddx(ipt,2,ix)*dxd(ipt,24,ix) )  j = 1
+        i = 1
+        if( wgt>ddx(ipt,1,ix)*dxd(ipt,24,ix) ) then
+          do i = 2,7
+            if( wgt<=talb(i,1)*dxd(ipt,24,ix) ) exit
+          enddo
+        endif
+      endif
+    endif
+    dxd(kdxd+ipt,i,ix)   = dxd(kdxd+ipt,i,ix)+one
+    dxd(kdxd+ipt,i+9,ix) = dxd(kdxd+ipt,i+9,ix)+wgt
+    if( wgt>dxd(kdxd+ipt,21,ix) )  dxd(kdxd+ipt,23,ix) = npstc
+    dxd(kdxd+ipt,21,ix)  = max(dxd(kdxd+ipt,21,ix),wgt)
+    dxc(kdxc+1,ld) = dxc(kdxc+1,ld)+one
+    dxc(kdxc+2,ld) = dxc(kdxc+2,ld)+wgt
+
+    ! print a line for especially heavy created particles.
+    if( j==0 ) go to 330
+
+    !$OMP CRITICAL (PRINT_OUTPUT)
+    if( dxd(ipt,22,ix)<hundred ) then
+      dxd(ipt,22,ix) = dxd(ipt,22,ix)+one
+      if( mct>=-1 ) then
+        write(iuo,260) hnp(ipt),idx,wgt,wgt9(npb),psc,amfp,&
+          & dd,erg,ncl(icl9(npb)),npstc,ncp
+260     format(/ " idx",4x, "dx wgt",4x, "col wgt",5x, "psc",7x, "amfp",8x,&
+          &  "dd",9x, "erg",5x, "cell",8x, " nps  ncp",13x,a1,&
+          & /i3,6es11.4,i6,i12,i5)
+      endif
+    endif
+    !$OMP END CRITICAL (PRINT_OUTPUT)
+
+    go to 330
+
+    ! increment nziytc for particles that failed.
+280 continue
+    nziytc(1,ix,ipt) = nziytc(1,ix,ipt)+1
+    go to 330
+290 continue
+    nziytc(2,ix,ipt) = nziytc(2,ix,ipt)+1
+    go to 330
+295 continue
+    if( kdb/=0 )  go to 380
+    nziytc(iz,ix,ipt) = nziytc(iz,ix,ipt)+1
+    if( iz/=5  )  go to 330
+    j = nziy(5,ix,ipt)+nziytc(5,ix,ipt)-1
+    call errprn(1,j,2,real(ix,dknd),real(ncl(icl),dknd),'dxt','cel',&
+      & 'contribution to dxtran sphere prevented by zero importance.')
+    go to 330
+300 continue
+    nziytc(6,ix,ipt) = nziytc(6,ix,ipt)+1
+    go to 330
+310 continue
+    nziytc(7,ix,ipt) = nziytc(7,ix,ipt)+1
+    go to 330
+320 continue
+    nziytc(8,ix,ipt) = nziytc(8,ix,ipt)+1
+
+330 continue
+    ! restore the particle description.
+    call getpar(nb,2)
+  enddo DO_370
+  npb = nb-1
+  return
+
+380 continue
+  ! return with kdb=5 or 6 if the particle gets lost.
+  kdb = kdb+4
+  npb = nb-1
+  return
+end subroutine dxtran
+
+!-------------------------------------------------------------------
+
+subroutine dxdiag
+  ! print dxtran diagnostics - called by main thread in master
+  implicit none
+  real(dknd) ::  t, t1, t2, t3, t4
+  integer    ::  i, ip,  j,  l,  m,  n
+
+  ! print separate tables for neutron and photon spheres.
+  DO_140: do ip=1,mipt
+    if( ndx(ip)==0 )  cycle DO_140
+
+    write(iuo,10) hnp(ip)(1:len_trim(hnp(ip)))//'s',nps
+10  format(  "1dxtran diagnostics -- ",a9,7x, "nps =",i12,48x,"print table 150")
+
+    ! print a table for each dxtran sphere.
+    DO_130: do n=1,ndx(ip)
+      if( ddx(ip,1,n)<zero  ) then
+        write(iuo,20) n,(dxx(ip,i,n),i=1,3),'      maximum weight'
+      else
+        write(iuo,20) n,(dxx(ip,i,n),i=1,3),'times average weight'
+20      format(2/," dxtran sphere no.",i2,  " located at x,y,z =",3es12.5/&
+          & 45x,  "cumulative",8x, "weight",11x,  "cumulative"/&
+          & 45x,  "fraction of",7x,  "transmitted",6x,  "fraction of"/&
+          & 2x,a20,5x,  "transmissions",5x,  "transmissions",5x,&
+          &   "per history",6x,  "total weight"/)
+      endif
+      t1 = zero
+      t2 = zero
+      t3 = sum( dxd(ip,1:9,n) )
+      do j=1,8
+        if( t3/=zero )  t1=t1+dxd(ip,j,n)/t3
+        if( dxd(ip,20,n)/=zero )  t2=t2+dxd(ip,j+9,n)/dxd(ip,20,n)
+        if( ddx(ip,1,n)>=zero )  then
+          if( j==1 ) then
+            t4 = ddx(ip,1,n)
+          else
+            t4 = talb(j,1)
+          endif
+        else
+          t4 = -talb(j,2)*ddx(ip,1,n)
+        endif
+        write(iuo,50) t4,int(dxd(ip,j,n)),t1,fpi*dxd(ip,j+9,n),t2
+      end do
+      if( ddx(ip,1,n)<zero ) then
+        write(iuo,50) huge_float,int(dxd(ip,9,n)),one,fpi*dxd(ip,18,n),one
+50      format(es19.4,i15,f20.5,es20.5,f15.5)
+      endif
+      if( ddx(ip,1,n)>=zero ) then
+        write(iuo,60) int(dxd(ip,9,n)),one,fpi*dxd(ip,18,n),one
+60      format(  "  1st 200 histories",i15,f20.5,es20.5,f15.5)
+      endif
+      t = one
+      if( dxd(ip,20,n)/=zero )  t = dxd(ip,21,n)/(dxd(ip,20,n)*fpi)
+      write(iuo,70) dxd(ip,20,n)*fpi,dxd(ip,21,n),t,int(dxd(ip,23,n))
+70    format(/  " average weight per history =",es12.5,7x,"largest weight =",es12.5/&
+        &   " largest/average =",es12.5,18x,  "nps of largest =",i12)
+      write(iuo,80)
+80    format(2/," contributions by cell",2/,8x, "cell",4x, "misses",&
+        & 6x, "hits",4x,  "weight per history",4x,  "weight per hit"/)
+      j = 0
+      m = 0
+      t = zero
+      l = mxa*((ip-1)*ndx(1)+n-1)
+      do i=1,mxa
+        if( dxc(3,l+i)==zero )  cycle
+        j = j+dxc(3,l+i)
+        m = m+dxc(1,l+i)
+        t = t+dxc(2,l+i)
+        write(iuo,90) i,ncl(i),int(dxc(3,l+i)-dxc(1,l+i)),&
+          & int(dxc(1,l+i)),dxc(2,l+i)*fpi,dxc(2,l+i)/max(dxc(1,l+i),one)
+90      format(2i6,2i10,es18.5,es20.5)
+      end do
+      if( m/=0 ) then
+        write(iuo,110) j-m,m,t*fpi,t/m
+110     format(7x, "total",2i10,es18.5,es20.5)
+      endif
+      write(iuo,120) (nziy(i,n,ip),i=1,8)
+120   format(2/," misses"/  "   russian roulette on dxc",i19/&
+        &  "   psc=0.",i36/  "   russian roulette in transmission",i10/&
+        &  "   underflow in transmission",i17/&
+        &  "   hit a zero-importance cell",i16/  "   time cutoff",i31/&
+        &  "   energy cutoff",i29/  "   weight cutoff",i29)
+    end do DO_130
+  end do DO_140
+  return
+end subroutine dxdiag
+
+!--------------------------------------------------------------------
+
+function inside_dxtran_sphere()
+  ! see if particle is inside a dxtran sphere.
+  ! if so, return the sphere number, otherwise return 0
+  implicit none
+  integer :: inside_dxtran_sphere
+  integer :: j
+
+  inside_dxtran_sphere = 0
+  if( lev==0 ) then
+    do j=1,ndx(ipt)
+      if( sum((xyz(1:3)-dxx(ipt,1:3,j))**2) <  dxx(ipt,5,j) ) then
+        inside_dxtran_sphere = j
+        exit
+      endif
+    enddo
+  else
+    do j=1,ndx(ipt)
+      if( sum((udt(1:3,0)-dxx(ipt,1:3,j))**2) <  dxx(ipt,5,j) ) then
+        inside_dxtran_sphere = j
+        exit
+      endif
+    enddo
+  endif
+  return
+end function inside_dxtran_sphere
+
+!--------------------------------------------------------------------
+
+function dist_dxtran_sphere()
+  ! Calculate the distance to the nearest dxtran sphere, dxl.
+  implicit none
+  real(dknd) :: dist_dxtran_sphere
+  real(dknd) :: sr(3), sd(3), q, c
+  integer :: i
+
+  dist_dxtran_sphere = huge_float
+
+  do i = 1,ndx(ipt)
+    if( idx/=i ) then
+      if( lev==0 )  then
+        sr = dxx(ipt,1:3,i) - xyz
+        sd = uvw
+      else
+        sr = dxx(ipt,1:3,i) - udt(1:3,0)
+        sd = udt(4:6,0)
+      endif
+      q = sum( sr*sd )
+      c = min(max(zero,q),dist_dxtran_sphere)
+      if( sum( (sr-sd*c)**2 ) < dxx(ipt,5,i) ) then
+        dist_dxtran_sphere = min( dist_dxtran_sphere, &
+          &  q-sqrt(max(zero,q**2+dxx(ipt,5,i)-sum(sr**2))) )
+      endif
+    endif
+  enddo
+  return
+end function dist_dxtran_sphere
+
+!--------------------------------------------------------------------
```

```
+end module dxtran_mod
diff  -Naurd MCNP5/Source/src/electr.F90 MCNP5_new/Source/src/electr.F90
--- MCNP5/Source/src/electr.F90      2003-04-30 20:10:42.000000000 -0600
+++ MCNP5_new/Source/src/electr.F90  2004-07-22 15:14:41.000000000 -0600
@@ -5,6 +5,7 @@
   ! run one electron track.

   use mcnp_global
+  use dxtran_mod
   use mcnp_debug
   use fmesh_mod, only:  mesh_score, nmesh

@@ -47,12 +48,12 @@
       ! calculate the distances to time cutoff and energy substep.
       dtc = vel*(tco(3)-tme)
       n1  = ngp
-      pmf = huge
+      pmf = huge_float
       if( mkc/=0 )  pmf=drs(nq+nee*(mkc-1))/rho(icl)

       ! calculate the distance to the cell boundary, dls.
       if( mbd(icl)==0 .and. jsu==0 ) then
-        dls = huge
+        dls = huge_float
         rr  = rr-pmf
         if( rr>0. )  go to 70
         rr  = dbmin()-pmf
@@ -98,7 +99,7 @@
       pac(kpac+3,5,icl) = pac(kpac+3,5,icl)+wgt*dt*erg
       pac(kpac+3,6,icl) = pac(kpac+3,6,icl)+wgt*d*erg
       pac(kpac+3,7,icl) = pac(kpac+3,7,icl)+d
-      if( pmf=huge ) then
+      if( pmf/=huge_float ) then
         pac(kpac+3,8,icl) = pac(kpac+3,8,icl)+wgt*d*pmf
       endif
       pac(kpac+3,9,icl)  = pac(kpac+3,9,icl)+wgt*dt
@@ -241,11 +242,7 @@
   jsu = 0
   npa = 1
   ncp = 0
-  idx = 0
-  do ii=1,ndx(2)
-    if( (xxx-dxx(2,1,ii))**2+(yyy-dxx(2,2,ii))**2+&
-      & (zzz-dxx(2,3,ii))**2 < dxx(2,5,ii) )  idx=ii
-  end do
+  idx = inside_dxtran_sphere()
   ipsc = 102
   vel = slite
   call isos(uvw,lev)
diff  -Naurd MCNP5/Source/src/ephcom.F90 MCNP5_new/Source/src/ephcom.F90
--- MCNP5/Source/src/ephcom.F90      2003-04-30 20:10:44.000000000 -0600
+++ MCNP5_new/Source/src/ephcom.F90  2004-07-22 15:14:41.000000000 -0600
@@ -19,38 +19,27 @@
   public
   save

-  ! Ephcom lengths and equivalences
+  !-------------------------------------------------------------------------------------
--
+  !  (1) parameters for the length of the 3 portions of /ephcm/
+
   integer, parameter :: nephcm =& != Length of real part of /EPHCM/.
     & 12*(1) + 1*(4) + 1*(11)

   integer, parameter :: lephcm =& != Length of integer part of /EPHCM/.
-    & 49*(1) + 2*(2) + 1*(3) + 1*(novr) + 1*(nptr)
+    & 46*(1) + 2*(2) + 1*(3) + 1*(novr)

-  real(dknd) :: gephcm(nephcm) != Equivalence to real part of /EPHCM/.
-  integer    :: jephcm(lephcm) != Equivalence to integer part of /EPHCM/.
-  equivalence  (cp1,gephcm),  (ics,jephcm)
```

```
+  integer, parameter :: l8ephcm = &  != Length of integer*8 part of /EPHCM/.
+    &  3*1 + 1*(nptr)

-  ! reals.
-  common /ephcm/ &
-    &  cp1,   cp3,   ctme,  fpi,   &
-    &  freq,  ssb,   tdc,   tlc,   trm,   &
-    &  wnvp,  xhom,  xnum,  yhom,  zephcm
+  !-------------------------------------------------------------------------------
--
+  !  (2) declarations for 3 arrays equivalenced to /ephcm/

-  ! integers.
-  common /ephcm/ &
-    & ics,    idmp,   ifile,  iln,    iln1,   &
-    & inform, iovr,   inpd,   iptr,   iptra,  &
-    & irup,   itask,  iterm,  itfxs,  itotnu, &
-    & iuou,   jchar,  jfcn,   jgf,    jgxa,   &
-    & jgxo,   jovr,   jtasks, jtfc,   jvp,    &
-    & kbp,    kdbnps, kmplot, komout, konrun, &
-    & kprod,  krflg,  krtm,   ksr,    ktfile, &
-    & lchnk,  lfatl,  locki,  lspeed, ltasks, &
-    & mazf,   mcolor, mdc,    mmkdb,  mnk,    &
-    & mpc,    mrm,    nde,    nkrp,   nomore, &
-    & nrc,    nst,    ntasks, mephcm, lockl,  &
-    & mynum
+  real(dknd) :: gephcm(nephcm) != Equivalence to real part of /EPHCM/.
+  integer(i8knd) :: i8ephcm(l8ephcm) != Equivalence to integer*8 part of /EPHCM/.
+  integer    :: jephcm(lephcm) != Equivalence to integer part of /EPHCM/.

+  !-------------------------------------------------------------------------------
--
+  !  (3) declarations for /ephcm/ REALS
   real(dknd) :: cp1                  != Computer time used after beginning MCRUN.
   real(dknd) :: cp3                  != Computer time of multiprocessing subtasks.
   real(dknd) :: ctme                 != User requested ctm endtime.
@@ -66,6 +55,14 @@
   real(dknd) :: yhom                 != Vertical coordinate of home position.
   real(dknd) :: zephcm               != Marker after floating-point part of /EPHCM/.

+  !-------------------------------------------------------------------------------
--
+  !  (4) declarations for /ephcm/ INTEGER*8
+  integer(i8knd) :: kdbnps           != NPS of bad-trouble history in multitasking.
+  integer(i8knd) :: inpd             != TFC rendezvous frequency (5th PRDMP entry).
+  integer(i8knd) :: nde              != Value of execute-message item DBUG n.
+  integer(i8knd) :: iptra(nptr)             != Pointer to PTR() for each PTRAC keyword.
+  !-------------------------------------------------------------------------------
--
+  !  (5) declarations for /ephcm/ INTEGERS
   integer :: ics                     != Flag for error on current input card.
   integer :: idmp                    != Number of the dump to start a continue run from.
   integer :: ifile                   != I/O unit of current plot input file.
@@ -73,9 +70,7 @@
   integer :: iln1                    != Saved count of lines of input data.
   integer :: inform                  != Flag for output to plot user.
   integer :: iovr                    != Index of the current code section.
-  integer :: inpd                    != TFC rendezvous frequency (5th PRDMP entry).
   integer :: iptr                    != PTRAC option flag.
-  integer :: iptra(nptr)             != Pointer to PTR() for each PTRAC keyword.
   integer :: irup                    != Flag set by user with ctrl-c interrupt.
   integer :: itask                   != Number of active tasks.
   integer :: iterm                   != Type of plotting display.
@@ -92,7 +87,6 @@
   integer :: jtfc                    != Flag to indicate TFC update is due.
   integer :: jvp                     != Flag for square viewport.
   integer :: kbp                     != Interrupt flag for multitasking mode.
-  integer :: kdbnps                  != NPS of bad-trouble history in multitasking.
   integer :: kmplot                  != Indicator of < ctrl-e > MCPLOT interrupt.
   integer :: komout                  != Indicator that COMOUT has been created.
   integer :: konrun                  != Continue-run flag.
```

```
@@ -113,7 +107,6 @@
   integer :: mnk                     != Flag to indicate maximum printing is wanted.
   integer :: mpc                     != Flag indicating that printing is due to be done.
   integer :: mrm                     != Flag indicating that plotting is due to be done.
-  integer :: nde                     != Value of execute-message item DBUG n.
   integer :: nkrp                    != Latch for warning in CALCPS.
   integer :: nomore                  != Flag for exhausted surface-source file.
   integer :: nrc                     != Count of restarts in the run.
@@ -124,21 +117,37 @@

   logical :: lockl                   ! = Logical lock variable.

-contains

-  subroutine eph_cast(mh,mx,ie)
-    ! Description:
-    ! DMMP bcast of ephcom data.
-    ! Arguments:
-    integer,intent(in)    :: mh ! action flag, 0 -> sender.
-    integer,intent(in)    :: mx ! message chunk size (max).
-    integer,intent(inout) :: ie ! return status.
+  !------------------------------------------------------------------------------------
--
+  !  (6) declaration for /ephcm/   (real, integer*8, integer)

-    call dm_bcast(mh,gephcm,nephcm,mx,ie)
-    call dm_bcast(mh,jephcm,lephcm,mx,ie)
+  common /ephcm/ &
+    &  cp1,   cp3,   ctme,  fpi,   &
+    &  freq,  ssb,   tdc,   tlc,   trm,   &
+    &  wnvp,  xhom,  xnum,  yhom,  zephcm

-    return
-  end subroutine eph_cast
+  common /ephcm/ &
+    & kdbnps, inpd,  nde, iptra
+
+
+  common /ephcm/ &
+    & ics,    idmp,   ifile,  iln,    iln1,   &
+    & inform, iovr,           iptr,           &
+    & irup,   itask,  iterm,  itfxs,  itotnu, &
+    & iuou,   jchar,  jfcn,   jgf,    jgxa,   &
+    & jgxo,   jovr,   jtasks, jtfc,   jvp,    &
+    & kbp,            kmplot, komout, konrun, &
+    & kprod,  krflg,  krtm,   ksr,    ktfile, &
+    & lchnk,  lfatl,  locki,  lspeed, ltasks, &
+    & mazf,   mcolor, mdc,    mmkdb,  mnk,    &
+    & mpc,    mrm,            nkrp,   nomore, &
+    & nrc,    nst,    ntasks, mephcm, lockl,  &
+    & mynum

+  equivalence  (cp1,     gephcm)
+  equivalence  (kdbnps, i8ephcm)
+  equivalence  (ics,     jephcm)
+
+ ! -------------------------------------------------------------------------
 end module ephcom
 !-
diff  -Naurd MCNP5/Source/src/eqpbbn.F90 MCNP5_new/Source/src/eqpbbn.F90
--- MCNP5/Source/src/eqpbbn.F90      2003-04-30 20:10:44.000000000 -0600
+++ MCNP5_new/Source/src/eqpbbn.F90   2004-07-22 15:14:41.000000000 -0600
@@ -13,6 +13,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)
   external spec
diff  -Naurd MCNP5/Source/src/ergimp.F90 MCNP5_new/Source/src/ergimp.F90
```

```
--- MCNP5/Source/src/ergimp.F90      2003-04-30 20:10:46.000000000 -0600
+++ MCNP5_new/Source/src/ergimp.F90  2004-07-22 15:14:41.000000000 -0600
@@ -19,7 +19,7 @@
   sf = wtfasv
   wtfasv = 1.
   e0 = eg0
-  if( i0==0 )  e0=huge
+  if( i0==0 )  e0=huge_float
   ! find esplt/tsplt importance ratio bins for increasing values.
   tf = erg
   do ib=1,2
diff  -Naurd MCNP5/Source/src/erprnt.F90 MCNP5_new/Source/src/erprnt.F90
--- MCNP5/Source/src/erprnt.F90      2003-04-30 20:10:46.000000000 -0600
+++ MCNP5_new/Source/src/erprnt.F90  2004-07-22 15:14:41.000000000 -0600
@@ -1,89 +1,360 @@
-!+ $Id$
+!+ $Id: erprnt.F90,v 1.3 2004/05/19 21:09:38 fbrown Exp $
 ! Copyright LANL/UC/DOE - see file COPYRIGHT_INFO

-subroutine erprnt(im,jf,nk,k1,k2,k3,k4,js,hf)
+module erprnt_mod
   ! Description:
   ! Print  messages about errors in the problem specifications.
   ! Notice:
   ! Terminal messages on some systems are limited to 70 characters.
   ! So fatal errors should have only 55 characters, warnings 59.
+  ! Four versions are supplied: erprnt, erprnt_i8_k1, erprnt_i8_k2, and
+  ! erprnt_i8_k1k2.  The specializations allow passing of i8knd integers.

-  ! Modules used:
-  use mcnp_global
-  use mcnp_debug
-  implicit none
+  contains

-  ! Argument declarations:
-  integer :: im,jf,nk,k1,k2,k3,k4,js
-  !  im=1  print message only.    im=2  klin also, to terminal.
-  !  jf=1  fatal error.           jf=2  warning only.   jf=3=comment only
-  !  nk = how many of the parameters k1 thru k4 are to be printed.
-  !  js > 0 set ics = -1;  js < 0 print to jtty only.
-  character(len=*)   :: hf ! partial printing format.
+  subroutine erprnt(im,jf,nk,k1,k2,k3,k4,js,hf)
+    ! erprnt subroutine when all variables are the same type of integers.
+    ! Note that on some platforms they will all be I4, on other
+    ! platforms (usually compiled with -I8R8) all arguments will be I8.

-  ! Local declarations:
-  integer :: kk(4), j
-  character(len=100) :: hp
-  character(len=20),dimension(3) :: hs = &
-    & (/  '(" fatal error.  ", ', &
-    &     '(" warning. ",      ', &
-    &     '(" comment. ",      ' /)
+    ! Modules used:
+    use mcnp_global
+    use mcnp_debug

+    implicit none

-  ! Move the parameters to be printed into the printing block.
-  kk = (/k1,k2,k3,k4/)
+    ! Argument declarations:
+    integer :: im,jf,nk,k1,k2,k3,k4,js
+    !  im=1  print message only.    im=2  klin also, to terminal.
+    !  jf=1  fatal error.           jf=2  warning only.   jf=3  comment only.
+    !  nk = how many of the parameters k1 thru k4 are to be printed.
+    !  js > 0 set ics = -1;  js < 0 print to jtty only.
+    character(len=*)   :: hf ! partial printing format.

-  ! Assemble the format from the leaders and the partial format, hf.
-  j = jf
-  if( jf<1 .or. jf>3 ) j = 1
-  hp = hs(j) // hf // ')'
+    ! Local declarations:
+    integer :: kk(4)
+    integer :: j
+    character(len=100) :: hp
+    character(len=20),dimension(3) :: hs = &
+      & (/  '(" fatal error.  ", ', &
+      &     '(" warning. ",      ', &
+      &     '(" comment. ",      ' /)

-  ! Print the error message to the output file.
-  if( js>=0 ) then
-    if( im==1 )  write(iuo,'(" ")')
-    if( iln1<0 .and. im==2 )  write(iuo,'(1x,a69)')  klin(1:69)
-    if( nk==0 )  then
-      write(iuo,hp)
+
+    ! Move the parameters to be printed into the printing block.
+    kk = (/k1,k2,k3,k4/)
+
+    ! Assemble the format from the leaders and the partial format, hf.
+    j = jf
+    if( jf<1 .or. jf>3 ) j = 1
+    hp = hs(j) // hf // ')'
+
+    ! Print the error message to the output file.
+    if( js>=0 ) then
+      if( im==1 )  write(iuo,'(" ")')
+      if( iln1<0 .and. im==2 )  write(iuo,'(1x,a69)')  klin(1:69)
+      if( nk==0 )  then
+        write(iuo,hp)
+      else
+        write(iuo,hp)  kk(1:min(4,nk))
+      endif
+    endif
+
+    ! Also print, if required, to the terminal.
+    ! In production mode, messages are not sent to the terminal.
+    if( (kprod==0 .and. nwer+nfer<20) .or. (nfer==0 .and. jf==1) ) then
+      if( iovr==1 .and. im/=2 .and. iln1/=0 )  write(jtty,'(" ")')
+      if( im/=2 )  iln1 = 0
+      if( iln/=iln1 .and. im==2 )  write(jtty,'(1x,a69)')  klin(1:69)
+      if( im==2 )  iln1 = iln
+      if( nk==0 )  then
+        write(jtty,hp)
+      else
+        write(jtty,hp)  kk(1:min(4,nk))
+      endif
+    endif
+
+
+    ! Limit the number of error messages to the controller to 20.
+    if( kprod==0 .and. nfer+nwer==20 )  write(jtty,20)  outp
+20  format(" additional error messages on file ",a8)
+
+    ! Increment count of fatal errors or warnings.
+    if(     jf==3 )  then
+      ! don't count the "comments"
+      continue
+    elseif( jf==2 )  then
+      ! Warning only.
+      if( npp>=0 )  nwer = nwer+1
     else
-      write(iuo,hp)  kk(1:min(4,nk))
+      ! Fatal condition.
+      nfer = nfer+1
+      if( mct>=0 )  then
+        ink(1:mink) = 1
+        mnk = 1
+      endif
     endif
-  endif
+    if( js>0 )  ics = -1

-  ! Also print, if required, to the terminal.
-  ! In production mode, messages are not sent to the terminal.
-  if( (kprod==0 .and. nwer+nfer<20) .or. (nfer==0 .and. jf==1) ) then
-    if( iovr==1 .and. im/=2 .and. iln1/=0 )  write(jtty,'(" ")')
-    if( im/=2 )  iln1 = 0
-    if( iln/=iln1 .and. im==2 )  write(jtty,'(1x,a69)')  klin(1:69)
-    if( im==2 )  iln1 = iln
-    if( nk==0 )  then
-      write(jtty,hp)
+    return
+  end subroutine erprnt
+
+  subroutine erprnt_i8_k1(im,jf,nk,k1,k2,k3,k4,js,hf)
+    ! erprnt subroutine for explicit 8 byte integer argument k1.
+
+    ! Modules used:
+    use mcnp_global
+    use mcnp_debug
+
+    implicit none
+
+    ! Argument declarations:
+    integer        :: im,jf,nk,k2,k3,k4,js
+    integer(i8knd) :: k1
+    !  im=1  print message only.    im=2  klin also, to terminal.
+    !  jf=1  fatal error.           jf=2  warning only.   jf=3  comment only.
+    !  nk = how many of the parameters k1 thru k4 are to be printed.
+    !  js > 0 set ics = -1;  js < 0 print to jtty only.
+    character(len=*)   :: hf ! partial printing format.
+
+    ! Local declarations:
+    integer(i8knd) :: kk(4)
+    integer        :: j
+    character(len=100) :: hp
+    character(len=20),dimension(3) :: hs = &
+      & (/  '(" fatal error.  ", ', &
+      &     '(" warning. ",      ', &
+      &     '(" comment. ",      ' /)
+
+
+    ! Move the parameters to be printed into the printing block.
+    kk = (/k1,int(k2,i8knd),int(k3,i8knd),int(k4,i8knd)/)
+
+    ! Assemble the format from the leaders and the partial format, hf.
+    j = jf
+    if( jf<1 .or. jf>3 ) j = 1
+    hp = hs(j) // hf // ')'
+
+    ! Print the error message to the output file.
+    if( js>=0 ) then
+      if( im==1 )  write(iuo,'(" ")')
+      if( iln1<0 .and. im==2 )  write(iuo,'(1x,a69)')  klin(1:69)
+      if( nk==0 )  then
+        write(iuo,hp)
+      else
+        write(iuo,hp)  kk(1:min(4,nk))
+      endif
+    endif
+
+    ! Also print, if required, to the terminal.
+    ! In production mode, messages are not sent to the terminal.
+    if( (kprod==0 .and. nwer+nfer<20) .or. (nfer==0 .and. jf==1) ) then
+      if( iovr==1 .and. im/=2 .and. iln1/=0 )  write(jtty,'(" ")')
+      if( im/=2 )  iln1 = 0
+      if( iln/=iln1 .and. im==2 )  write(jtty,'(1x,a69)')  klin(1:69)
+      if( im==2 )  iln1 = iln
+      if( nk==0 )  then
+        write(jtty,hp)
+      else
+        write(jtty,hp)  kk(1:min(4,nk))
+      endif
+    endif
+
+
+    ! Limit the number of error messages to the controller to 20.
+    if( kprod==0 .and. nfer+nwer==20 )  write(jtty,20)  outp
+20  format(" additional error messages on file ",a8)
+
+    ! Increment count of fatal errors or warnings.
+    if(     jf==3 )  then
+      ! don't count the "comments"
+      continue
+    elseif( jf==2 )  then
+      ! Warning only.
+      if( npp>=0 )  nwer = nwer+1
     else
-      write(jtty,hp)  kk(1:min(4,nk))
+      ! Fatal condition.
+      nfer = nfer+1
+      if( mct>=0 )  then
+        ink(1:mink) = 1
+        mnk = 1
+      endif
     endif
-  endif
+    if( js>0 )  ics = -1

+    return
+  end subroutine erprnt_i8_k1

-  ! Limit the number of error messages to the controller to 20.
-  if( kprod==0 .and. nfer+nwer==20 )  write(jtty,20)  outp
-20 format(" additional error messages on file ",a8)
+  subroutine erprnt_i8_k2(im,jf,nk,k1,k2,k3,k4,js,hf)
+    ! erprnt subroutine for explicit 8 byte integer argument k2.

-  ! Increment count of fatal errors or warnings.
-  if(     jf==3 )  then
-    ! don't count the "comments"
-    continue
-  elseif( jf==2 )  then
-    ! Warning only.
-    if( npp>=0 )  nwer = nwer+1
-  else
-    ! Fatal condition.
-    nfer = nfer+1
-    if( mct>=0 )  then
-      ink(1:mink) = 1
-      mnk = 1
+    ! Modules used:
+    use mcnp_global
+    use mcnp_debug
+
+    implicit none
+
+    ! Argument declarations:
+    integer        :: im,jf,nk,k1,k3,k4,js
+    integer(i8knd) :: k2
+    !  im=1  print message only.    im=2  klin also, to terminal.
+    !  jf=1  fatal error.           jf=2  warning only.   jf=3  comment only.
+    !  nk = how many of the parameters k1 thru k4 are to be printed.
+    !  js > 0 set ics = -1;  js < 0 print to jtty only.
+    character(len=*)   :: hf ! partial printing format.
+
+    ! Local declarations:
+    integer(i8knd) :: kk(4)
+    integer        :: j
+    character(len=100) :: hp
+    character(len=20),dimension(3) :: hs = &
+      & (/  '(" fatal error.  ", ', &
+      &     '(" warning. ",      ', &
+      &     '(" comment. ",      ' /)
+
+
+    ! Move the parameters to be printed into the printing block.
+    kk = (/int(k1,i8knd),k2,int(k3,i8knd),int(k4,i8knd)/)
+
+    ! Assemble the format from the leaders and the partial format, hf.
+    j = jf
+    if( jf<1 .or. jf>3 ) j = 1
+    hp = hs(j) // hf // ')'
+
+    ! Print the error message to the output file.
+    if( js>=0 ) then
+      if( im==1 )  write(iuo,'(" ")')
+      if( iln1<0 .and. im==2 )  write(iuo,'(1x,a69)')  klin(1:69)
+      if( nk==0 )  then
+        write(iuo,hp)
+      else
+        write(iuo,hp)  kk(1:min(4,nk))
+      endif
     endif
-  endif
-  if( js>0 )  ics = -1

-  return
-end subroutine erprnt
+    ! Also print, if required, to the terminal.
+    ! In production mode, messages are not sent to the terminal.
+    if( (kprod==0 .and. nwer+nfer<20) .or. (nfer==0 .and. jf==1) ) then
+      if( iovr==1 .and. im/=2 .and. iln1/=0 )  write(jtty,'(" ")')
+      if( im/=2 )  iln1 = 0
+      if( iln/=iln1 .and. im==2 )  write(jtty,'(1x,a69)')  klin(1:69)
+      if( im==2 )  iln1 = iln
+      if( nk==0 )  then
+        write(jtty,hp)
+      else
+        write(jtty,hp)  kk(1:min(4,nk))
+      endif
+    endif
+
+
+    ! Limit the number of error messages to the controller to 20.
+    if( kprod==0 .and. nfer+nwer==20 )  write(jtty,20)  outp
+20  format(" additional error messages on file ",a8)
+
+    ! Increment count of fatal errors or warnings.
+    if(     jf==3 )  then
+      ! don't count the "comments"
+      continue
+    elseif( jf==2 )  then
+      ! Warning only.
+      if( npp>=0 )  nwer = nwer+1
+    else
+      ! Fatal condition.
+      nfer = nfer+1
+      if( mct>=0 )  then
+        ink(1:mink) = 1
+        mnk = 1
+      endif
+    endif
+    if( js>0 )  ics = -1
+
+    return
+  end subroutine erprnt_i8_k2
+
+  subroutine erprnt_i8_k1k2(im,jf,nk,k1,k2,k3,k4,js,hf)
+    ! erprnt subroutine for explicit 8 byte integer arguments k1 and k2.
```

```
+
+    ! Modules used:
+    use mcnp_global
+    use mcnp_debug
+
+    implicit none
+
+    ! Argument declarations:
+    integer        :: im,jf,nk,k3,k4,js
+    integer(i8knd) :: k1,k2
+    !  im=1  print message only.    im=2  klin also, to terminal.
+    !  jf=1  fatal error.           jf=2  warning only.   jf=3  comment only.
+    !  nk = how many of the parameters k1 thru k4 are to be printed.
+    !  js > 0 set ics = -1;  js < 0 print to jtty only.
+    character(len=*)   :: hf ! partial printing format.
+
+    ! Local declarations:
+    integer(i8knd) :: kk(4)
+    integer        :: j
+    character(len=100) :: hp
+    character(len=20),dimension(3) :: hs = &
+      & (/  '(" fatal error.  ", ', &
+      &     '(" warning. ",      ', &
+      &     '(" comment. ",      ' /)
+
+
+    ! Move the parameters to be printed into the printing block.
+    kk = (/k1,k2,int(k3,i8knd),int(k4,i8knd)/)
+
+    ! Assemble the format from the leaders and the partial format, hf.
+    j = jf
+    if( jf<1 .or. jf>3 ) j = 1
+    hp = hs(j) // hf // ')'
+
+    ! Print the error message to the output file.
+    if( js>=0 ) then
+      if( im==1 )  write(iuo,'(" ")')
+      if( iln1<0 .and. im==2 )  write(iuo,'(1x,a69)')  klin(1:69)
+      if( nk==0 )  then
+        write(iuo,hp)
+      else
+        write(iuo,hp)  kk(1:min(4,nk))
+      endif
+    endif
+
+    ! Also print, if required, to the terminal.
+    ! In production mode, messages are not sent to the terminal.
+    if( (kprod==0 .and. nwer+nfer<20) .or. (nfer==0 .and. jf==1) ) then
+      if( iovr==1 .and. im/=2 .and. iln1/=0 )  write(jtty,'(" ")')
+      if( im/=2 )  iln1 = 0
+      if( iln/=iln1 .and. im==2 )  write(jtty,'(1x,a69)')  klin(1:69)
+      if( im==2 )  iln1 = iln
+      if( nk==0 )  then
+        write(jtty,hp)
+      else
+        write(jtty,hp)  kk(1:min(4,nk))
+      endif
+    endif
+
+
+    ! Limit the number of error messages to the controller to 20.
+    if( kprod==0 .and. nfer+nwer==20 )  write(jtty,20)  outp
+20  format(" additional error messages on file ",a8)
+
+    ! Increment count of fatal errors or warnings.
+    if(     jf==3 )  then
+      ! don't count the "comments"
+      continue
+    elseif( jf==2 )  then
+      ! Warning only.
+      if( npp>=0 )  nwer = nwer+1
+    else
+      ! Fatal condition.
+      nfer = nfer+1
+      if( mct>=0 )  then
+        ink(1:mink) = 1
+        mnk = 1
+      endif
+    endif
+    if( js>0 )  ics = -1
+
+    return
+  end subroutine erprnt_i8_k1k2
+
+end module erprnt_mod
diff  -Naurd MCNP5/Source/src/errprn.F90 MCNP5_new/Source/src/errprn.F90
--- MCNP5/Source/src/errprn.F90      2003-04-30 20:10:48.000000000 -0600
+++ MCNP5_new/Source/src/errprn.F90   2004-07-22 15:14:41.000000000 -0600
@@ -21,14 +21,14 @@
   real(dknd),       intent(in)    :: v1, v2
   character(len=*), intent(in)    :: h1, h2, hf
   integer(i8knd) :: i8_count
-  character      :: ha*16, hb*26, hc*37, hd*60
+  character      :: ha*18, hb*26, hc*37, hd*60

   ! Print warning if this is the first time encountered.
   if( n1<=0 ) then

     !$OMP CRITICAL (PRINT_OUTPUT)
     hd = hf
-    write(ha,'(" nps =",i10)')  npstc
+    write(ha,'(" nps =",i12)')  npstc
     !*** Note: next line differs from versions prior to 5
     call RN_query( count=i8_count )
     write(hb,'("nrn =",i20,1x)') i8_count
@@ -46,22 +46,24 @@
       case default
         hc = ' '
     end select
-    write(iuo,'(  " warning.  ",a60/a16,5x,a26,a37)')  hd,ha,hb,hc
+    write(iuo,'(  " warning.  ",a60/a18,5x,a26,a37)')  hd,ha,hb,hc
     if( ntasks<=1 .and. ltasks<=1 )&
-      & write(jtty,'(  " warning. ",a60/a16,5x,a37)')  hd,ha,hc
+      & write(jtty,'(  " warning. ",a60/a18,5x,a37)')  hd,ha,hc
     !$OMP END CRITICAL (PRINT_OUTPUT)

     if( npp>=0 ) then
-      !$OMP ATOMIC
+      !$OMP CRITICAL      (UPDATE_VARCOM)
       nwer = nwer+1
+      !$OMP END CRITICAL  (UPDATE_VARCOM)
     endif

   endif

   ! Increment counts and unlock multitasking as required.
   if( n1>=0 ) then
-    !$OMP ATOMIC
+    !$OMP CRITICAL        (UPDATE_VARCOM)
     n1 = n1+1
+    !$OMP END CRITICAL    (UPDATE_VARCOM)
   endif

   return
diff  -Naurd MCNP5/Source/src/etsplt.F90 MCNP5_new/Source/src/etsplt.F90
--- MCNP5/Source/src/etsplt.F90      2003-04-30 20:10:48.000000000 -0600
+++ MCNP5_new/Source/src/etsplt.F90   2004-07-22 15:14:41.000000000 -0600
@@ -6,6 +6,7 @@
   ! print esplt and tsplt importance tables if requested (ink(80)).
   use mcnp_global
   use mcnp_debug
+  use erprnt_mod
```

```
   integer :: js(2)      = (/ 0, 0 /)
   integer :: ke(2,mipt) = reshape( (/ (0,i=1,2*mipt) /), (/2,mipt/) )
diff  -Naurd MCNP5/Source/src/exemes.F90 MCNP5_new/Source/src/exemes.F90
--- MCNP5/Source/src/exemes.F90      2003-04-30 20:10:50.000000000 -0600
+++ MCNP5_new/Source/src/exemes.F90  2004-07-22 15:14:41.000000000 -0600
@@ -9,6 +9,7 @@
   use mcnp_global
   use dmmp, only: dm_nhost
   use mcnp_debug
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

@@ -28,6 +29,7 @@
   ! Do the execute line message and the inp message block.
   DO_250: do ls = 1,2

+
     call set_filenames
     hm(:) = ' '

diff  -Naurd MCNP5/Source/src/expire.F90 MCNP5_new/Source/src/expire.F90
--- MCNP5/Source/src/expire.F90      2003-04-30 20:10:52.000000000 -0600
+++ MCNP5_new/Source/src/expire.F90  2004-07-22 15:14:41.000000000 -0600
@@ -41,11 +41,11 @@
   if( mm/=0 ) then
     kd = 1
     if( kdbnps==0 ) then
-      write( hd(2),'(a,i10)')   ' source particle no. ',nps
+      write( hd(2),'(a,i12)')   ' source particle no. ',nps
       call RN_query( first=i8_first, nps=int(nps,i8knd) )
       write( hd(3),'(a,i20)') ' starting random number = ',i8_first
     else
-      write( hd(2),'(a,i10)')   ' source particle no. ',kdbnps
+      write( hd(2),'(a,i12)')   ' source particle no. ',kdbnps
       hd(3) = ' '
     endif
   endif
@@ -60,9 +60,9 @@
   ! If required, back up to history beginning.
   nst = nst+512
   if( iovr==4 .and. (ntasks>1.or.ltasks>1) )  return
-  if( mm/=0 .and. nps>=50 )  then
+  if( mm/=0 .and. nps>=50_i8knd )  then
       call backup
-      nps = nps-1
+      nps = nps-1_i8knd
       call RN_init_particle( int(nps,i8knd) )
       call vtask
       call output
diff  -Naurd MCNP5/Source/src/expung.F90 MCNP5_new/Source/src/expung.F90
--- MCNP5/Source/src/expung.F90      2003-04-30 20:10:52.000000000 -0600
+++ MCNP5_new/Source/src/expung.F90  2004-07-22 15:14:41.000000000 -0600
@@ -8,6 +8,7 @@

   use mcnp_global
   use mcnp_debug
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/ffetch.F90 MCNP5_new/Source/src/ffetch.F90
--- MCNP5/Source/src/ffetch.F90      2003-04-30 20:10:54.000000000 -0600
+++ MCNP5_new/Source/src/ffetch.F90  2004-07-22 15:14:41.000000000 -0600
@@ -3,14 +3,13 @@

 subroutine ffetch(hf,hr)
   ! Description:
-  ! Fetch file hf from CFS by the route hr.
+  ! Formerly - Fetch file hf from CFS by the route hr.
+  ! Presently - warn user that the xsdir xs library file is not present.
   use mcnp_global, only : jtty
   use mcnp_debug
   character(len=*) :: hf
   character(len=*) :: hr
   !
-  write(jtty,10)hr,hf
-10 format("CFS node:",a," ",a)
-  call expire(0,'ffetch','CFS fetching no longer possible')
+  call expire(0,'ffetch','cannot find xs library file specified in xsdir')
   return
 end subroutine ffetch
diff  -Naurd MCNP5/Source/src/FILE.list MCNP5_new/Source/src/FILE.list
--- MCNP5/Source/src/FILE.list       2003-04-30 20:09:58.000000000 -0600
+++ MCNP5_new/Source/src/FILE.list   2004-07-22 15:14:40.000000000 -0600
@@ -14,7 +14,7 @@
 covar.F90    cprinp.F90   crit1_mod.F90            crit2_mod.F90 \
 crspro.F90   crtcze.F90   dbmin.F90    dddet.F90    \
 dddiag.F90   dddlev.F90   den1.F90     den2.F90     dmmp.F90     dosef.F90    \
-dotrcl.F90   dunlev.F90   dxdiag.F90   dxtran.F90   dynamic_arrays.F90 \
+dotrcl.F90   dunlev.F90   dxtran_mod.F90           dynamic_arrays.F90 \
 echkcl.F90   electr.F90   emaker.F90   ephcom.F90   eqpbbn.F90   erf2.F90     \
 ergimp.F90   erprnt.F90   errprn.F90   escat.F90    esloss.F90   eventp.F90   \
 exemes.F90   exmg.F90     expire.F90   expirx.F90   expung.F90   extran.F90   \
@@ -25,7 +25,7 @@
 inpert.F90   inter.F90    intsec.F90   ipbc.F90     isheet.F90   \
 isos.F90     isourc.F90   issrc.F90    itally.F90   italpr.F90   \
 items.F90    iwtwnd.F90   ixsdir.F90   jbin.F90     jdecod.F90   \
-jsourc.F90   kdarg.F90    kdata.F90    keypro.F90   klein.F90    knock.F90    \
+jsourc.F90   kdarg.F90    kdata.F90    klein.F90    knock.F90    \
 kxray.F90    latcon.F90   levcel.F90   \
 levchk.F90   lgeval.F90   likebt.F90   \
 lx5_mod.F90  main.F90     mapmaz.F90   matmpy.F90   mbody.F90    \
@@ -74,7 +74,7 @@
 #################################################
 # Files not needed, depending on CONFIG options #
 #################################################
-UNWANTED_F_SRC :=    cgsdci.F90     keypro.F90
+UNWANTED_F_SRC :=    cgsdci.F90
 UNWANTED_C_SRC :=

 ifeq (,$(findstring plot,$(CONFIG)))
diff  -Naurd MCNP5/Source/src/fixcom.F90 MCNP5_new/Source/src/fixcom.F90
--- MCNP5/Source/src/fixcom.F90      2003-04-30 20:10:56.000000000 -0600
+++ MCNP5_new/Source/src/fixcom.F90  2004-07-22 15:14:41.000000000 -0600
@@ -18,25 +18,32 @@
   ! Fixcom tag:
   character(len=64), parameter, private :: fx_tag = "fixcom 07/11/2001"

-  ! Fixcom length parameters.:
-  integer, parameter :: nfixcm =& != Size of floating-point part of /FIXCM/.
+  !---------------------------------------------------------------------------------
--
+  !  (1) parameters for the length of the 3 portions of /fxcom/
+
+   integer, parameter :: nfixcm =& != Size of floating-point part of /FIXCM/.
     & 10*(1) + 1*(9) + 2*(26) + 1*(51) + 1*(maxi) + 5*(mipt) + 1*(mbng)&
     & + 2 *(mtop) + 1*(nsp) + 1 *(3*maxv) + (2*mipt*42) + 1*(mipt*3)&
     & + 1*(mipt*8) + 1*(2*mxdt) + 1*(mipt+1) + 1*(mipt*5*mxdx)&
     & + 1*(mipt*2*mxdx)
+
+  integer, parameter :: l8fixcm = 6    != Size of integer*8 part of /fxcom/.
+
   integer, parameter :: lfixcm =& != Size of integer part of /FIXCM/.
-    & 167*(1) + 3*(3) + 1*(2*6) + 1*(2*mxdt)  &
+    & 165*(1) + 3*(3) + 1*(2*6) + 1*(2*mxdt)  &
     & + 4*(maxv) + 1*(mink) + 9*(mipt) + 3*(mipt+1) + 1*(mxdt)&
     & + 3*(mxss) + 1*(nmkey) + mipt + 2*mipt + 1
-  integer, parameter :: l8fixcm = 3    != Size of integer*8 part of /FIXCM/.

-  ! Common block equivalences:
```

```
+  !-------------------------------------------------------------------------------------
--
+  !  (2) declarations for 3 arrays equivalenced to parts of  /fxcom/
+
   real(dknd)     :: gfixcm(nfixcm) != Equivalence to real part of /FIXCM/.
-  integer        :: jfixcm(lfixcm) != Equivalence to integer part of /FIXCM/.
   integer(i8knd) :: i8fixcm(l8fixcm) != Equivalence to integer*8 part of /FIXCM/.
-  equivalence (bbrem,gfixcm), (ibad,jfixcm), (RN_seed_input,i8fixcm)
+  integer        :: jfixcm(lfixcm) != Equivalence to integer part of /FIXCM/.

-  ! Fixcom reals:
+  !-------------------------------------------------------------------------------------
--
+  !  (3) declarations for /fxcom/ REALS
+
   real(dknd) :: &
     & bbrem(mtop),        &  != Bremsstrahlung energy bias factors.
     & bnum,               &  != Bremsstrahlung bias number.
@@ -76,8 +83,23 @@
     & xunru                  != Highest energy of any unresolved
                              !=  resonance probability table.

-  ! Fixcom integers:
+  !-------------------------------------------------------------------------------------
--
+  !  (4) declarations for /fxcom/ INTEGER*8
+
+  integer(i8knd) :: &
+    & niss,               &  != Number of histories in input surface source.
+    & np1,                &  != Number of histories in surface source write run.
+    & nrss,               &  != Number of tracks on input surface source file.
+    & RN_seed_input,      &  != user input, starting RN seed
+    & RN_stride_input,    &  != user input, RN stride
+    & RN_hist_input          != user input, start RN sequence with this history
+
+  !-------------------------------------------------------------------------------------
--
+  !  (5) declarations for /fxcom/ INTEGERS
+
   integer ::  &
+    & flag_speed_tally_used, & != Flag to indicate if lattice speed tally modifications
used.
+                                 !  1 = used,  -1 = not used
     & ibad,               &  != Flag for simple bremsstrahlung distribution.
     & icw,                &  != Reference cell for generated weight windows.
     & idefv(maxv),        &  != Flags for presence of variable names on SDEF.
@@ -170,7 +192,6 @@
     & nilr(mxss),         &  != Number of cells on SSR card.
     & nilw,               &  != Number of cells on SSW card.
     & nips,               &  != Source particle type.
-    & niss,               &  != Number of histories in input surface source.
     & njsr(mxss),         &  != Number of surfaces in JASR.
     & njss,               &  != Number of surfaces in JSS.
     & njsx(mxss),         &  != Number of surfaces in ISS.
@@ -185,7 +206,6 @@
     & nocoh,              &  != Flag to inhibit coherent photon scattering.
     & nodop,              &  != Flag to inhibit Doppler photon scattering.
     & nord,               &  != Number of source variables to be sampled.
-    & np1,                &  != Number of histories in surface source write run.
     & npikmt,             &  != Number of PIKMT entries.
     & npn,                &  != Length of adjustable dimension of PAN.
     & ipert,              &  != Number of PERT card keywords, dimension of RPTB.
@@ -196,7 +216,6 @@
     & nmzu,               &  != Length of MAZU array.
     & npert,              &  != Number of perturbations.
     & nrcd,               &  != Number of values in a surface-source record.
-    & nrss,               &  != Number of tracks on input surface source file.
     & nsph,               &  != Flag for spherical output surface source.
     & nsr,                &  != Source type.
     & nsrc,               &  != Number of entries on SRC card.
@@ -236,27 +255,27 @@
```

```
     & iptra2,kfl1,  indt1, ndup1(3),     naw1,  lrt1,  nmfm1, mseb1, ktl1,   &
     & ides1

-  ! Offset l<var>s with parallel k<var>s in common /itskpt/.
-  integer :: lpac = 0           != Offset for PAC array.
-  integer :: lpan = 0           != Offset for PAN array.
-  integer :: lpwb = 0           != Offset for PWB array.
-
   ! Random number generator input parameters
-  integer        :: RN_gen_input
-  ! fixcom integer*8
-  integer(i8knd) :: RN_seed_input
-  integer(i8knd) :: RN_stride_input
-  integer(i8knd) :: RN_hist_input
+  integer :: RN_gen_input       != user input, index of RN generator parameter set
+
+  !------------------------------------------------------------------------------------
--
+  !  (6) declaration for /fxcom/   (real, interger*8, integer)

   ! Fixed common -- constant after the problem is initiated.
+  ! Real
   common /fxcom/ &
     & bbrem, bnum,  calph, coincd,ddg,   ddx,   dnb,   dxw,   dxx,   ecf,    &
     & efac,  emcf,  emx,   enum,  etspl, fnw,   hsb,   rim,   rkt,   rka,    &
     & rnok,  srv,   tco,   thgf,  wc1,   wc2,   wwg,   wwp,   wwm,   wwma,   &
     & xunrl, xunru
-  !
-  ! Fixcom integers.
+
+  ! Integer*8
+  common /fxcom/ &
+    & niss, np1, nrss, RN_seed_input, RN_stride_input, RN_hist_input
+
+  ! Integer
   common /fxcom/ &
+    & flag_speed_tally_used,                                                &
     & ibad,  icw,   idefv, ides,  idrc,  iets,  ifft,  igm,   ikz,   img,    &
     & imt,   imesh, indt,  ink,   ioid,  iphot, iplt,  ipty,  isb,   ism,    &
     & ispn,  issw,  istern,istrg, its30, iunr,  ivdd,  ivdis, ivord, iwwg,   &
@@ -265,9 +284,9 @@
     & mcal,  mct,   mgegbt,mgm,   mgww,  mix,   mjss,  mlaj,  mlja,  mrkp,   &
     & mrl,   msd,   msrk,  mtasks,mww,   mxa,   mxafs, mxe,   mxf,   mxj,    &
     & mxt,   mxtr,  mxxs,  ndet,  ndnd,  ndtt,  ndx,   nee,   nets,  ngww,   &
-    & nhb,   nilr,  nilw,  nips,  niss,  njsr,  njss,  njsx,  nkxs,  nlat,   &
-    & nlev,  nlja,  nmat,  nmxf,  nnpos, nocoh, nodop, nord,  np1,   npikmt, &
-    & npn,   ipert, mnnm,  mxfp,  nmaz,  nmip,  nmzu,  npert, nrcd,  nrss,   &
+    & nhb,   nilr,  nilw,  nips,         njsr,  njss,  njsx,  nkxs,  nlat,   &
+    & nlev,  nlja,  nmat,  nmxf,  nnpos, nocoh, nodop, nord,         npikmt, &
+    & npn,   ipert, mnnm,  mxfp,  nmaz,  nmip,  nmzu,  npert, nrcd,         &
     & nsph,  nsr,   nsrc,  nsrck, nstp,  ntal,  ntop,  numb,  nvec,  nwang,  &
     & nwgeoa,nwgeom,nwgm,  nwgma, nwng,  nww,   nwwm,  nwwma, nxnx

@@ -283,178 +302,22 @@

   common /fxcom/ &
     & RN_gen_input
-  common /fxcom/ &
-    & RN_seed_input, RN_stride_input, RN_hist_input
+
+  EQUIVALENCE   (bbrem, gfixcm)
+  EQUIVALENCE   (niss, i8fixcm)
+  EQUIVALENCE   (flag_speed_tally_used, jfixcm)

   ! Length of and Equivalence array to match /itskpt/.
   integer,parameter :: ltskpt = 47

-contains
-
-  ! --------------------------------------------------------------------------
-  subroutine fx_init
-    ! Description:
-    ! Initialize fixcom variables.
-    gfixcm  = 0.0d+0
-    jfixcm  = 0
-    i8fixcm = 0
-    return
-  end subroutine fx_init
-
-  ! ---------------------------------------------------------------------------
-
-  subroutine fx_write(iu,ierr)
-    ! Description:
-    ! Write fixcom data to pre-positioned file.
-
-    ! Argument declarations:
-    integer, intent(in)  :: iu      ! file unit number.
-    integer, intent(out) :: ierr    ! status.
-
-    ! Local declarations:
-    logical           :: lopen
-    character(len=11) :: hformat
-
-    ! File must be opened and unformatted.
-    inquire(UNIT = iu, OPENED = lopen, FORM = hformat)
-
-    if( lopen .and. hformat == "UNFORMATTED") then
-      ierr = 0
-
-      ! Write tag.
-      write(iu) fx_tag
-
-      ! Write reals.
-      write(iu) &
-        & bbrem,bnum,calph,coincd,ddg,ddx,dnb,dxw,dxx,ecf,efac,emcf,&
-        & emx,enum,etspl,fnw,hsb,rim,rkt,rka,rnok,&
-        & srv,tco,thgf,wc1,wc2,wwg,wwp,wwm,wwma,xunrl,xunru
-
-      ! Write integers in separate blocks.
-      ! Generic integers.
-      write(iu) &
-        & ibad,icw,idefv,ides,idrc,iets,ifft,igm,ikz,img,imt,imesh,indt,ink,ioid,&
-        & iphot,iplt,ipty,isb,ism,ispn,issw,istern,istrg,its30,iunr,ivdd,ivdis,&
-        & ivord,iwwg,jgm,jtlx,junf,kf8,kfl,kfq,kjaq,knods,knrm,kpt,ktls,&
-        & kufil,lfcdg,lfcdj,locdt,lvcdg,lvcdj,lxs,mai,mbnk,mcal,mct,mgegbt,mgm,&
-        & mgww,mix,mjss,mlaj,mlja,mrkp,mrl,msd,msrk,mtasks,mww,mxa,mxafs,mxe,&
-        & mxf,mxj,mxt,mxtr,mxxs,ndet,ndnd,ndtt,ndx,nee,nets,ngww,nhb,nilr,nilw,&
-        & nips,niss,njsr,njss,njsx,nkxs,nlat,nlev,nlja,nmat,nmxf,nnpos,nocoh,nodop,&
-        & nord,np1,npikmt,npn,ipert,mnnm,mxfp,nmaz,nmip,nmzu,npert,nrcd,nrss,&
-        & nsph,nsr,nsrc,nsrck,nstp,ntal,ntop,numb,nvec,nwang,nwgeoa,&
-        & nwgeom,nwgm,nwgma,nwng,nww,nwwm,nwwma,nxnx
-
-      ! Variables used for dynamic allocation.
-      write(iu)&
-        & nee1,nmat1,istrg1,mix1,mxe1,iplt1,mxa1,igm1, &
-        & npert1,mnnm1,nxnx1,msd1,mbbm1,mgww1,mcal1,npn1,kpt1, &
-        & ipert1,nsc1,mxxs1,ltd1,mxt1,mxtr1,nlat1,nvec1,nwgm1,nwwm1, &
-        & nwgma1,nwma1,mlja1,mxj1,ntal1,mww1,njsx1,lit1,njsr1,nilr1, &
-        & njsw1,niwr1,mssc1,mjss1,nilw1,junf1,mkcp1,mlaf1,mxafs1, &
-        & nmzu1,mipts1,mlaj1,ndnd1,icw1,msrk1,nwwma1,nmaz1,njss1, &
-        & npikmt1,iptra1,iptra2,kfl1,indt1,ndup1,naw1,lrt1,nmfm1, &
-        & mseb1,ktl1,ides1
-
-      ! Offset l<var>s with parallel k<var>s in common /itskpt/.
-      write(iu) lpac, lpan, lpwb
-
-      ! random number generator parameters
-      write(iu) RN_gen_input,  RN_seed_input, RN_stride_input, RN_hist_input
-    else                                ! either not open or not unformatted.
-      ierr = -1                         ! return an error.
-    endif
-
```

```
-    return
-  end subroutine fx_write
-
-  ! --------------------------------------------------------------------------
-
-  subroutine fx_read(iu,ierr)
-    ! Description:
-    ! Read fixcom data from pre-positioned file.
-    ! Position verified by module tag.
-
-    ! Argument declarations:
-    integer, intent(in) :: iu       ! file unit number.
-    integer, intent(out) :: ierr    ! status.
-
-    ! Local declarations:
-    logical           :: lopen
-    character(len=11) :: hformat
-    character(len=64) :: tag
-
-    ! File must be opened and unformatted.
-    inquire(UNIT = iu, OPENED = lopen, FORM = hformat)
-
-    if ( .not. lopen .or. hformat/="UNFORMATTED" )  then
-      ierr = -1 ! return error
-
-    else
-      ! Read tag and verify it against module tag parameter.
-      read(iu) tag
-      if( tag/=fx_tag ) then
-        ierr = -2                      ! return error
-
-      else
-        ierr = 0
-        ! Read reals.
-        read(iu) &
-          & bbrem,bnum,calph,coincd,ddg,ddx,dnb,dxw,dxx,ecf,efac,emcf,&
-          & emx,enum,etspl,fnw,hsb,rim,rkt,rka,rnok,&
-          & srv,tco,thgf,wc1,wc2,wwg,wwp,wwm,wwma,xunrl,xunru
-
-        ! Read integers in separate blocks.
-        ! Generic integers.
-        read(iu) &
-        & ibad,icw,idefv,ides,idrc,iets,ifft,igm,ikz,img,imt,imesh,indt,ink,ioid,&
-        & iphot,iplt,ipty,isb,ism,ispn,issw,istern,istrg,its30,iunr,ivdd,ivdis,&
-        & ivord,iwwg,jgm,jtlx,junf,kf8,kfl,kfq,kjaq,knods,knrm,kpt,ktls,&
-        & kufil,lfcdg,lfcdj,locdt,lvcdg,lvcdj,lxs,mai,mbnk,mcal,mct,mgegbt,mgm,&
-        & mgww,mix,mjss,mlaj,mlja,mrkp,mrl,msd,msrk,mtasks,mww,mxa,mxafs,mxe,&
-        & mxf,mxj,mxt,mxtr,mxxs,ndet,ndnd,ndtt,ndx,nee,nets,ngww,nhb,nilr,nilw,&
-        & nips,niss,njsr,njss,njsx,nkxs,nlat,nlev,nlja,nmat,nmxf,nnpos,nocoh,nodop,&
-        & nord,np1,npikmt,npn,ipert,mnnm,mxfp,nmaz,nmip,nmzu,npert,nrcd,nrss,&
-        & nsph,nsr,nsrc,nsrck,nstp,ntal,ntop,numb,nvec,nwang,nwgeoa,&
-        & nwgeom,nwgm,nwgma,nwng,nww,nwwm,nwwma,nxnx
-
-      ! Variables used for dynamic allocation.
-      read(iu) &
-        & nee1,nmat1,istrg1,mix1,mxe1,iplt1,mxa1,igm1, &
-        & npert1,mnnm1,nxnx1,msd1,mbbm1,mgww1,mcal1,npn1,kpt1, &
-        & ipert1,nsc1,mxxs1,ltd1,mxt1,mxtr1,nlat1,nvec1,nwgm1,nwwm1, &
-        & nwgma1,nwma1,mlja1,mxj1,ntal1,mww1,njsx1,lit1,njsr1,nilr1, &
-        & njsw1,niwr1,mssc1,mjss1,nilw1,junf1,mkcp1,mlaf1,mxafs1, &
-        & nmzu1,mipts1,mlaj1,ndnd1,icw1,msrk1,nwwma1,nmaz1,njss1, &
-        & npikmt1,iptra1,iptra2,kfl1,indt1,ndup1,naw1,lrt1,nmfm1, &
-        & mseb1,ktl1,ides1
-
-        ! Offset l<var>s with parallel k<var>s in common /itskpt/.
-        read(iu) lpac, lpan, lpwb
-
-        ! random number generator parameters
-        read(iu) RN_gen_input,  RN_seed_input, RN_stride_input, RN_hist_input
-      endif
```

```
+  !-------------------------------------------------------------------------------------
--
+  !  (7) declaration & initialization of  lpac,lpan,lpwb  variables
+  !  Offset l<var>s with parallel k<var>s in common /itskpt/.

-    endif
-    return
-  end subroutine fx_read
+  integer :: lpac = 0           != Offset for PAC array.
+  integer :: lpan = 0           != Offset for PAN array.
+  integer :: lpwb = 0           != Offset for PWB array.

   ! -----------------------------------------------------------------------
-
-  subroutine fx_cast(mh,mx,ie)
-    ! Description:
-    ! DMMP bcast of fixcom data.
-    ! Arguments:
-    integer,intent(in)    :: mh ! action flag, 0 -> sender.
-    integer,intent(in)    :: mx ! message chunk size (max).
-    integer,intent(inout) :: ie ! return status.
-
-    call dm_bcast(mh,gfixcm, nfixcm, mx,ie)
-    call dm_bcast(mh,jfixcm, lfixcm, mx,ie)
-    call dm_bcast(mh,i8fixcm,l8fixcm,mx,ie)
-
-    return
-  end subroutine fx_cast
-
 end module fixcom
 !-
diff  -Naurd MCNP5/Source/src/flaug.F90 MCNP5_new/Source/src/flaug.F90
--- MCNP5/Source/src/flaug.F90       2003-04-30 20:10:58.000000000 -0600
+++ MCNP5_new/Source/src/flaug.F90   2004-07-22 15:14:41.000000000 -0600
@@ -14,6 +14,7 @@
   !       and variance reduction; =2 for colidp; =3 for kxray.

   use mcnp_global
+  use dxtran_mod
   use mcnp_debug

   implicit real(dknd) (a-h,o-z)
@@ -171,10 +172,7 @@
   paxtc(2,15,2) = paxtc(2,15,2)+wgt
   paxtc(3,15,2) = paxtc(3,15,2)+wgt*erg
   pwb(kpwb+2,16,icl) = pwb(kpwb+2,16,icl)+wgt
-  do i=1,ndx(2)
-    if( (xxx-dxx(2,1,i))**2+(yyy-dxx(2,2,i))**2+&
-      & (zzz-dxx(2,3,i))**2 < dxx(2,5,i) )  idx=i
-  end do
+  idx = inside_dxtran_sphere()
   if( ndet(2)/=0 ) then
     call tallyd
   endif
diff  -Naurd MCNP5/Source/src/fmesh_mod.F90 MCNP5_new/Source/src/fmesh_mod.F90
--- MCNP5/Source/src/fmesh_mod.F90   2003-11-05 17:23:12.000000000 -0700
+++ MCNP5_new/Source/src/fmesh_mod.F90       2004-07-22 15:14:41.000000000 -0600
@@ -8,6 +8,7 @@

   use mcnp_debug
   use mcnp_params, only: dknd
+  use erprnt_mod
   implicit none
   public
   save
@@ -70,14 +71,10 @@
        &  ndfb,      &  ! Number of dose function bins.
        &  ifm_card,  &  ! Flag for mesh tally FM card.
        &  nreact,    &  ! Number of reactions on the FM card.
-       &  nireact,   &  ! size of ireact array.
-       &  mat,       &  ! Material number for FM reactions.
```

```
+       &  nireact,   &  ! size of react array.
+       &  mat,       &  ! Material index number for FM reactions.
        &  outf          ! Output format:
                         !  0=column, 1=ij, 2=ik, 3=jk, 4=column-full.
-
-     integer, POINTER  :: &
-       &  ireact(:)        ! Reaction numbers for the FM card.
-
      real(dknd) ::  &
        &  fact,         &  ! Multiplication factor.
        &  fmult,        &  ! FM card multiplier.
@@ -87,6 +84,7 @@
        &  org(3)           ! Origin of mesh.

      real(dknd), POINTER :: &
+       &  react(:),     &  ! Reaction/attenuator values from the FM card.
        &  xrbin(:),     &  ! Bin values for x/r coord.
        &  yzbin(:),     &  ! Bin values for y/z coord.
        &  ztbin(:),     &  ! Bin values for z/theta  coord.
@@ -147,7 +145,7 @@
     ! Write the allocatable arrays of derived type fm
     do i = 1,nmesh
        if( fm(i)%nireact>0 ) then
-         write(iu) fm(i)%ireact
+         write(iu) fm(i)%react
        endif
        write(iu) fm(i)%xrbin,fm(i)%yzbin,fm(i)%ztbin,fm(i)%enbin
     enddo
@@ -191,13 +189,22 @@

     ! allocate the derived types
     if ( .not. allocated (fmtal) ) then
-    allocate (fmtal(nmesh),stat = is)
-    if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')
+       allocate (fmtal(nmesh),stat = is)
+       if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')
+       do i=1,nmesh
+         nullify (fmtal(i)%tally)
+       enddo
     endif

     if( .not. allocated (fm) ) then
-    allocate (fm(nmesh),stat = is)
-    if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')
+       allocate (fm(nmesh),stat = is)
+       if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')
+
+    ! nullify pointer arrays
+       do i=1,nmesh
+         nullify( fm(i)%react, fm(i)%xrbin, fm(i)%yzbin,  fm(i)%ztbin,  fm(i)%enbin,  &
+           &      fm(i)%de,     fm(i)%df,    fm(i)%fmarry, fm(i)%fmerr)
+       enddo
     endif

     ! Next read in the scalar and non-allocatable arrays  of derived type fm
@@ -211,27 +218,27 @@
     !  Allocate and read the allocatable arrays of derived type fm
     do i = 1,nmesh
       if( fm(i)%nireact>0 ) then
-        is_assoc = associated(fm(i)%ireact)
+        is_assoc = associated(fm(i)%react)
        if (.not. is_assoc ) then
-        allocate (fm(i)%ireact(fm(i)%nireact),stat = is)
-        if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')
-      endif
+           allocate (fm(i)%react(fm(i)%nireact),stat = is)
+           if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation
failure"')
+        endif
       endif
       is_assoc = associated(fm(i)%xrbin)
       if (.not. is_assoc ) then
-      allocate (fm(i)%xrbin(fm(i)%nxrb),stat = is)
-      if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')
-      allocate (fm(i)%yzbin(fm(i)%nyzb),stat = is)
-      if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')
-      allocate (fm(i)%ztbin(fm(i)%nztb),stat = is)
-      if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')
-      allocate (fm(i)%enbin(fm(i)%nenb),stat = is)
-      if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')
+         allocate (fm(i)%xrbin(fm(i)%nxrb),stat = is)
+         if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation
failure"')
+         allocate (fm(i)%yzbin(fm(i)%nyzb),stat = is)
+         if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation
failure"')
+         allocate (fm(i)%ztbin(fm(i)%nztb),stat = is)
+         if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation
failure"')
+         allocate (fm(i)%enbin(fm(i)%nenb),stat = is)
+         if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation
failure"')
       endif

       if( fm(i)%nireact>0 ) then
-        read (iu) fm(i)%ireact
+        read (iu) fm(i)%react
       endif
       read (iu) fm(i)%xrbin,fm(i)%yzbin,fm(i)%ztbin,fm(i)%enbin
     enddo
@@ -240,14 +247,14 @@
     do i = 1,nmesh
       if( fm(i)%intrpol>0 ) then
         is_assoc = associated(fm(i)%de)
-      if( .not. is_assoc )  then
-        allocate (fm(i)%de(fm(i)%ndfb),stat = is)
-        if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')
-        allocate (fm(i)%df(fm(i)%ndfb),stat = is)
-        if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')
+        if( .not. is_assoc )  then
+           allocate (fm(i)%de(fm(i)%ndfb),stat = is)
+           if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation
failure"')
+           allocate (fm(i)%df(fm(i)%ndfb),stat = is)
+           if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation
failure"')
+        endif
         read(iu) fm(i)%de,fm(i)%df
       endif
-      endif
     enddo


@@ -259,21 +266,21 @@
       ie = fm(i)%nenb-1
       is_assoc = associated(fm(i)%fmarry)
       if( .not. is_assoc )  then
-      allocate (fm(i)%fmarry(ix,iy,iz,ie,ntasks+1),stat = is)
-      if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')
-      allocate (fm(i)%fmerr(ix,iy,iz,ie,ntasks+1),stat = is)
-      if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')
-      allocate (fmtal(i)%tally(ix,iy,iz,ie,ntasks+1),stat = is)
-      if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')
+         allocate (fm(i)%fmarry(ix,iy,iz,ie,ntasks+1),stat = is)
+         if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation
failure"')
+         allocate (fm(i)%fmerr(ix,iy,iz,ie,ntasks+1),stat = is)
+         if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation
failure"')
+         allocate (fmtal(i)%tally(ix,iy,iz,ie,ntasks+1),stat = is)
+         if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation
failure"')
       endif

-       read(iu)  fm(i)%fmarry(:,:,:,:,1), &
-      &          fm(i)%fmerr( :,:,:,:,1)
+      read(iu)  fm(i)%fmarry(:,:,:,:,1), &
+        &       fm(i)%fmerr(:,:,:,:,1)
        do j=2,ntasks+1
           fm(i)%fmarry(:,:,:,:,j) = 0
           fm(i)%fmerr(:,:,:,:,j) = 0
        enddo
-     fmtal(i)%tally = 0
+       fmtal(i)%tally = 0
      enddo

     ! Allocate scratch arrays
@@ -285,13 +292,13 @@
     i_size_bins=i_size_bins*0.2/ntasks + 1

     if( .not. allocated (i_bins) ) then
-    allocate (i_bins(5,i_size_bins,ntasks),stat = is)
-    if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')
+       allocate (i_bins(5,i_size_bins,ntasks),stat = is)
+       if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')
     endif

     if( .not. allocated (num_bins)) then
-    allocate (num_bins(ntasks),stat = is)
-    if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')
+       allocate (num_bins(ntasks),stat = is)
+       if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')
     endif
     num_bins=0
     i_bins=0
@@ -304,11 +311,12 @@
   subroutine ifmesh_print
     ! Subroutine to write initial information about the mesh tallies to OUTP

-    use mcnp_global, only:ink
+    use mcnp_global, only:ink,nmt
     use mcnp_params, only:iuo

-    integer      ::    i,j        ! Loop variables
-    character(6)  :: intpol_mode   ! Dose response function interpolation mod
+    integer       :: i,ii,iw,j,k  ! Loop variables & misc integers
+    character(6)  :: intpol_mode  ! Dose response function interpolation mod
+    character(13) :: ht(18)       ! FM card reaction list/attenuation set

     do i = 1,nmesh

@@ -331,13 +339,47 @@
        endif

        ! print the energy-dependent-multiplier bins.
-       if( fm(i)%nreact/=0 .or. fm(i)%fmult/=1. ) then
-          if( fm(i)%nreact/=0 ) then
+       if ( fm(i)%ifm_card /= 0 .or. fm(i)%fmult /= 1 ) then
+          if ( fm(i)%nreact==0 .and. fm(i)%ifm_card > -1 ) then
             write(iuo,  &
               & '(/  " all scores are multiplied by",es13.5)') fm(i)%fmult*fm(i)%fact
-          else
-             write(iuo,'(/  " multiplier bins"/ " att",2x, "constant",4x, "material",&
-               & 3x,  "reactions or material-rho*x pairs")')
+          else
+             if ( fm(i)%ifm_card == -1 ) then
+               write(iuo,'(/  " multiplier bins"/ 6x, "constant")')
+               write(iuo,'(1pe16.5,4x, "tracks")') fm(i)%fmult
+             else if ( fm(i)%ifm_card == -2 ) then
+               write(iuo,'(/  " multiplier bins"/ 6x, "constant")')
+                write(iuo,'(1pe16.5,4x,  "1/velocity")') fm(i)%fmult
+             else
+               if ( fm(i)%mat /= -1 ) then      ! FM card contains a reaction list
+                 write(iuo,'(/  " multiplier bins"/ 6x, "constant",4x, "material",&
+                   & 3x,  "reactions")')
+                 do iw=1,fm(i)%nreact
+                   ii = mod(iw-1,14)+1
+                   k = fm(i)%react(iw)
+                   ht(ii) = '     :'
+                   if ( k /= 1000003 ) write(ht(ii)(1:6),'(i6)') k
+                   if ( ii < 14 .and. iw < fm(i)%nreact ) cycle
+                   if ( iw <= 14 ) then
+                     if ( fm(i)%mat /= 0 ) then
+                       write(iuo,'(3x,1pe13.5,i7,5x,14a6)') &
+                         &  fm(i)%fmult,nmt(fm(i)%mat),(ht(k),k=1,ii)
+                     else
+                       write(iuo,'(3x,1pe13.5,"  default",3x,14a6)') &
+                         &  fm(i)%fmult,(ht(k),k=1,ii)
+                     endif
+                   endif
+                   if ( iw > 14 )  write(iuo,'(28x,14a6)') (ht(k),k=1,ii)
+                 enddo
+               else                             ! FM card contains an attenuator set
+                 write(iuo,'(/  " multiplier bins"/ 6x, "constant",15x,  &
+                    &   "material-rho*x pairs")')
+                 write (iuo,340) fm(i)%fmult,(nmt(int(fm(i)%react(j))),fm(i)%react(j+1),
&
+                   &  j=1,fm(i)%nreact,2)
+340              format(1pe16.5,i17,e12.4,4(i6,e12.4)/  &
+                   & (i33,e12.4,i6,e12.4,i6,e12.4,i6,e12.4,i6,e12.4))
+               endif
+             endif
           endif
        endif

@@ -435,6 +477,13 @@

       nmesh = 0     !reset nmesh to zero for 2nd read of INP file

+    ! nullify pointer arrays
+       do i=1,nmesh
+         nullify( fm(i)%react, fm(i)%xrbin, fm(i)%yzbin,  fm(i)%ztbin,  fm(i)%enbin,  &
+           &      fm(i)%de,     fm(i)%df,    fm(i)%fmarry, fm(i)%fmerr )
+         nullify( fmtal(i)%tally )
+       enddo
+
     endif
     return
   end subroutine fmesh_allocate
@@ -447,7 +496,7 @@
     ! fm(kk)%ifm_card = FM card flag
     ! fm(kk)%mat = index of the material
     ! fm(kk)%nreact= number of mt reactions (negative if elastic or total)
-    ! fm(kk)%ireact(1,2,...) = mt reaction numbers
+    ! fm(kk)%react(1,2,...) = mt reaction numbers or attenuation factors

     use mcnp_global

@@ -466,7 +515,7 @@
     if( fm(kk)%mat<0. )  go to 220
     h = 0.
     mk = fm(kk)%mat
-    if( mk==0  )  go to 200
+    if( mk==0  )  mk = mat(icl)
     if( fm(kk)%ipt==2 )  go to 140

     ! neutron multipliers
@@ -533,7 +582,7 @@
        f = 0.
        g = 1.
        do n = 1,mod(int(abs(fm(kk)%nreact)),10000000)
-          mt = fm(kk)%ireact(n)
```

```
+          mt = fm(kk)%react(n)
           if( mt==1000003 ) then
              f = f+g
              g = 1.
@@ -554,11 +603,11 @@
        f = 0.
        g = 1.
        do n = 1,mod(int(fm(kk)%nreact),10000000)
-          if( fm(kk)%ireact(n)==1000003 ) then
+          if( fm(kk)%react(n)==1000003 ) then
              f = f+g
              g = 1.
           else
-             j = -fm(kk)%ireact(n)
+             j = -fm(kk)%react(n)
              if( mcal==0 ) then
                 xs = rtc(krtc+j,iex)
                 if( j>1 )  xs = xs-rtc(krtc+j-1,iex)
@@ -592,7 +641,7 @@
     h = 0.
     ttn = 253e-10
     do n = 1,int(fm(kk)%nreact),2
-       mk = fm(kk)%ireact(n)
+       mk = fm(kk)%react(n)
        if( mk==0   )  cycle
        if( mcal==0 ) then
          if( ipt==1 )  jf = 1
@@ -605,7 +654,7 @@
              totm = totm+getxs(1)*fme(m)
          enddo
        endif
-      h = h+ fm(kk)%ireact(n+1)*totm
+      h = h+ fm(kk)%react(n+1)*totm
     enddo
     if( h>80.  )  t = 0.
     if( h<=80. )  t = t*exp(-h)
@@ -660,7 +709,7 @@

       call msg_put (fm(i)%nireact)
       if( fm(i)%nireact>0 ) then
-         call msg_put (fm(i)%ireact,1,fm(i)%nireact)
+         call msg_put (fm(i)%react,1,fm(i)%nireact)
       endif

       call msg_put (fm(i)%axs,  1, 3)
@@ -704,6 +753,13 @@
    allocate (fmtal(nmesh),stat = is)
    if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation failure"')

+    ! nullify pointer arrays
+    do i=1,nmesh
+      nullify( fm(i)%react, fm(i)%xrbin, fm(i)%yzbin,  fm(i)%ztbin,  fm(i)%enbin,  &
+        &      fm(i)%de,     fm(i)%df,    fm(i)%fmarry, fm(i)%fmerr )
+      nullify( fmtal(i)%tally )
+    enddo
+
    Loop1: do i = 1,nmesh
       call msg_get (fm(i)%id)
       call msg_get (fm(i)%ipt)
@@ -723,11 +779,11 @@
       call msg_get (fm(i)%fact)
       call msg_get (fm(i)%fmult)

-      call msg_get(fm(i)%nireact)  ! get size of ireact array
+      call msg_get(fm(i)%nireact)  ! get size of react array
       if( fm(i)%nireact>0 ) then
-         allocate (fm(i)%ireact(fm(i)%nireact),stat = is)
+         allocate (fm(i)%react(fm(i)%nireact),stat = is)
          if(is/=0) call erprnt(1,1,0,0,0,0,0,1,' "mesh tally memory allocation 
failure"')
-         call msg_get (fm(i)%ireact,1,fm(i)%nireact)
```

```
+          call msg_get (fm(i)%react,1,fm(i)%nireact)
        endif

        call msg_get (fm(i)%axs,  1, 3)
@@ -919,7 +975,7 @@
     ! Called from hstory

     use mcnp_params, only:dknd
-    use mcnp_global, only:trf,huge,rho,icl,ktask,kc8
+    use mcnp_global, only:trf,huge_float,rho,icl,ktask,kc8

     integer   , intent(in) :: ipt
     real(dknd), intent(in) :: x,y,z,u,v,w,erg,wgt,d
@@ -932,7 +988,7 @@
     real(dknd) :: dstnce_to_travel  ! Distance remaining for the particle to travel
     real(dknd) :: dstnce_to_bin     ! Distance to the next mesh bin boundary

-    real(dknd), parameter :: tiny = 1/huge
+    real(dknd), parameter :: tiny = 1/huge_float
     integer  :: i,j,k,il,iu,im,ien,ixr,iyz,izt,kt,kt1

     ! return if kcode problem is not settled
@@ -1004,7 +1060,7 @@
                 else if( r(1)-fm(i)%xrbin(fm(i)%nxrb)>=1d-10 ) then
                    cycle Loop1
                 else
-                   dr(1) = huge
+                   dr(1) = huge_float
                 endif
              else if( dircos(1)<-tiny ) then
                 if( r(1)-fm(i)%xrbin(fm(i)%nxrb)>1d-10 ) then
@@ -1014,10 +1070,10 @@
                 else if( r(1)-fm(i)%xrbin(1)<= -1d-10 ) then
                    cycle Loop1
                 else
-                   dr(1) = huge
+                   dr(1) = huge_float
                 endif
              else
-                   dr(1) = huge
+                   dr(1) = huge_float
              endif

             if( dircos(2)>tiny ) then
@@ -1028,7 +1084,7 @@
                 else if( r(2)-fm(i)%yzbin(fm(i)%nyzb)>=1d-10 ) then
                    cycle Loop1
                 else
-                   dr(2) = huge
+                   dr(2) = huge_float
                 endif
              else if( dircos(2)<-tiny ) then
                 if( r(2)-fm(i)%yzbin(fm(i)%nyzb)>1d-10 ) then
@@ -1038,10 +1094,10 @@
                 else if( r(2)-fm(i)%yzbin(1)<= -1d-10 ) then
                    cycle Loop1
                 else
-                   dr(2) = huge
+                   dr(2) = huge_float
                 endif
              else
-                dr(2) = huge
+                dr(2) = huge_float
              endif

             if( dircos(3)>tiny ) then
@@ -1052,7 +1108,7 @@
                 else if( r(3)-fm(i)%ztbin(fm(i)%nztb)>=1d-10 ) then
                    cycle Loop1
                 else
-                   dr(3) = huge
```

```
+                   dr(3) = huge_float
                 endif
              else if( dircos(3)<-tiny ) then
                 if( r(3)-fm(i)%ztbin(fm(i)%nztb)>1d-10 ) then
@@ -1062,10 +1118,10 @@
                 else if( r(3)-fm(i)%ztbin(1)<= -1d-10 ) then
                    cycle Loop1
                 else
-                   dr(3) = huge
+                   dr(3) = huge_float
                 endif
              else
-                dr(3) = huge
+                dr(3) = huge_float
              endif

              dstnce_to_bin = min(dr(1),dr(2),dr(3))
@@ -1115,27 +1171,18 @@

        ! find distance to next cell
        Loop3: do while (dstnce_to_travel>1d-9*d)
-          dr(1) =  huge
+          dr(1) =  huge_float
           if( dircos(1)>0.)  dr(1) = (fm(i)%xrbin(ixr+1)-r(1))/dircos(1)
           if( dircos(1)<0.)  dr(1) = (fm(i)%xrbin(ixr)-r(1))/dircos(1)
-          dr(2) =  huge
+          dr(2) =  huge_float
           if( dircos(2)>0.)  dr(2) = (fm(i)%yzbin(iyz+1)-r(2))/dircos(2)
           if( dircos(2)<0.)  dr(2) = (fm(i)%yzbin(iyz)-r(2))/dircos(2)
-          dr(3) =  huge
+          dr(3) =  huge_float
           if( dircos(3)>0.)  dr(3) = (fm(i)%ztbin(izt+1)-r(3))/dircos(3)
           if( dircos(3)<0.)  dr(3) = (fm(i)%ztbin(izt)-r(3))/dircos(3)
           dstnce_to_bin = min(dr(1),dr(2),dr(3),dstnce_to_travel)
           if( dstnce_to_bin>1d-15*d ) then

-            t = 1.*fm(i)%fmult
-            if( fm(i)%fmult<0)  t = -t*rho(icl)
-            if( fm(i)%ifm_card/=0 ) then
-              call wtmult_fmesh(t,i)
-            endif
-
-            ! Score the track length
-            dt = dstnce_to_bin
-
             ! Enter values in scratch arrays
              if( i_size_bins>num_bins(kt1) ) then
                 num_bins(kt1)=num_bins(kt1)+1
@@ -1146,23 +1193,21 @@
                 i_bins(5,num_bins(kt1),kt1)=ien
              endif

-            ! No dose response function
-
-            if( fm(i)%intrpol==0 ) then
-              if( fm(i)%icx==0 ) then
-                score=dt*wgt*t
-              else if ( fm(i)%icx==1 ) then
-                score=dt*wgt*erg*t
-              endif
+            ! Score the track length

-            ! Dose response function
+            t = 1._dknd
+            if( fm(i)%ifm_card/=0 ) then
+               call wtmult_fmesh(t,i)
+            endif
+            t = t*fm(i)%fmult
+            if( fm(i)%fmult < 0 ) t = -t*rho(icl)

-            else if( fm(i)%intrpol==1 ) then
-              if( fm(i)%icx==0 ) then
-                 score=dt*wgt*t*dosef_fmesh(erg,i)
-              else if( fm(i)%icx==1 ) then
-                 score=dt*wgt*t*erg*dosef_fmesh(erg,i)
-              endif
+            if ( fm(i)%ifm_card == -1 ) then ! Special tally multiplier -- # of tracks
+              score = t
+            else
+              score = dstnce_to_bin*wgt*t
+              if ( fm(i)%icx == 1 ) score = score*erg
+              if ( fm(i)%intrpol /= 0 ) score = score*dosef_fmesh(erg,i)
             endif

             fmtal(i)%tally(ixr,iyz,izt,ien,kt)=
fmtal(i)%tally(ixr,iyz,izt,ien,kt)+score
@@ -1203,12 +1248,12 @@
     ! Algorithm based on cylindrical weight-window mesh

     use mcnp_params, only:dknd,one,pie
-    use mcnp_global, only:trf,huge,rho,icl,ktask
+    use mcnp_global, only:trf,huge_float,rho,icl,ktask

     integer   , intent(in) :: i
     real(dknd), intent(in) :: x,y,z,u,v,w,erg,wgt,d

-    real(dknd), parameter :: tiny = 1/huge
+    real(dknd), parameter :: tiny = 1/huge_float

     real(dknd) :: dstnce_to_travel  ! Distance remaining for the particle to travel
     real(dknd) :: dstnce_to_bin     ! Distance to the next mesh bin boundary
@@ -1297,7 +1342,7 @@
           rv = r(1)*dircos(1)+r(2)*dircos(2)
           ! if rv >= 0, particle is moving away from the mesh
           if( r(1)>fm(i)%xrbin(fm(i)%nxrb).and.rv>=0 ) return
-          dr(1) = huge
+          dr(1) = huge_float
           if( dcos/=0. ) then
              rs = r(1)**2+r(2)**2
              a = rv/dcos
@@ -1311,7 +1356,7 @@
              endif
           endif
           if( dr(1)<=-1d-10 ) then
-            dr(1) = huge
+            dr(1) = huge_float
           else if( dr(1)<=1d-10 ) then
             dr(1) = 1d-10
           endif
@@ -1324,7 +1369,7 @@
              else if ( cylr(2)-fm(i)%yzbin(fm(i)%nyzb)>=1d-10 ) then
                  return
              else
-                dr(2) = huge
+                dr(2) = huge_float
              endif
           else if( dircos(3)<-tiny ) then
              if( cylr(2)-fm(i)%yzbin(fm(i)%nyzb)>1d-10 ) then
@@ -1334,10 +1379,10 @@
              else if( cylr(2)-fm(i)%yzbin(1)<= -1d-10 ) then
                 return
              else
-                dr(2) = huge
+                dr(2) = huge_float
              endif
           else
-             dr(2) = huge
+             dr(2) = huge_float
           endif

           dstnce_to_bin = min(dr(1),dr(2))
@@ -1390,7 +1435,7 @@
```

```
     ! find distance to next mesh cell
     do while (dstnce_to_travel>1d-15*d)
        rv = r(1)*dircos(1)+r(2)*dircos(2)
-       dr(1) =  huge
+       dr(1) =  huge_float
        if( dcos/=0. ) then
           rs = r(1)**2+r(2)**2
           a = rv/dcos
@@ -1412,7 +1457,7 @@
           endif
        endif

-       dr(2) = huge
+       dr(2) = huge_float
        if( dircos(3)>0. ) then
           dr(2) = (fm(i)%yzbin(iyz+1)-r(3))/dircos(3)
           nextbin(2) = 1
@@ -1422,7 +1467,7 @@
        endif
        !   theta
        !   if single bin, point is always in it
-       dr(3) = huge
+       dr(3) = huge_float
        if( fm(i)%nztb/=2 ) then
           rp = -dircos(1)*r(2)+dircos(2)*r(1)
           if( rp<0. ) then
@@ -1433,7 +1478,7 @@
              if( vn<0 ) then
                 dr(3) = (st*r(1)-ct*r(2))/vn
                 nextbin(3) = -1
-                if( dr(3)<-1d-9 ) dr(3)=huge  !allow for round-off errors
+                if( dr(3)<-1d-9 ) dr(3)=huge_float  !allow for round-off errors
              endif
           else
              t = fm(i)%ztbin(izt+1)
@@ -1443,21 +1488,13 @@
              if( vn>0 ) then
                 dr(3) = (st*r(1)-ct*r(2))/vn
                 nextbin(3) = 1
-                if( dr(3)<-1d-9 ) dr(3)=huge  !allow for round-off errors
+                if( dr(3)<-1d-9 ) dr(3)=huge_float  !allow for round-off errors
              endif
           endif
        endif

        dstnce_to_bin = min(dr(1),dr(2),dr(3),dstnce_to_travel)
        if( dstnce_to_bin>1d-10*d ) then
-          t = 1.*fm(i)%fmult
-          if( fm(i)%fmult<0) t = -t*rho(icl)
-          if( fm(i)%ifm_card/=0 ) then
-             call wtmult_fmesh(t,i)
-          endif
-
-             ! Score the track length
-          dt = dstnce_to_bin

          ! Enter values in scratch arrays
           if( i_size_bins>num_bins(kt1) ) then
@@ -1469,23 +1506,20 @@
              i_bins(5,num_bins(kt1),kt1)=ien
           endif

-         ! No dose response function
-
-         if( fm(i)%intrpol==0 ) then
-           if( fm(i)%icx==0 ) then
-             score=dt*wgt*t
-           else if ( fm(i)%icx==1 ) then
-             score=dt*wgt*erg*t
-           endif
-
```

```
-         ! Dose response function
+         ! Score the track length
+         t = 1._dknd
+         if( fm(i)%ifm_card /= 0 ) then
+            call wtmult_fmesh(t,i)
+         endif
+         t = t*fm(i)%fmult
+         if( fm(i)%fmult <0 ) t = -t*rho(icl)

-         else if(fm(i)%intrpol==1) then
-           if(fm(i)%icx==0) then
-              score=dt*wgt*t*dosef_fmesh(erg,i)
-           else if(fm(i)%icx==1) then
-              score=dt*wgt*t*erg*dosef_fmesh(erg,i)
-           endif
+         if ( fm(i)%ifm_card == -1 ) then ! Special tally multiplier -- # of tracks
+           score = t
+         else
+           score = dstnce_to_bin*wgt*t
+           if ( fm(i)%icx == 1 ) score = score*erg
+           if ( fm(i)%intrpol /= 0 ) score = score*dosef_fmesh(erg,i)
          endif

          fmtal(i)%tally(ixr,iyz,izt,ien,kt)=  fmtal(i)%tally(ixr,iyz,izt,ien,kt)+score
@@ -1723,18 +1757,43 @@
     use mcnp_global,   only: emx,mct,fpi
     implicit real(dknd) (a-h,o-z)
     integer i,j,jj,k,kk,l
+    integer        :: maxbin = 1
+    integer        :: n_reclength = 1
     real(dknd), ALLOCATABLE :: rel_err(:,:)
     real(dknd) sp_norm

     ! Get  the source particle normalization.
     sp_norm = 1./fpi

+    ! Determine maximum record length
+    do i=1,nmesh
+       maxbin=max (fm(i)%nxrb,fm(i)%nyzb,fm(i)%nztb,maxbin)
+       select case (fm(i)%outf)
+         !
+         !  column format
+         !
+       case (0,4)
+         n_reclength = max (105,35+10*maxbin,n_reclength)
+         !
+         !  ij, ik format
+         !
+       case (1,2)
+         n_reclength = max (105,17+12*fm(i)%nxrb,35+10*maxbin,n_reclength)
+         !
+         !  jk format
+         !
+       case (3)
+         n_reclength = max (105,17+12*fm(i)%nyzb,35+10*maxbin,n_reclength)
+       end select
+    enddo
+
     !  open the meshtal file
+
     if( lmtout ) then
-       rewind(iumt)
+       open(iumt,status='unknown',file=meshtal,form='formatted',recl=n_reclength)
     else
        call unique (meshtal,jtty)
-       open(iumt,status='new',file=meshtal,form='formatted',recl=262144)
+       open(iumt,status='new',file=meshtal,form='formatted',recl=n_reclength)
        lmtout = .true.
     endif
     if(mct >=0) write(iumt,'(a6, " version ",a5, " ld=",a8,2x, "probid = ",a19 )')  &
@@ -1771,7 +1830,7 @@
```

```
          write (iumt,"(4x,'Y direction:',10000f10.2)")(fm(j)%yzbin(k),k=1,fm(j)%nyzb)
          write (iumt,"(4x,'Z direction:',10000f10.2)")(fm(j)%ztbin(k),k=1,fm(j)%nztb)
        else if( fm(j)%icrd==2 ) then
-          write(iumt,'(x," Cylinder origin at ",3es10.2,", axis in ",3es10.3,   &
+          write(iumt,'(1x," Cylinder origin at ",3es10.2,", axis in ",3es10.3,   &
               & " direction")') fm(j)%org,fm(j)%axs
          write (iumt,"(4x,'R direction:',10000f10.2)")(fm(j)%xrbin(k),k=1,fm(j)%nxrb)
          write (iumt,"(4x,'Z direction:',10000f10.2)")(fm(j)%yzbin(k),k=1,fm(j)%nyzb)
@@ -2103,28 +2162,42 @@

        ! First set up the dose response functions

-       ! Check to see if the number of energy values equal the number of
-       ! response function values
-       if( ndeitm(i)/=ndfitm(i) ) then
-          call erprnt(1,1,1,meshid(i),0,0,0,0,&
-               & '  "number of values on de and df of mesh",i4,  " not equal."')
-       endif
        ! If meshid = 0, enter the default response function for
        !  those mesh tallies that do not have response functions assigned
        if( meshid(i)==0 ) then
+
+       ! Compenstate for incorrect number of items on de,df card if 'lin' is specified
+          indx_de = 0
+          indx_df = 0
+          if ( intrpol(i) > 2 ) indx_de = 1
+          if ( mod(intrpol(i),2) == 0 ) indx_df = 1
+
+       ! Check to see if the number of energy values equal the number of
+       ! response function values
+          if( ndeitm(i)-indx_de /= ndfitm(i)-indx_df ) then
+             call erprnt(1,1,1,meshid(i),0,0,0,0,&
+                  & '  "number of values on de and df of mesh",i4,  " not equal."')
+          endif
           do j = 1,nmesh
              if( fm(j)%ndfb==0 ) then  !ndfb = 0 for all meshes w/out dose functions
                fm(j)%intrpol = intrpol(i)
-               fm(j)%ndfb  =  ndfitm(i)
+               fm(j)%ndfb  =  ndfitm(i)-indx_df
                allocate (fm(j)%de(fm(j)%ndfb))
                allocate (fm(j)%df(fm(j)%ndfb))
                do k = 1,fm(j)%ndfb
-                  fm(j)%de(k) = detmp(i,k)
-                  fm(j)%df(k) = dftmp(i,k)
+                  fm(j)%de(k) = detmp(i,k+indx_de)
+                  fm(j)%df(k) = dftmp(i,k+indx_df)
                enddo
             endif
          enddo
        else    !  find the mesh for this response function and fill in the values
+
+       ! Check to see if the number of energy values equal the number of
+       ! response function values
+          if( ndeitm(i) /= ndfitm(i) ) then
+             call erprnt(1,1,1,meshid(i),0,0,0,0,&
+                  & '  "number of values on de and df of mesh",i4,  " not equal."')
+          endif
          do j = 1,nmesh
             if( fm(j)%id==meshid(i) ) then
                if( fm(j)%ndfb==0 ) then
@@ -2165,19 +2238,18 @@

     !  find the mesh for this FM card and fill in the values
   nfm_do: do i = 1,nfm
-       iflg = 0    ! zero mesh tally id flag
-  nmesh_do: do j = 1,nmesh
+    nmesh_do: do j = 1,nmesh
          if( fm(j)%id==meshfm(i) ) then
-            iflg = 1.
+            kl = fm(j)%id
             ! count bins
```

```
              k = 1
-             do while (fmtmp(k,i)/=0)
+             do while (fmtmp(k,i) /= 0 .or. fmtmp(k+1,i) /= 0)
                 k = k+1
              enddo
              n = k-1
              fm(j)%nireact = n+10
-             allocate(fm(j)%ireact(fm(j)%nireact))
-             fm(j)%ireact = 0
+             allocate(fm(j)%react(fm(j)%nireact))
+             fm(j)%react = 0

              if( fm(j)%ipt==3 .and. n>1 )  call erprnt(1,1,0,0,0,0,0,0,0,&
                   & '  "fm for electron mesh tally has more than just a constant."')
@@ -2212,7 +2284,7 @@
                 fm(j)%fmult = fmtmp(k1+1,i)
                 ni = ni+1
                 if( ni>1) call erprnt(1,1,1,meshfm(i),0,0,0,0,&
-                     & '  "mesh tally ",i4,": only one mulitplier set per mesh tally"')
+                     & '  "mesh tally ",i4," only one mulitplier set per mesh tally"')
 420             continue
                 if( m+m1+max(0,k2)>2 .or. m<0 .or. k==n .and. (m/=0) )  go to 580
              enddo Loop_420
@@ -2253,12 +2325,16 @@
                 if( fmtmp(k+2,i)==1000002 )  go to 490
                 ii = -1
                 if( fmtmp(k+1,i)<0 )  go to 480
-                do ii = 1,nmat
-                   if( nmt(ii)==fmtmp(k+1,i)) exit
-                enddo
+                if (fmtmp(k+1,i) == 0 ) then
+                   ii = 0
+                else
+                   do ii = 1,nmat
+                     if( nmt(ii)==fmtmp(k+1,i)) exit
+                   enddo
+                endif
 480             continue
                 fm(j)%mat = ii
-                call erprnt(1,2,2,ii,fm(j)%id,0,0,0, &
+                if (fm(j)%mat > 0 ) call erprnt(1,2,2,nmt(ii),fm(j)%id,0,0,0, &
           & '" FM card uses material ",i4," cross sections over all of mesh tally ",i4')
                 kq = 1
                 l1 = ld+1
@@ -2271,23 +2347,23 @@
                 cycle Loop_560
 500             continue
                 if( k<k1+3 )  cycle Loop_560
-                if( ii>0 )  go to 540
+                if( ii>=0 )  go to 540
                 if( ii==-2 )  go to 530
                 do jj = 1,nmat
                    if( nmt(jj)==fmtmp(k,i) )  go to 520
                 enddo
 520             continue
-                fm(j)%ireact(ld) = jj
+                fm(j)%react(ld-2) = jj
                 fm(j)%nreact = fm(j)%nreact+2.
                 ii = -2
                 go to 550
 530             continue
-                fm(j)%ireact(ld) = fmtmp(k,i)
+                fm(j)%react(ld-2) = fmtmp(k,i)
                 ii = -3
                 go to 550
 540             continue
-                fm(j)%ireact(ld-2) = fmtmp(k,i)
-                jj = fm(j)%ireact(ld-2)
+                fm(j)%react(ld-2) = fmtmp(k,i)
+                jj = fm(j)%react(ld-2)
                 if( fm(j)%ipt==1 .and. jj<=-6 )  itfxs = 1
                 if( fm(j)%ipt==1 .and. jj<-8 .or. &
                      & fm(j)%ipt==2 .and. jj<-6 )  call erprnt(1,1,2,j,kl,0,0,0,&
@@ -2302,7 +2378,7 @@
              go to 590
 580          continue
                 call erprnt(1,1,1,kl,0,0,0,0,&
-                  & '  "fm card of tally",i4,  " has wrong format."')
+                  & '  "fm card of mesh tally",i4,  " has wrong format."')

 590          continue
              exit nmesh_do
@@ -2367,16 +2443,16 @@
   !-------------------------------------------------------------------------------------
----

   subroutine fmesh_vtask(ktask)
-
+
     ! Subroutine to merge the mesh tally values into the 1st array bin.
     ! This needs to be done even if run as a sequential code
     ! called from vtask
-
+
     integer, intent(in) :: ktask
     integer  :: i,kt
-
+
     kt = ktask+2
-    do i = 1,nmesh
+    do i=1,nmesh
       fm(i)%fmarry(:,:,:,:,1) = fm(i)%fmarry(:,:,:,:,1)+fm(i)%fmarry(:,:,:,:,kt)
       fm(i)%fmerr(:,:,:,:,1) = fm(i)%fmerr(:,:,:,:,1)+fm(i)%fmerr(:,:,:,:,kt)
       fm(i)%fmarry(:,:,:,:,kt) = 0
@@ -2418,3 +2494,4 @@
   !-------------------------------------------------------------------------------------
----

 end module fmesh_mod
+
diff  -Naurd MCNP5/Source/src/getexm.F90 MCNP5_new/Source/src/getexm.F90
--- MCNP5/Source/src/getexm.F90       2003-04-30 20:11:00.000000000 -0600
+++ MCNP5_new/Source/src/getexm.F90   2004-07-22 15:14:41.000000000 -0600
@@ -26,7 +26,6 @@
   character(len=80) :: hm
   integer :: eol

-#ifndef PCDOS
   integer*4 ia
   hm = ' '
   j  = 0
@@ -45,16 +44,10 @@
   end do
   hm(:) = hm(1:eol)

-#endif
-
 #ifdef HPUX
   $hp9000_800 system off
 #endif /*def.hpux*/

-#ifdef PCDOS
-  call getcl(hm)
-#endif /*def.pcdos*/
-

   return
 end subroutine getexm
diff  -Naurd MCNP5/Source/src/getxst.F90 MCNP5_new/Source/src/getxst.F90
--- MCNP5/Source/src/getxst.F90       2003-04-30 20:11:02.000000000 -0600
+++ MCNP5_new/Source/src/getxst.F90   2004-07-22 15:14:41.000000000 -0600
@@ -11,6 +11,7 @@
```

```
   use mcnp_global
   use mcnp_debug
   use dynamic_arrays
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

@@ -24,7 +25,7 @@

   ! Expand xss array to accomodate the largest table,
   !     or exs array to hold all tables
-  et = huge
+  et = huge_float
   em = 0.
   ih = 0
   th = 0.
@@ -93,20 +94,12 @@
     inquire( file=ha,  exist=file_exists )
     if( .not.file_exists ) then
       if( len_trim(hdpath)>0 ) then
-#ifdef PCDOS
-        ha = hdpath(1:len_trim(hdpath))//'\'//hf
-#else
         ha = hdpath(1:len_trim(hdpath))//'/'//hf
-#endif
       endif
       inquire( file=ha,  exist=file_exists )
       if( .not.file_exists ) then
         if( len_trim(hdpth)>0 ) then
-#ifdef PCDOS
-          ha = hdpth(1:len_trim(hdpth))//'\'//hf
-#else
           ha = hdpth(1:len_trim(hdpth))//'/'//hf
-#endif
         endif
         inquire( file=ha,  exist=file_exists )
         if( .not.file_exists ) then
@@ -379,12 +372,12 @@

   ! Print the total storage used and the expunge limits.
   write(iuo,'(/ "  total",i12)') lxs
-  if( emx(1)<huge ) then
+  if( emx(1)<huge_float ) then
     write(iuo,430) emx(1)
 430 format(/  " any neutrons with energy greater than emax =",1pe12.5,&
     &   " from the source or from a collision will be resampled.")
   endif
-  if( (emx(1)<huge .or. ecf(1)/=0.) .and. mcal==0 ) then
+  if( (emx(1)<huge_float .or. ecf(1)/=0.) .and. mcal==0 ) then
     write(iuo,440) ecf(1),emx(1)
 440 format(/  " neutron cross sections outside the range from",&
     & 1pe11.4, " to",e11.4,  " mev are expunged.")
@@ -403,7 +396,7 @@
     call erprnt(1,3,1,ih,0,0,0,0,&
     & 'i4,  " cross sections modified by free gas thermal treatment."')
   endif
-  if( kpt(2)/=0 .and. et/=huge )  emx(2) = exp(et)
+  if( kpt(2)/=0 .and. et/=huge_float )  emx(2) = exp(et)
   if( kpt(3)/=0 .or. (kpt(2)/=0 .and. ides==0) )  emx(2) = min(emx(2),emx(3))
   if( abs(emx(2)-emx(3))<.00001*emx(3) )  emx(2) = emx(3)
   if( emx(2)<emx(3) ) then
diff  -Naurd MCNP5/Source/src/ginst.F90 MCNP5_new/Source/src/ginst.F90
--- MCNP5/Source/src/ginst.F90       2003-04-30 20:11:04.000000000 -0600
+++ MCNP5_new/Source/src/ginst.F90   2004-07-22 15:14:41.000000000 -0600
@@ -1,7 +1,7 @@
-!+ $Id: ginst.F90,v 1.3 2002/12/03 19:21:47 ljcox Exp $
+!+ $Id: ginst.F90,v 1.2 2004/04/06 16:01:09 jgoorley Exp $
 ! Copyright LANL/UC/DOE - see file COPYRIGHT_INFO

-#ifdef GKSSIM
+#if defined(PLOT) || defined(MCPLOT)
```

```
 subroutine ginst(nw,nd,l1,h1,ip,x1,x2,y1,y2,lb,ld,hd)
   ! perform gks function 'initialise string'.
   ! nw = workstation identifier
@@ -27,4 +27,4 @@
 #endif /*def.xlib*/
   return
 end subroutine ginst
-#endif /*def.gkssim*/
+#endif
diff  -Naurd MCNP5/Source/src/gkssim.F90 MCNP5_new/Source/src/gkssim.F90
--- MCNP5/Source/src/gkssim.F90       2003-04-30 20:11:04.000000000 -0600
+++ MCNP5_new/Source/src/gkssim.F90   2004-07-22 15:14:41.000000000 -0600
@@ -3,14 +3,11 @@

 module gkssim
   ! Description:
-  ! Low level graphics routines. Access QuickWin, Winteracter or Xlib.
+  ! Low level graphics routines using Xlib.
   ! Postscript is also written if requested.
   ! The original genesis of these routines seems to have been written to
   ! be used on systems without GKS graphics.

-  ! Note: In LAHEY mode, this module depends on gxsub:gxon. But gxsub also
-  !       depends on gkssim. A quandry.
-
   ! Modules used:
   use mcnp_plot
   use mcnp_debug
@@ -22,7 +19,7 @@
     &       gsplci, gstxci, gqcf

 contains
-#ifdef GKSSIM
+#if defined(PLOT) || defined(MCPLOT)

   !-----------------------------------------------------------------------------------

@@ -82,53 +79,6 @@

   subroutine gacwk(nw)
     ! perform gks function 'activate workstation'.
-#ifdef QWIN
-    use dflib
-
-    character h*20
-    logical*4 st
-    type (qwinfo) wi,wf
-    type (windowconfig) wc
-
-    if( nw/=1 )  go to 10
-
-    ! move the execute window to the bottom of the frame.
-    ! this assumes a font size of 8x16 (wf has units of pixels).
-    k = getwsizeqq(qwin$framewindow,qwin$sizecurr,wf)
-    wi.type = qwin$set
-    wi.x = 0
-    wi.y = max(0,int(wf.h/16.)-32)
-    wi.h = 30
-    wi.w = 80
-    k = setwsizeqq(0,wi)
-
-    ! open the plot window and set its size (height=3/4 of width).
-    ! may not get this exact shape - use viewport to rectify.
-    open(99,file='user')
-    k = setactiveqq(99)
-    wc.numypixels = max(480,min(int(wf.h*0.8),int(wf.w*3./4.)))
-    wc.numxpixels = int(4./3.*wc.numypixels)
-    wc.numtextcols = -1
-    wc.numtextrows = -1
-    wc.numcolors = -1
-    wc.fontsize = -1
-    wc.title="MCNP Plot Window"C
-    st = setwindowconfig(wc)
-    IF(.not.st )  st = setwindowconfig(wc)
-    st = getwindowconfig(wc)
-
-    ! set font width based on 32 characters in 1/4 of window.
-    ! if font not found, use a 5x10 font.
-    k = initializefonts()
-    write(h,'(a,i1)')'bfw',min(8,int(wc.numxpixels/4./32.))
-    k = setfont(h)
-    IF(k < 0)k=setfont('bw5')
-
-    ! once window is maximized, force it to stay that way.
-    wi.type = qwin$max
-    k = setwsizeqq(99,wi)
-10  continue
-#endif /*def.qwin*/

 #ifdef XLIB
     if( nw==1 )  call xgacwk()
@@ -150,10 +100,6 @@
     if( nw==1 )  call xgdawk()
 #endif /*def.xlib*/

-#ifdef QWIN
-    if( nw==1 )  close(99)
-#endif /*def.qwin*/
-
     jta(nw) = 0

     return
@@ -164,15 +110,6 @@
   subroutine gclrwk(nw,k)
     ! perform gks function 'clear workstation'.

-#ifdef QWIN
-    use dflib
-
-    if( nw==1 ) then
-      j = setactiveqq(99)
-      call clearscreen($gclearscreen)
-    endif
-#endif /*def.qwin*/
-
 #ifdef XLIB
     if( nw==1) call xgclrw()
 #endif /*def.xlib*/
@@ -184,11 +121,6 @@

   subroutine guwk(nw,k)
     ! perform gks function 'update workstation'.
-#ifdef QWIN
-    use dflib
-
-    if( nw==1 .and. k==0 )  j = focusqq(99)
-#endif /*def.qwin*/

 #ifdef XLIB
     if( nw==1 .and. k==0 )  call xguwk()
@@ -203,14 +135,7 @@

   subroutine gpl(n,x,y)
     ! perform gks function 'polyline'.
-#ifdef LAHEY
-    use WINTERACTER
-#endif /*def.lahey*/
-#ifdef QWIN
-    use dflib
-
-    type (wxycoord) xy
-#endif /*def.qwin*/
```

```
+
     real :: cr(4,3),x(n),y(n),xw(20),yw(20)
     data cr/.006,.004,.006,.004,4*.003,.009,3*.003/

@@ -242,20 +167,6 @@
       call xgpl(x(i-1),y(i-1),x(i),y(i))
     enddo
 #endif /*def.xlib*/
-#ifdef QWIN
-    call moveto_w(dble(x(1)),dble(y(1)),xy)
-    do i = 2,n
-      k = lineto_w(dble(x(i)),dble(y(i)))
-    enddo
-#endif /*def.qwin*/
-#ifdef LAHEY
-    call IGrMoveto(11.*(x(1)-xlf)/(xrt-xlf)*.75/chite(4),&
-      & 8.5*(y(1)-ybt)/(ytp-ybt))
-    do i = 2,n
-      call IGrLineTo(11.*(x(i)-xlf)/(xrt-xlf)*.75/chite(4),&
-        & 8.5*(y(i)-ybt)/(ytp-ybt))
-    enddo
-#endif /*def.lahey*/
     return

     ! draw interrupted line.
@@ -292,20 +203,6 @@
         call xgpl(xw(i-1),yw(i-1),xw(i),yw(i))
       enddo
 #endif /*def.xlib*/
-#ifdef QWIN
-      call moveto_w(dble(xw(1)),dble(yw(1)),xy)
-      do i = 2,m
-        k = lineto_w(dble(xw(i)),dble(yw(i)))
-      enddo
-#endif /*def.qwin*/
-#ifdef LAHEY
-      call IGrMoveTo(11.*(xw(1)-xlf)/(xrt-xlf)*.75/chite(4),&
-        & 8.5*(yw(1)-ybt)/(ytp-ybt))
-      do i = 2,m
-        call IGrLineTo(11.*(xw(i)-xlf)/(xrt-xlf)*.75/chite(4),&
-          & 8.5*(yw(i)-ybt)/(ytp-ybt))
-      enddo
-#endif /*def.lahey*/
 50    continue
       xw(1) = xw(m)
       yw(1) = yw(m)
@@ -327,13 +224,6 @@
   subroutine gtx(x,y,h)
     ! perform gks function 'text'.

-#ifdef LAHEY
-    use WINTERACTER
-#endif /*def.lahey*/
-#ifdef QWIN
-    use dflib
-    type (wxycoord) xy
-#endif /*def.qwin*/
     character(len=*) :: h
     real             :: x,y
     character(len=120) :: hn
@@ -377,28 +267,6 @@
 #ifdef XLIB
     call xgtx(h,n,x,y)
 #endif /*def.xlib*/
-#ifdef QWIN
-
-    ! top left of character string is the origin.
-    ! font size is not set to chite(5) in gschup, but this works.
-    x1 = x
-    y1 = y
-    if( abs(chup(1))>abs(chup(2)))  x1 = x1-2.*chite(5)
-    IF(abs(chup(1))<=abs(chup(2)))  y1 = y1+chite(5)
-    call moveto_w(dble(x1),dble(y1),xy)
-    call outgtext(h(1:n))
-#endif /*def.qwin*/
-#ifdef LAHEY
-    m = 1
-    if( abs(chup(1))>abs(chup(2)))  m = n
-    ic = nint((x-xlf)/(xrt-xlf)*.75/chite(4)*10000)
-    if( abs(chup(1))>abs(chup(2)))  ic = max(0,ic-1)
-    il = max(0,nint((((ytp-y)/(ytp-ybt))-m+1)*10000))
-    do i = 1,m
-      call WindowOutString(ic,il,h(i:i+n-m))
-      il = il+1
-    enddo
-#endif /*def.lahey*/
     return
   end subroutine gtx

@@ -439,9 +307,6 @@

   subroutine gschup(x,y)
     ! perform gks function 'set character up vector'.
-#ifdef QWIN
-    use dflib
-#endif /*def.qwin*/

     chup(1) = x
     chup(2) = y
@@ -451,15 +316,6 @@
     call xgschu(x,y)
 #endif /*def.xlib*/

-#ifdef QWIN
-    ! tenths of degrees for counterclockwise rotation.
-    if( x==0. .and. y>0.)  k = 0
-    if( x==0. .and. y<0.)  k = 1800
-    if( x<0. .and. y==0.)  k = 900
-    if( x>0. .and. y==0.)  k = -900
-    if( x/=0. .and. y/=0.) k = nint(1800./3.14159265*atan2(x,y))-900
-    call setgtextrotation(k)
-#endif /*def.qwin*/
     return
   end subroutine gschup

@@ -467,9 +323,6 @@

   subroutine gswn(nt,xm,xx,ym,yx)
     ! perform gks function 'set window'.
-#ifdef QWIN
-    use dflib
-#endif /*def.qwin*/

     integer :: nt
     real :: xm,xx,ym,yx
@@ -486,10 +339,6 @@
     call xgswn(xm,xx,ym,yx)
 #endif /*def.xlib*/

-#ifdef QWIN
-    k = setwindow(.true.,dble(xm),dble(ym),dble(xx),dble(yx))
-#endif /*def.qwin*/
-
     return
   end subroutine gswn

@@ -497,24 +346,13 @@

   subroutine gsvp(nt,xm,xx,ym,yx)
     ! perform gks function 'set viewport'.
-#ifdef QWIN
-    use dflib
-    logical*4 st
-    type (windowconfig) wc
-#endif /*def.qwin*/
     integer :: nt
     real    :: xm,xx,ym,yx

     chite(3) = xx-xm
     chite(4) = yx-ym

-#ifdef QWIN
-    st = getwindowconfig(wc)
-    call setviewport(int2(xm),int2(ym),min(int2(wc.numxpixels),&
-      & int2(wc.numypixels/yx)),int2(wc.numypixels))
-#endif /*def.qwin*/
-
     return
   end subroutine gsvp

@@ -550,9 +388,6 @@
   !------------------------------------------------------------------------------

   subroutine grqlc(nw,j,is,i,x,y)
-#ifdef LAHEY
-    use WINTERACTER
-#endif /*def.lahey*/
     ! perform gks function 'request locator'.

     integer :: nw,j,is,i
@@ -564,47 +399,6 @@
     call xgrqlc(j,x,y)
 #endif /*def.xlib*/

-#ifdef LAHEY
-    type (win_message) ::mesg
-    logical click
-    do i = 1,10
-      call WMessageEnable(i,0)
-    enddo
-    call WindowOutString(100,9500,'Click for Cursor Location')
-    call WindowOutString(100,9700,'Click on Title Bar to Redraw')
-    call WindowOutString(100,9900,'Drag on Title Bar to Move Window')
-    call WMessageEnable(MouseButDown,1)
-    call WMessageEnable(Expose,1)
-    click = .false.
-    do while (.not.click)
-      call WmessagePeek(itype,mesg)
-      select case(itype)
-      case(Expose)
-        call gxon(1)
-        call tekdvr
-        call WindowOutString(100,9500,'Click for Cursor Location')
-        call WindowOutString(100,9700,'Click on Title Bar to Redraw')
-        call WindowOutString(100,9900,'Drag on Title Bar to Move Window')
-      case(mousebutdown)
-        click = .true.
-      end select
-    enddo
-    call WMessageEnable(MouseButDown,0)
-    call WMessageEnable(Expose,0)
-    call gxon(1)
-    call tekdvr
-    call WindowClearArea(0,9700,2000,9990)
-    x = mesg%gx * (xrt-xlf)/11.0/0.75*chite(4)  + xlf
-    y = mesg%gy * (ytp-ybt)/8.5   + ybt
-    !  draw cross hairs.
-    do i = -1,1,2
-      call IGrColourN(40)
-      call IGrMoveTo(mesg%gx,mesg%gy)
-      call IGrLineTo(mesg%gx+i*0.1,mesg%gy)
-      call IGrMoveTo(mesg%gx,mesg%gy)
-      call IGrLineTo(mesg%gx,mesg%gy+i*0.1)
-    enddo
-#endif /*def.lahey*/
     return
   end subroutine grqlc

@@ -613,13 +407,6 @@
   subroutine gfa(n,x,y)
     !  perform gks function 'fill area'.

-#ifdef LAHEY
-    use WINTERACTER
-#endif /*def.lahey*/
-#ifdef QWIN
-    use dflib
-#endif /*def.qwin*/
-
     integer :: n
     real    :: x(n),y(n)
     integer :: nx(10),ny(10)
@@ -642,20 +429,6 @@
     call xgfa(x(1),y(1),x(3),y(3))
 #endif /*def.xlib*/

-#ifdef QWIN
-    k = rectangle_w($gfillinterior,dble(x(1)),dble(y(1)),&
-      & dble(x(3)),dble(y(3)))
-#endif /*def.qwin*/
-
-#ifdef LAHEY
-    do i = 1,n
-      xn(i) = 11.*(x(i)-xlf)/(xrt-xlf)*.75/chite(4)
-      yn(i) = 8.5*(y(i)-ybt)/(ytp-ybt)
-    enddo
-    call IGrFillPattern(4,2,3)
-    call IGrPolygonComplex(xn,yn,n)
-#endif /*def.lahey*/
-
     return
   end subroutine gfa

@@ -663,14 +436,6 @@

   subroutine gsfaci(ic,md)
     !  perform gks function 'set fill area color index'.
-    !*  need to check on allowed color numbering schemes for LAHEY
-
-#ifdef LAHEY
-    use WINTERACTER
-#endif /*def.lahey*/
-#ifdef QWIN
-    use dflib
-#endif /*def.qwin*/

     integer :: ic ! color index.
     integer :: md ! mode: 0 = use colors; 1 = use shades.
@@ -689,7 +454,6 @@
     case default ! use contrasted shades

       c = colors(ic)
-
     end select


@@ -703,16 +467,6 @@
     call xgsplc(c%rgb)
 #endif /*def.xlib*/

-#ifdef QWIN
-    ! Set QuickWin graphics foreground RGB color.
-    ik = c%rgb(1) + 256*c%rgb(2) + 256**2*c%rgb(3)
-    k = setcolorrgb(ik)
-#endif /*def.qwin*/
-
-#ifdef LAHEY
-    call IGrColourN(16*(ic-1))
-#endif /*def.lahey*/
-
     return
   end subroutine gsfaci

@@ -720,12 +474,6 @@

   subroutine gsplci(ic,md)
     !  perform gks function 'set polyline color index'.
-#ifdef LAHEY
-    use WINTERACTER
-#endif /*def.lahey*/
-#ifdef QWIN
-    use dflib
-#endif /*def.qwin*/

     integer :: ic ! index of color to set.
     integer :: md ! mode: 0 = use colors; 1 = use shades
@@ -756,16 +504,6 @@
     call xgsplc(c%rgb)
 #endif /*def.xlib*/

-#ifdef QWIN
-    ! Set QuickWin graphics foreground RGB color.
-    ik = c%rgb(1) + 256*c%rgb(2) + 256**2*c%rgb(3)
-    k = setcolorrgb(ik)
-#endif /*def.qwin*/
-
-#ifdef LAHEY
-    call IGrColourN(16*(ic-1))
-#endif /*def.lahey*/
-
     return
   end subroutine gsplci

@@ -774,13 +512,6 @@
   subroutine gstxci(ic,md)
     !  perform gks function 'set text color index'.

-#ifdef QWIN
-    use dflib
-#endif /*def.qwin*/
-#ifdef LAHEY
-    use WINTERACTER
-    type (win_font) :: MCNPfont
-#endif /*def.lahey*/
     integer :: ic ! Index of color to set.
     integer :: md ! mode: 0 = use colors; 1 = use shades.
     type(color) :: c
@@ -807,21 +538,6 @@
     call xgsplc(c%rgb)
 #endif /*def.xlib*/

-#ifdef QWIN
-    ! Set QuickWin graphics foreground RGB color.
-    ik = c%rgb(1) + 256*c%rgb(2) + 256**2*c%rgb(3)
-    k = setcolorrgb(ik)
-#endif /*def.qwin*/
-
-#ifdef LAHEY
-    MCNPfont%ifontnum = 0
-    MCNPfont%ifcol = (ic-1)/2
-    MCNPfont%ibcol = -1
-    MCNPfont%iwidth = 100
-    MCNPfont%iheight = 200
-    MCNPfont%ibold = 1
-    call WindowFont(MCNPfont)
-#endif /*def.lahey*/
     return
   end subroutine gstxci

@@ -830,18 +546,6 @@
   subroutine gqcf(ic,ni)
     !  perform gks function 'inquire color facilities'.

-#ifdef QWIN
-    use dflib
-    type (windowconfig) wc
-#endif /*def.qwin*/
-#ifdef LAHEY
-    use WINTERACTER
-    type (win_style) :: MCNPwin
-    type (win_font) :: MCNPfont
-    real(dknd) :: ra(7)
-    integer    :: ia(9)
-    character hl*40
-#endif /*def.lahey*/
     integer :: ic ! color flag.
     integer :: ni ! number of colors.

@@ -853,35 +557,6 @@
     if( ic/=0) ni = ncolor
 #endif /*def.xlib*/

-#ifdef QWIN
-    k  = getwindowconfig(wc)
-    ni = min(ncolor,wc.numcolors)
-    if( ni>2) ic = 1
-#endif /*def.qwin*/
-
-#ifdef LAHEY
-    call WInitialise(' ')
-    MCNPwin%flags = MinButton + MaxButton
-    MCNPwin%x = 0
-    MCNPwin%y = 0
-    MCNPwin%width = 0
-    MCNPwin%height = 0
-    MCNPwin%menuid = 0
-    MCNPwin%title = "MCNP Plot Window"
-    call WindowOpen(MCNPwin)
-    call IGrArea(0.0,0.0,1.0,1.0)
-    call IGrUnits(0.0,0.0,11.0,8.5)
-    ni = min(ncolor,InfoGrScreen(30))
-    MCNPfont%ifontnum = 4
-    MCNPfont%iwidth = 100
-    MCNPfont%iheight = 200
-    MCNPfont%ibold = 1
-    call WindowFont(MCNPfont)
-    nctext = 10000/WInfoFont(9)
-    nltext = 10000/WInfoFont(10)
-    call WindowClose
-    if( ni>2 )  ic = 1
-#endif /*def.lahey*/
     return
   end subroutine gqcf

@@ -956,5 +631,5 @@
     return
   end subroutine gqcf

-#endif /*def.gkssim*/
+#endif /*def PLOT or MCPLOT */
 end module gkssim
diff  -Naurd MCNP5/Source/src/gmgww.F90 MCNP5_new/Source/src/gmgww.F90
--- MCNP5/Source/src/gmgww.F90       2003-04-30 20:11:04.000000000 -0600
+++ MCNP5_new/Source/src/gmgww.F90   2004-07-22 15:14:41.000000000 -0600
@@ -69,7 +69,7 @@
```

```
     if( rim/=0. )  r=log10(rim)
     do jg=1,jgm(ip)
       tm = 0.
-      ts = huge
+      ts = huge_float
       do ia=1,mxa
         if( fim(ip,ia)==0. )  cycle
         t = scr(mxa*(jg-1)+ia)
diff  -Naurd MCNP5/Source/src/gxsub.F90 MCNP5_new/Source/src/gxsub.F90
--- MCNP5/Source/src/gxsub.F90        2003-04-30 20:11:04.000000000 -0600
+++ MCNP5_new/Source/src/gxsub.F90    2004-07-22 15:14:41.000000000 -0600
@@ -61,10 +61,6 @@

     ! Modules used:
     use mcnp_plot
-#ifdef LAHEY
-    use WINTERACTER
-    type (win_style) :: MCNPwin
-#endif
     integer :: nw
     integer :: kn(2),ic,ni,i,jt,ie,nc
     character(len=12) :: hf
@@ -98,11 +94,6 @@
           if( ic/=0 .and. ni>2 ) then
             mcolor = sign(min(ni,ncolor),mcolor)
             do i = 1,ncolor+7
-#if defined(LAHEY) || defined(QWIN)
-              if( i<=ncolor )  kcolor(i) = i
-              if( i<=7 )  kcolor(ncolor+i) = i
-              if( ni<ncolor+7 )  cycle
-#endif
               kcolor(i) = abs(kci(i))
               colors(i)%rgb(1) = nint(255*colors(i)%r)
               colors(i)%rgb(2) = nint(255*colors(i)%g)
@@ -124,21 +115,6 @@
     call gsvp(1,0.,1.,0.,.75+jvp*.25)
     wnvp(3) = 1.
     wnvp(4) = .75+jvp*.25
-#ifdef LAHEY
-    if( nw==1 ) then
-      call WInitialise(' ')
-      MCNPwin%flags = MinButton + MaxButton
-      MCNPwin%x = 0
-      MCNPwin%y = 0
-      MCNPwin%width = 0
-      MCNPwin%height = 0
-      MCNPwin%menuid = 0
-      MCNPwin%title = "MCNP Plot Window"
-      call WindowOpen(MCNPwin)
-      call IGrArea(0.0,0.0,1.0,1.0)
-      call IGrUnits(0.0,0.0,11.0,8.5)
-    endif
-#endif
     return
   end subroutine gxon

@@ -162,34 +138,12 @@
     ! Description:
     ! Return home, update the display, and be ready for fortran i/o.

-    ! Modules used
-#ifdef LAHEY
-    use WINTERACTER
-#endif
-
     real :: s,t
-#ifdef LAHEY
-    integer           :: ia(9)
-    real              :: ra(7), t1, t2
-    character(len=40) :: hl
-#endif
```

```
     if( jgxa(1)==0 )  return
     call gschh(.01*real(wnvp(2)))

-#ifndef LAHEY
     call gtx(s,t,'.')
-
-#else
-    !* seems to be a 4 sec pause for LAHEY only.
-    call secnd(t1)
-    if( ifile/=itty ) then
-      do
-        call secnd(t2)
-        if( t2-t1>=4.d0 )  exit
-      enddo
-    endif
-#endif
     call guwk(1,0)
     return
   end subroutine gxhome
@@ -455,3 +409,4 @@
   end subroutine gxaxis
 #endif
 end module gxsub
+
diff  -Naurd MCNP5/Source/src/hpsort.F90 MCNP5_new/Source/src/hpsort.F90
--- MCNP5/Source/src/hpsort.F90      2003-04-30 20:11:06.000000000 -0600
+++ MCNP5_new/Source/src/hpsort.F90  2004-07-22 15:14:41.000000000 -0600
@@ -1,84 +1,181 @@
-!+ $Id: hpsort.F90,v 1.4 2002/12/03 19:22:02 ljcox Exp $
+!+ $Id: hpsort.F90,v 1.2 2004/04/01 23:48:02 jgoorley Exp $
 ! Copyright LANL/UC/DOE - see file COPYRIGHT_INFO

-subroutine hpsort(it,kn,ks,np,ns)
-  ! sort (heapsort) the np extreme tallies into ascending order.
-  ! it is the tally number, kn and ks are offsets in the nhsd and
-  ! shsd arrays, np is the number of extreme tallies in the shsd
-  ! array, and ns is the number of big,small tallies in array stt.
-  use mcnp_global
-  use mcnp_debug
+module hpsort_mod

-  implicit real(dknd) (a-h,o-z)
+  interface hpsort
+    ! ==> specific routines used for generic subroutine:
+    module procedure hpsort_i4, hpsort_i8
+  end interface

-  real(dknd) :: ar(2*ntp)
-
-  ! fill array ar with the history tally points to be sorted.
-  do j=1,ns
-    ar(j) = stt(kstt+j,it)
-  end do
-  do j=1,np
-    ar(ns+j) = shsd(ks+j,it)
-  end do
-  nh = ns+np
-  ir = nh
-  l  = ir/2+1
-
-  ! perform the sort using the heapsort (nlogn) algorithm.
-40 continue
-  if( l<=1 )  go to 90
-  l = l-1
-  a = ar(l)
-50 continue
-  ii = l
-  j  = l+l
-  if( j>ir ) then
-    ar(ii) = a
-    go to 40
-  elseif( j==ir ) then
-    go to 70
-  else
-    go to 60
-  endif
-60 continue
-  if( ar(j)<ar(j+1) )  j=j+1
-70 continue
-  if( a>=ar(j) )  go to 80
-  ar(ii) = ar(j)
-  ii = j
-  j  = j+j
-  if( j>ir ) then
-    ar(ii) = a
-    go to 40
-  elseif( j==ir ) then
-    go to 70
-  else
-    go to 60
-  endif
-80 continue
-  j = ir+1
-  if( j>ir ) then
-    ar(ii) = a
-    go to 40
-  elseif( j==ir ) then
-    go to 70
-  else
-    go to 60
-  endif
-90 continue
-  a = ar(ir)
-  ar(ir) = ar(1)
-  ir = ir-1
-  if( ir/=1 )  go to 50
-  ar(1) = a
-
-  ! fill shsd(kshs+nsp+6,it) with the ntp largest,smallest values.
-  nt = min(ntp,nh)
-  do j=1,nt
-    shsd(ks+j,it) = ar(nh-nt+j)
-  enddo
-  nhsd(kn+1,it) = nt
-  nhsd(kn+2,it) = 0
-  if( nt==ntp )  shsd(ks+ntp+1,it)=shsd(ks+1,it)
-  return
-end subroutine hpsort
+  contains
+
+  subroutine hpsort_i4(it,kn,ks,np,ns)
+    ! sort (heapsort) the np extreme tallies into ascending order.
+    ! it is the tally number, kn and ks are offsets in the nhsd and
+    ! shsd arrays, np is the number of extreme tallies in the shsd
+    ! array, and ns is the number of big,small tallies in array stt.
+    use mcnp_global
+    use mcnp_debug
+
+    implicit real(dknd) (a-h,o-z)
+
+    integer        :: it,kn,ks
+    integer(i4knd) :: np,ns
+    real(dknd)     :: ar(2*ntp)
+
+    ! fill array ar with the history tally points to be sorted.
+    do j=1,ns
+      ar(j) = stt(kstt+j,it)
+    end do
+    do j=1,np
+      ar(ns+j) = shsd(ks+j,it)
+    end do
+    nh = ns+np
+    ir = nh
+    l  = ir/2+1
+
+    ! perform the sort using the heapsort (nlogn) algorithm.
+40  continue
+    if( l<=1 )  go to 90
+    l = l-1
+    a = ar(l)
+50  continue
+    ii = l
+    j  = l+l
+    if( j>ir ) then
+      ar(ii) = a
+      go to 40
+    elseif( j==ir ) then
+      go to 70
+    else
+      go to 60
+    endif
+60  continue
+    if( ar(j)<ar(j+1) )  j=j+1
+70  continue
+    if( a>=ar(j) )  go to 80
+    ar(ii) = ar(j)
+    ii = j
+    j  = j+j
+    if( j>ir ) then
+      ar(ii) = a
+      go to 40
+    elseif( j==ir ) then
+      go to 70
+    else
+      go to 60
+    endif
+80  continue
+    j = ir+1
+    if( j>ir ) then
+      ar(ii) = a
+      go to 40
+    elseif( j==ir ) then
+      go to 70
+    else
+      go to 60
+    endif
+90  continue
+    a = ar(ir)
+    ar(ir) = ar(1)
+    ir = ir-1
+    if( ir/=1 )  go to 50
+    ar(1) = a
+
+    ! fill shsd(kshs+nsp+6,it) with the ntp largest,smallest values.
+    nt = min(ntp,nh)
+    do j=1,nt
+      shsd(ks+j,it) = ar(nh-nt+j)
+    enddo
+    nhsd(kn+1,it) = int(nt,i8knd)
+    nhsd(kn+2,it) = 0_i8knd
+    if( nt==ntp )  shsd(ks+ntp+1,it)=shsd(ks+1,it)
+    return
+  end subroutine hpsort_i4
+
+  subroutine hpsort_i8(it,kn,ks,np,ns)
+    ! sort (heapsort) the np extreme tallies into ascending order.
+    ! it is the tally number, kn and ks are offsets in the nhsd and
+    ! shsd arrays, np is the number of extreme tallies in the shsd
+    ! array, and ns is the number of big,small tallies in array stt.
+    use mcnp_global
+    use mcnp_debug
+
+    implicit real(dknd) (a-h,o-z)
+
+    integer        :: it, kn, ks
+    integer(i8knd) :: np, ns
+    real(dknd)     :: ar(2*ntp)
+
+    ! fill array ar with the history tally points to be sorted.
+    do j=1,ns
+      ar(j) = stt(kstt+j,it)
+    end do
+    do j=1,np
+      ar(ns+j) = shsd(ks+j,it)
+    end do
+    nh = ns+np
+    ir = nh
+    l  = ir/2+1
+
+    ! perform the sort using the heapsort (nlogn) algorithm.
+40  continue
+    if( l<=1 )  go to 90
+    l = l-1
+    a = ar(l)
+50  continue
+    ii = l
+    j  = l+l
+    if( j>ir ) then
+      ar(ii) = a
+      go to 40
+    elseif( j==ir ) then
+      go to 70
+    else
+      go to 60
+    endif
+60  continue
+    if( ar(j)<ar(j+1) )  j=j+1
+70  continue
+    if( a>=ar(j) )  go to 80
+    ar(ii) = ar(j)
+    ii = j
+    j  = j+j
+    if( j>ir ) then
+      ar(ii) = a
+      go to 40
+    elseif( j==ir ) then
+      go to 70
+    else
+      go to 60
+    endif
+80  continue
+    j = ir+1
+    if( j>ir ) then
+      ar(ii) = a
+      go to 40
+    elseif( j==ir ) then
+      go to 70
+    else
+      go to 60
+    endif
+90  continue
+    a = ar(ir)
+    ar(ir) = ar(1)
+    ir = ir-1
+    if( ir/=1 )  go to 50
+    ar(1) = a
+
+    ! fill shsd(kshs+nsp+6,it) with the ntp largest,smallest values.
+    nt = min(ntp,nh)
+    do j=1,nt
+      shsd(ks+j,it) = ar(nh-nt+j)
+    enddo
+    nhsd(kn+1,it) = int(nt,i8knd)
+    nhsd(kn+2,it) = 0_i8knd
+    if( nt==ntp )  shsd(ks+ntp+1,it)=shsd(ks+1,it)
+    return
+  end subroutine hpsort_i8
+
+end module hpsort_mod
diff  -Naurd MCNP5/Source/src/hstory.F90 MCNP5_new/Source/src/hstory.F90
--- MCNP5/Source/src/hstory.F90      2003-04-30 20:11:06.000000000 -0600
+++ MCNP5_new/Source/src/hstory.F90  2004-07-22 15:14:41.000000000 -0600
@@ -8,6 +8,7 @@
   ! Modules:
   use smmp, only: sm_lon, sm_loff
   use mcnp_global
+  use dxtran_mod
   use mcnp_debug
   use rmc_mod
   use ra1_mod
@@ -20,9 +21,9 @@

   ! Debug features:  Set up event log.  Print debug line.
   krflg = 0
-  if( npstc>=dbcn(3) .and. npstc<=dbcn(4) )  krflg = 1
+  if( npstc>=int(dbcn(3),i8knd) .and. npstc<=int(dbcn(4),i8knd) )  krflg = 1
   if( dbcn(2)/=0. ) then
-    if( mod(npstc,int(dbcn(2)))==0 ) then
+    if( mod(npstc,int(dbcn(2),i8knd))==0 ) then
       call RN_query( first = i8_first, nps = int(nps,i8knd) )
       if( ltasks>1 .or. ntasks>1 .or. mct<0 ) then
         write(iuo,10) npstc, i8_first
@@ -82,26 +83,7 @@
   if( kdb/=0 )  go to 390

   ! Calculate the distance to the nearest dxtran sphere, dxl.
-  dxl = huge
-  do i = 1,ndx(ipt)
-    if( idx/=i ) then
-      if( lev==0 )  then
-        sr = (/ xxx, yyy, zzz /)
-        sd = (/ uuu, vvv, www /)
-      else
-        sr = udt(1:3,0)
-        sd = udt(4:6,0)
-      endif
-      f = dxx(ipt,1,i)-sr(1)
-      g = dxx(ipt,2,i)-sr(2)
-      h = dxx(ipt,3,i)-sr(3)
-      q = f*sd(1)+g*sd(2)+h*sd(3)
-      c = min(max(zero,q),dxl)
-      if( (f-sd(1)*c)**2+(g-sd(2)*c)**2+(h-sd(3)*c)**2<dxx(ipt,5,i) ) then
-        dxl = min(dxl,q-sqrt(max(zero,q**2+dxx(ipt,5,i)-f**2-g**2-h**2)))
-      endif
-    endif
-  enddo
+  dxl = dist_dxtran_sphere()

   ! Calculate the distance to time cutoff, dtc.
   dtc = vel*(tco(ipt)-tme)
@@ -112,7 +94,7 @@
   totm = 0.
   pfp  = 0.
   stp  = 0.
-  deb  = huge
+  deb  = huge_float
   if( mk/=0  ) then
     if(     ipt==1 ) then
       call acetot(0,mk)
@@ -132,9 +114,10 @@

   ! Calculate the mean free path, gs, and its reciprocal, qpl.
   gs  = 0.
-  pmf = huge
-  dw  = huge
+  pmf = huge_float
+  dw  = huge_float
   qpl = (totm+pfp)*rh
+
   if( ipt==1 )  qpl = qpl + abs(ra_qpl())
   ple = qpl
   if( ple/=0. ) then
@@ -157,7 +140,7 @@
         call forcol
         if( kdb/=0  )  go to 390
         if( nter/=0 )  go to 260
-        dw = huge
+        dw = huge_float
       endif
     endif
   endif
@@ -262,6 +245,7 @@
     udt(1:3,l) = udt(1:3,l)+udt(4:6,l)*d
   enddo

+
   ! Special treatment for multigroup electrons.
   if( stp/=0. ) then
     t1 = d*stp
@@ -487,7 +471,7 @@
       rh   = rho(icl)
       mk   = mat(icl)
       totm = 0.
-      deb  = huge
+      deb  = huge_float
       if( mk/=0 ) then
         if(     ipt==1 ) then
           call acetot(0,mk)
@@ -495,8 +479,8 @@
           call photot(mk)
         endif
       endif
-      pmf  = huge
-      dw   = huge
+      pmf  = huge_float
+      dw   = huge_float
       qpl  = totm*rh
       if( ipt==1 )  qpl = qpl + abs(ra_qpl())
       ple  = qpl
diff  -Naurd MCNP5/Source/src/igeom.F90 MCNP5_new/Source/src/igeom.F90
--- MCNP5/Source/src/igeom.F90        2003-04-30 20:11:06.000000000 -0600
+++ MCNP5_new/Source/src/igeom.F90    2004-07-22 15:14:41.000000000 -0600
@@ -6,6 +6,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/imcn.F90 MCNP5_new/Source/src/imcn.F90
--- MCNP5/Source/src/imcn.F90 2003-04-30 20:11:08.000000000 -0600
+++ MCNP5_new/Source/src/imcn.F90     2004-07-22 15:14:41.000000000 -0600
@@ -14,6 +14,10 @@
   use fmesh_mod, only: nmesh, fmesh_initialize

   implicit real(dknd) (a-h,o-z)
+
+  integer :: i, ip, it, iu, j, k, la, m, mt, n, ne, nl, nn
+  integer(i8knd) :: n_i8knd
+
   character(len=8) :: hs
   real(dknd), parameter :: eb(16) = &
     &  (/ 20.d0, 15.d0, 10.d0, 9.d0, 8.d0,  7.d0,  6.d0,   5.d0, &
@@ -141,26 +145,26 @@
```

```
     ! Initialize general common not yet done.
-    ddg( 1:2, 1:mxdt ) = huge
-    ddx( 1:mipt, 1:2, 1:mxdx ) = huge
+    ddg( 1:2, 1:mxdt ) = huge_float
+    ddx( 1:mipt, 1:2, 1:mxdx ) = huge_float
     dbcn(10) = dftint
     dmp     = dfdmp
     dnb     = -1.
     ecf(2)  = .001d0
     efac    = sqrt(sqrt(sqrt(.5+zero)))
     emcf( 2:mipt) = 100.
-    emx(1)  = huge
-    emx(2)  = huge
+    emx(1)  = huge_float
+    emx(2)  = huge_float
     ikz     = 30
     kct     = -1
     ivdd    = (/ (i,i = 1,maxv) /)
     lost(1) = 10
     lost(2) = 10
-    npd     = 1000
+    npd     = 1000_i8knd
     ntc     = 50
     ntc1    = 50
     rkk     = 1.
-    tco( 1:mipt ) = .001d0*huge
+    tco( 1:mipt ) = .001d0*huge_float
     bnum    = 1.
     xnum    = 1.
     rnok    = 1.
@@ -168,10 +172,10 @@
     wc1(1)  = -.50d0
     wc2(1)  = -.25d0
     if( kpt(2)/=0 ) then
-      wc1(2) = huge
-      wc2(2) = huge
+      wc1(2) = huge_float
+      wc2(2) = huge_float
     endif
-    wgts(1) = huge
+    wgts(1) = huge_float
     if( wwp(1,4)>0. )  wc1(1) = 0.
     if( wwp(1,4)>0. )  wc2(1) = 0.
     if( wwp(2,4)>0. )  wc1(2) = 0.
@@ -181,7 +185,7 @@
     ! Initialize dynamically allocated common.
     jasw = 0
     do j = 1,mipt
-      if(  ndx(j)/=0 )  dxcp( :, j, : ) = huge
+      if(  ndx(j)/=0 )  dxcp( :, j, : ) = huge_float
       if( ngww(j)/=0 )  ewwg( ngww(j)+mgww(j) ) = 100.
     enddo
     if( kpt(1)*kpt(2)/=0 )  gwt(:) = -1.
@@ -196,11 +200,11 @@
     lxd( 2, 1:nmat1 ) = ichar(' ')+256*(ichar(' ')+256*ichar('p'))
     lxd( 3, 1:nmat1 ) = ichar(' ')+256*(ichar(' ')+256*ichar('e'))
     lxd( 4, 1:nmat1 ) = ichar(' ')+256*(ichar(' ')+256*ichar('u'))
-    pnt( 1:nmat1 ) = huge
+    pnt( 1:nmat1 ) = huge_float
     trf(5, 0) = 1.
     trf(9, 0) = 1.
     trf(13,0) = 1.
-    trf(5:13, 1: ) = huge
+    trf(5:13, 1: ) = huge_float
     tmp(:) = 253.d-10


@@ -453,7 +457,7 @@
     endif
     allocate( gbnk( 1:mbnk*mtasks ) )
-    allocate( ibnk( 1:(nbmx*(lpblcm+2*abs(iunr))+1)*mtasks ) )
+    allocate( ibnk( 0:(nbmx*(lpblcm+2*abs(iunr))+1)*mtasks ) )
     allocate( tal(  1:(nmxf*mxf+ktls)*mt ) )
     gbnk = 0.0
     ibnk = 0
@@ -482,9 +486,9 @@
     wcs2(1) = max( wc2(1), -wc2(1)*swtm )
     if( kf8==3 )  wc1(2) = 0.
     do i = 2,mipt
-      if( wc1(i)==huge .and. kpt(1)/=0 )  wc1(i) = wc1(1)
-      if( wc1(i)==huge .and. kpt(1)==0 )  wc1(i) = -.5d0
-      if( wc2(i)==huge )  wc2(i) = .5*wc1(i)
+      if( wc1(i)==huge_float .and. kpt(1)/=0 )  wc1(i) = wc1(1)
+      if( wc1(i)==huge_float .and. kpt(1)==0 )  wc1(i) = -.5d0
+      if( wc2(i)==huge_float )  wc2(i) = .5*wc1(i)
       wcs1(i) = max(wc1(i),-wc1(i)*swtm)
       wcs2(i) = max(wc2(i),-wc2(i)*swtm)
     enddo
@@ -557,9 +561,9 @@
       &                   new_count_max_nps = int(nrnh(3), i8knd)   )

     if( nde/=0 )  dbcn(2) = nde
-    if( nsr==6 .and. nrrs>=nrss .and. npp>=0 ) &
+    if( nsr==6 .and. nrrs>=nrss .and. npp>=0_i8knd ) &
       & call expire(0,'imcn', 'no more tracks on '//rssa//' file.')
-    if( (nfer==0 .or. lfatl/=0) .and. jovr(4)/=0 .and. npp>=0 ) then
+    if( (nfer==0 .or. lfatl/=0) .and. jovr(4)/=0 .and. npp>=0_i8knd ) then
       call tpefil(6)
     endif

@@ -567,14 +571,14 @@
       ! Set npc(nn) to zero if tfc nps entry was not scheduled.
       nn = 0
       do i = 1,20
-        if( npc(i)/=0) nn = i
+        if( npc(i)/=0_i8knd) nn = i
       enddo
       if( nn/=1 ) then
-        n = npd
-        if( nn==20 )  n = npd/2
-        if( npc(nn)/=npc(nn-1)+n )  npc(nn) = 0
+        n_i8knd = npd
+        if( nn==20 )  n_i8knd = npd/2
+        if( npc(nn)/=npc(nn-1)+n_i8knd )  npc(nn) = 0_i8knd
       else
-        if( npc(1)/=npd )  npc(1) = 0
+        if( npc(1)/=npd )  npc(1) = 0_i8knd
       endif
     endif

diff  -Naurd MCNP5/Source/src/inpert.F90 MCNP5_new/Source/src/inpert.F90
--- MCNP5/Source/src/inpert.F90      2003-04-30 20:11:08.000000000 -0600
+++ MCNP5_new/Source/src/inpert.F90  2004-07-22 15:14:41.000000000 -0600
@@ -8,6 +8,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/ipbc.F90 MCNP5_new/Source/src/ipbc.F90
--- MCNP5/Source/src/ipbc.F90 2003-04-30 20:11:10.000000000 -0600
+++ MCNP5_new/Source/src/ipbc.F90    2004-07-22 15:14:41.000000000 -0600
@@ -7,6 +7,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod
```

```
   implicit real(dknd) (a-h,o-z)

   ! set up trf and ksu array pointers.
@@ -60,8 +61,8 @@
   if( jd==1 )  go to 360

   ! check reciprocity.
-  db = huge
-  dt = -huge
+  db = huge_float
+  dt = -huge_float
   do j1 = 1,mxj
     if( kst(j1)>4 .or. ksu(j1)==-3 )  cycle
     if( ksu(j1)>0 ) then
@@ -110,14 +111,14 @@
     ! tpp(i+3) in perimeter direction trf(7+i,k).
     n = 1
     do m = mx,mxtr
-      trf(3,m) = huge
+      trf(3,m) = huge_float
       t = dot_product(trf(5:7,m),trf(8:10,k))
       if( abs(t)>=1.e-6 ) then
         trf(3,m) = (trf(4,m)-dot_product(trf(5:7,m),tpp(4:6)))/t
       endif
       if( trf(3,m)<trf(3,n) )  n = m
     enddo
-    if( trf(3,n)==huge ) then
+    if( trf(3,n)==huge_float ) then
       call erprnt(1,1,0,0,0,0,0,0,&
       & '  "not all periodic boundaries specified."')
     endif
@@ -126,14 +127,14 @@
     ! gpblcm(i) is the center of each facet.
     ! check inward/outward jsu surface normal, gpblcm(3+i).
     ! set trf(2,k) = +1/-1 for jsu normal in/out of cell.
-    d2 = huge
+    d2 = huge_float
     do
       d1 = trf(3,n)
-      trf(3,n) = huge
+      trf(3,n) = huge_float
       do m = mx,mxtr
         if( trf(3,m)<trf(3,n) )  n = m
       enddo
-      if( trf(3,n)==huge )  exit
+      if( trf(3,n)==huge_float )  exit

       dx = .5*(d1+trf(3,n))
       do i = 1,3
@@ -176,7 +177,7 @@
       if( fim(1,i1)==0. .and. fim(1,i2)/=0. ) iz = -1.
       if( iz==0 ) cycle

-      if( d2==huge ) d2 = d1
+      if( d2==huge_float ) d2 = d1
       d3 = trf(3,n)
       trf(2,k) = iz
       jz = max(-iz*i1,iz*i2)
@@ -188,7 +189,7 @@
           & '  "zero-importance cell",i5,  " cannot have fill."')
       endif
     enddo
-    if( d2==huge ) then
+    if( d2==huge_float ) then
       call erprnt(1,1,1,nsf(jsu),0,0,0,0,&
         & '  "periodic surface",i5,  " must bound zero-importance cell."')
     endif
diff  -Naurd MCNP5/Source/src/isourc.F90 MCNP5_new/Source/src/isourc.F90
--- MCNP5/Source/src/isourc.F90      2003-04-30 20:11:12.000000000 -0600
+++ MCNP5_new/Source/src/isourc.F90  2004-07-22 15:14:41.000000000 -0600
@@ -9,15 +9,13 @@
```

```
   use mcnp_debug
   use mcnp_input
   use crit1_mod
+  use qttyin_mod, only:qttyin

   implicit real(dknd) (a-h,o-z)
   character(len=80) ::  hq
   character(len=9)  ::  hf
   character(len=8)  ::  hp(mipt)

-#ifdef PCDOS
-  if(lockl) call pttyin
-#endif /*def.pcdos*/
   if( irup/=0 ) then
     call qttyin(0,'  "isourc source processing."')
   endif
diff  -Naurd MCNP5/Source/src/issrc.F90 MCNP5_new/Source/src/issrc.F90
--- MCNP5/Source/src/issrc.F90       2003-04-30 20:11:12.000000000 -0600
+++ MCNP5_new/Source/src/issrc.F90   2004-07-22 15:14:41.000000000 -0600
@@ -7,11 +7,13 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)
   character(len=80) ::  hq
   character(len=9)  ::  hf
   character(len=8)  ::  hp(mipt)
+  integer(i8knd)    ::  n_i8knd

   ! replace problem names of surfaces by program names.
   nx = 1
@@ -82,9 +84,9 @@

   ! write summary totals.
   write(iuo,80)  np1,nrss,niss
-80 format(  " the original number of histories was",i10,/&
-    &  " the total number of tracks recorded was",i10,&
-    &  " from",i10,  " independent histories.")
+80 format(  " the original number of histories was",i12,/&
+    &  " the total number of tracks recorded was",i12,&
+    &  " from",i12,  " independent histories.")

   ! print surface information.
   if( ink(10)/=0 .and. njsw/=0 )  write(iuo,90)
@@ -132,19 +134,19 @@
   ! read information for the first track into the ssb array.
   read(iusr,end=320) (ssb(i),i=1,nrcd)
   nqss = dbcn(8)
-  do n=1,nqss-1
+  do n_i8knd=1,nqss-1_i8knd
     a = ssb(1)
     do
       read(iusr,end=320) (ssb(i),i=1,nrcd)
-      nrrs = nrrs+1
+      nrrs = nrrs+1_i8knd
       if( ssb(1)/=a ) exit
     enddo
   end do

   ! set up defaulted arrays.
   snit = 1.
-  if( npp>0 .and. npp/=np1 )  snit=npp/(np1+zero)
-  npp = 0
+  if( npp>0_i8knd .and. npp/=np1 )  snit=real(npp,dknd)/real(np1,dknd)
+  npp = 0_i8knd
   if( jasr(1,1)==0 ) then
     do j=1,njsr(1)+nilr(1)
       jasr(2,j) = 0
diff  -Naurd MCNP5/Source/src/itally.F90 MCNP5_new/Source/src/itally.F90
```

```
--- MCNP5/Source/src/itally.F90      2003-04-30 20:11:12.000000000 -0600
+++ MCNP5_new/Source/src/itally.F90  2004-07-22 15:14:41.000000000 -0600
@@ -16,6 +16,7 @@
   use mcnp_debug
   use mcnp_input
   use fmesh_mod,  only:  nmesh
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)
   character(len=1) :: h
@@ -58,7 +59,7 @@

     ! set ddm(1,ital)=-huge for tallies that are all negative.
     do i = 0,npert
-      ddm(1,ital+i*ntal) = -huge
+      ddm(1,ital+i*ntal) = -huge_float
     enddo

     ! set up the default bin counts.
@@ -693,7 +694,7 @@
         tds(ld+16) = rtp(j+8)
         if( tds(ld+15)==0. .and. jptal(8,ital)>3 ) then
           if( jptal(8,ital)==4 ) then
-            tds(ld+15) = huge
+            tds(ld+15) = huge_float
           else
             call erprnt(1,1,1,kl,0,0,0,0,'"cylindrical grid radius is zero for
tally",i4')
           endif
@@ -1191,13 +1192,13 @@

     ! process c, e, and t data:  cosine, energy, and time bins
 700 continue
-    tc = -huge
+    tc = -huge_float
     do i = 1,mipt
       if( ktp(i,ital)/=0 )  tc = max(tc,tco(i))
     enddo
     DO_820: do iw = 1,3
       n6 = 0
-      tn = huge
+      tn = huge_float
       if( iw==2 .and. ktp(1,ntal)/=0 .and. img==0 )  tn = emx(1)
       if( iw==3 .and. iy/=5 )  tn = tc
       if( iy==5 .and. iw==1 )  cycle DO_820
@@ -1233,7 +1234,7 @@
       n = i
 750   continue
       if( iw==2 .and. np/=3 ) then
-        b = -huge
+        b = -huge_float
         do i = 1,mipt
           if( ktp(i,ital)/=0 )  b = max(b,ecf(i))
         enddo
@@ -1368,13 +1369,13 @@
   if( ipnt(2,21,0)>=2 )  d2 = rtp(ipnt(1,21,0)+1)
   do ip = 1,mipt
     do i = 1,mxdx
-      if( ddx(ip,1,i)==huge )  ddx(ip,1,i) = d1
-      if( ddx(ip,2,i)==huge )  ddx(ip,2,i) = d2
+      if( ddx(ip,1,i)==huge_float )  ddx(ip,1,i) = d1
+      if( ddx(ip,2,i)==huge_float )  ddx(ip,2,i) = d2
     enddo
   enddo
   do i = 1,ndtt
-    if( ddg(1,i)==huge )  ddg(1,i) = d1
-    if( ddg(2,i)==huge )  ddg(2,i) = d2
+    if( ddg(1,i)==huge_float )  ddg(1,i) = d1
+    if( ddg(2,i)==huge_float )  ddg(2,i) = d2
   enddo
   ! set up the tally locators.
diff  -Naurd MCNP5/Source/src/italpr.F90 MCNP5_new/Source/src/italpr.F90
--- MCNP5/Source/src/italpr.F90      2003-04-30 20:11:14.000000000 -0600
+++ MCNP5_new/Source/src/italpr.F90  2004-07-22 15:14:41.000000000 -0600
@@ -328,7 +328,7 @@
     l = iptal(iw+5,1,ital)
     hf = ' -i'
     if( iw==1 )  write(hf,'(1pe13.5)') -1.
-    a = huge
+    a = huge_float
     do i=1,mipt
       if( ktp(i,ital)/=0 )  a=min(a,ecf(i))
     end do
diff  -Naurd MCNP5/Source/src/items.F90 MCNP5_new/Source/src/items.F90
--- MCNP5/Source/src/items.F90       2003-04-30 20:11:14.000000000 -0600
+++ MCNP5_new/Source/src/items.F90   2004-07-22 15:14:41.000000000 -0600
@@ -6,19 +6,22 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod, only: erprnt

   implicit real(dknd) (a-h,o-z)

   character(len=25) :: hh
   character(len=16) :: hs
-
+  integer(i8knd)    :: i8, il
+  integer           :: ii
+
   ! special treatment for pure text data cards.
   if( ich=='fc' .or. ich=='sc' .or. ich=='mplot' ) then
     if( mm==1 )  call nxtit1
     if( mm==2 )  call nextit
     return
   endif
-
+
   ! read the next item and determine its type.
   ! ki:  0=character  1=real  2=integer
   irc = max(1,irc)
@@ -31,28 +34,50 @@
     if( irc==-1            )  return
     ki = kdata(klin(it:iu))
     ii = 0
-    ri = 0.
-    if( ki/=0 ) then
+    i8 = 0_i8knd
+    ri = zero
+    if( ki/=0 ) then  !  Numeric item.
       hh = ' '
       hh(25-iu+it:25) = klin(it:iu)
-      if( iu-it > 8 )  ki = 1
-      if( ki==2 ) then
-        read(hh,'(i25)') ii
-        ri = ii
+
+      ! Integers that are too long must be read as reals.
+      if ( ki == 2 ) then
+        if ( klin(it:it) == '+' .or. klin(it:it) == '-' ) then
+          if ( (iu-it) > 19 ) ki = 1
+        else
+          if ( (iu-it) > 18 ) ki = 1
+        endif
+      endif
+
+      if( ki==2 ) then      ! Item read is integer
+        read(hh,'(i25)') i8
+        if (abs(i8) <= int(huge(ii),i8knd)) then
+          ii = int(i8)         ! ii may be I4 or I8 depending on platform
+        else
+          ii = -huge(ii)        ! overflow on I4
+        endif
+        ri = real(i8,dknd)
       else
         read(hh,'(e25.0)') ri
-        if( abs(ri) <= 2147483647. )  ii=nint(ri)
-        if( ri==0. )  ki=2
-        if( ii/=0  ) then
-          if( abs(ri-anint(ri)) <= 5e-14*abs(ri) ) then
-            ri = ii
+        if( abs(ri) <= real(i8limit,dknd) ) i8=nint(ri,i8knd)
+        if( abs(ri) <= real(huge(ii),dknd) ) then
+          ii=int(ri)            ! ii may be I4 or I8 depending on platform
+        else
+          ii = 0                ! overflow on I4, leave as 0 and not -huge(ii)
+        endif
+        if( ri==zero )  ki=2
+        if( i8/=0_i8knd  ) then
+          if( abs(ri-anint(ri)) <= 5e-14_dknd*abs(ri) ) then
+            ri = i8
             ki = 2
           endif
         endif
       endif
     endif
-
+    if( i8 > int(i4limit,i8knd) ) int_large_input = 1
+
     ! store nii interpolated values before the current item.
+    !  interpolation scheme does not apply to I8
     if( nii>0 ) then
       if( ki==0  )  go to 150
       dl = (ri-ritm)/(nii+1)
@@ -69,7 +94,7 @@
       end do
       go to 130
     endif
-
+
     ! look for a special r, i, m, or j item.
     if( ki/=0 )  go to 130
     ks = index('rimj',klin(iu:iu))
@@ -87,32 +112,32 @@
     select case( ks )
       case(1)  !  >>>>>  r:  repeat previous item n times.
        if( nwc==0 .or. nii/=0 )  go to 150
-        do i=1,n
+        do i=1,n
          if( mm==1 )  call nxtit1
         if( mm==2 )  call nextit
         if( ics<0 )  return
        end do
       cycle
-
+
      case(2)  !  >>>>>  i:  set up to store n interpolated values next time.
       if( nwc==0 .or. nii/=0 .or. kitm==0 )  go to 150
       hs = klin(it:iu)
      nii = n
     cycle
-
+
     case(3)  !  >>>>>  m:  store an item h times the value of the previous item.
      if( nwc==0 .or. nii/=0 .or. kitm==0 .or. iu==it )  go to 150
     ritm = h*ritm
     iitm = n*iitm
    go to 140
-
+
    case(4)  !  >>>>>  j:  skip n storage spaces.
     if( krq(4,ica)==0 )  go to 150
```

```
         nwc = nwc+n
         nii = -1
         cycle
     end select
-
+
     ! set up the current item and store it.
 130 continue
     kitm = ki
@@ -120,13 +145,14 @@
     nitm = iu-it+1
     ritm = ri
     iitm = ii
+    i8itm = i8
     nii  = 0
 140 continue
     if( mm==1 )  call nxtit1
     if( mm==2 )  call nextit
     if( ics<0 )  return
   enddo
-
+
 150 continue
   ics = -1
   if( mm==1  )  return
diff  -Naurd MCNP5/Source/src/iwtwnd.F90 MCNP5_new/Source/src/iwtwnd.F90
--- MCNP5/Source/src/iwtwnd.F90       2003-04-30 20:11:14.000000000 -0600
+++ MCNP5_new/Source/src/iwtwnd.F90   2004-07-22 15:14:41.000000000 -0600
@@ -6,6 +6,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

@@ -46,7 +47,7 @@
     ! check input importances and weight-windows.
     a = 0.
     b = 0.
-    c = huge
+    c = huge_float
     do i=1,mxa
       a = max(a,fim(ip,i))
       if( fim(ip,i)/=0. )  c=min(c,fim(ip,i))
diff  -Naurd MCNP5/Source/src/ixsdir.F90 MCNP5_new/Source/src/ixsdir.F90
--- MCNP5/Source/src/ixsdir.F90       2003-04-30 20:11:16.000000000 -0600
+++ MCNP5_new/Source/src/ixsdir.F90   2004-07-22 15:14:41.000000000 -0600
@@ -9,6 +9,8 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use qttyin_mod, only: qttyin
+  use erprnt_mod, only: erprnt

   implicit real(dknd) (a-h,o-z)
   character(len=1)   :: ha
@@ -81,18 +83,10 @@
   hl = xsdir
   inquire( file=hl,  exist=file_exists )
   if( .not.file_exists ) then
-#ifdef PCDOS
-    if( len_trim(hdpath) > 0 )  hl=hdpath(1:len_trim(hdpath))//'\'//xsdir
-#else
     if( len_trim(hdpath) > 0 )  hl=hdpath(1:len_trim(hdpath))//'/'//xsdir
-#endif
     inquire( file=hl,  exist=file_exists )
     if( .not.file_exists ) then
-#ifdef PCDOS
-      if( len_trim(hdpth) > 0 )  hl=hdpth(1:len_trim(hdpth))//'\'//xsdir
-#else
```

```
       if(len_trim(hdpth) > 0 )  hl=hdpth(1:len_trim(hdpth))//'/'//xsdir
-#endif
       inquire( file=hl,  exist=file_exists )
       if( .not.file_exists ) then
         write(jtty,'(  " searched directories:"/a80/a80)')hdpath,hdpth
@@ -197,9 +191,6 @@
 240 continue
   hz = hl(1:10)
   ha = hz(10:10)
-#ifdef PCDOS
-  if( lockl)  call pttyin
-#endif /*def.pcdos*/
   if( irup/=0 )  call qttyin(0,'"reading xsdir '//hz//'"')

   ! look for a match with the directory file entry.
diff  -Naurd MCNP5/Source/src/jdecod.F90 MCNP5_new/Source/src/jdecod.F90
--- MCNP5/Source/src/jdecod.F90      2003-04-30 20:11:16.000000000 -0600
+++ MCNP5_new/Source/src/jdecod.F90  2004-07-22 15:14:41.000000000 -0600
@@ -14,6 +14,7 @@
   use mcnp_plot
   use gxsub
   use mcplot_module
+  use qttyin_mod, only: qttyin

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/jsourc.F90 MCNP5_new/Source/src/jsourc.F90
--- MCNP5/Source/src/jsourc.F90      2003-04-30 20:11:18.000000000 -0600
+++ MCNP5_new/Source/src/jsourc.F90  2004-07-22 15:14:41.000000000 -0600
@@ -9,6 +9,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)
   character ht*12
diff  -Naurd MCNP5/Source/src/keypro.F90 MCNP5_new/Source/src/keypro.F90
--- MCNP5/Source/src/keypro.F90      2003-04-30 20:11:20.000000000 -0600
+++ MCNP5_new/Source/src/keypro.F90  1969-12-31 17:00:00.000000000 -0700
@@ -1,44 +0,0 @@
-!+ $Id: keypro.F90,v 1.3 2002/12/03 19:23:42 ljcox Exp $
-! Copyright LANL/UC/DOE - see file COPYRIGHT_INFO
-
-#ifdef QWIN
-function keypro(nc,mp,lp)
-  use mcnp_debug
-  integer*4  keypro
-  ! allow processing of keyboard input for interrupts.
-  !DEC$ ATTRIBUTES STDCALL, ALIAS: '_keypro@12' :: keypro
-  use dfwin
-  use dflib
-  keypro=0
-  !
-  ! check mp for character code 67 (letter c).
-  ! check if cntl key was depressed (16th bit nonzero).
-  ! only send signal when key is released (lp 32nd bit nonzero).
-  if(nc >= 0.and.mp == 67.and.getkeystate(vk_control) < 0)then
-    if(lp < 0)i=raiseqq(sig$int)
-    keypro=1
-  end if
-  return
-end function keypro
-function initialsettings()
-  logical*4 initialsettings
-  ! initialize the frame window and set its parameters.
-  ! this function is called before main execution.
-  use dfwin
-  use dflib
-  external keypro
-  !DEC$ ATTRIBUTES STDCALL, ALIAS: '_keypro@12' :: keypro
-  type (qwinfo) wi
-  k=setexitqq(qwin$exitpersist)
-  !
-  ! enable keyboard interrupts on the main execute thread.
-  k=setwindowshookex(wh_keyboard,loc(keypro),0,&
-    & getcurrentthreadid())
-  !
-  ! set the frame window to its maximum size.
-  k=getwsizeqq(qwin$framewindow,qwin$sizemax,wi)
-  k=setwsizeqq(qwin$framewindow,wi)
-  initialsettings=.true.
-  return
-end function initialsettings
-#endif /*def.qwin*/
diff  -Naurd MCNP5/Source/src/main.F90 MCNP5_new/Source/src/main.F90
--- MCNP5/Source/src/main.F90 2003-04-30 20:11:26.000000000 -0600
+++ MCNP5_new/Source/src/main.F90     2004-07-22 15:14:41.000000000 -0600
@@ -45,10 +45,6 @@
   ! Modules used:
   use lx5_mod
   use dmmp, only: dm_enrl, dm_term, dm_bcast, dm_ntasks
-#ifdef QWIN
-  use dflib,only: setactiveqq,  getwindowconfig, setwindowconfig, &
-    &             windowconfig, setexitqq,       qwin$exitpersist
-#endif
   use mcnp_global
   use mcnp_debug
   use gxsub, only : gxquit
@@ -62,12 +58,11 @@

   character(len=18) :: hn = '                  '
   character(len=80) :: hp
-#ifdef QWIN
-  logical*4 st
-  type (windowconfig) wc
-#endif
   ! --------------------------------------------------------------

+  ! Nullify pointer in mcnp_global
+  call mcnp_global_nullify
+
   ! Initialize cpu clock.
   call secnd(tq)

@@ -84,14 +79,6 @@
   ! set up terminal & graphics
   itty = 5
   jtty = 6
-#ifdef QWIN
-  ! Change the execute window title (unit 0).
-  k  = setactiveqq(0)
-  st = getwindowconfig(wc)
-  wc.title="MCNP Execute Window"C
-  st = setwindowconfig(wc)
-  k  = setexitqq(qwin$exitpersist)
-#endif

   ! Check for conversion of source code to upper case.
   ! One "A" must be upper case.
@@ -106,28 +93,35 @@

   ! Initialize module data.
   gfixcm(1:nfixcm)   = 0.      ! call fix_init
+  i8fixcm(1:l8fixcm) = 0_i8knd
   jfixcm(1:lfixcm)   = 0
-  i8fixcm(1:l8fixcm) = 0
-
-  gvarcm(1:nvarcm) = 0.      ! call var_init
-  avarcm(1:13)     = 0.
-  jvarcm(1:lvarcm) = 0
+
+  gvarcm(1:nvarcm)   = 0.      ! call var_init
+  i8varcm(1:l8varcm) = 0_i8knd
+  avarcm(1:13)       = 0.
+  jvarcm(1:lvarcm)   = 0

-  gephcm(1:nephcm) = 0.      ! call eph_init
-  jephcm(1:lephcm) = 0
+  gephcm(1:nephcm)   = 0.      ! call eph_init
+  i8ephcm(1:l8ephcm) = 0_i8knd
+  jephcm(1:lephcm)   = 0
   lockl  = .false.

-  gtskcm(1:ntskcm) = 0.      ! call tsk_init
-  jtskcm(1:ltskcm) = 0
-  ktskpt(1:ltskpt) = 0
+  gtskcm(1:ntskcm)   = 0.      ! call tsk_init
+  i8tskcm(1:l8tskcm) = 0_i8knd
+  jtskcm(1:ltskcm)   = 0
+  ktskpt(1:ltskpt)   = 0

   gpblcm(1:npblcm) = 0.      ! call pbl_init
   jpblcm(1:lpblcm) = 0
   gpb9cm(1:mpb,1:npblcm) = 0.
   jpb9cm(1:mpb,1:lpblcm) = 0

+  ! Initialize specific variables in fixcom to correct default.
+  flag_speed_tally_used=-1
+
+
   iterm = 3
-#ifndef GKSSIM
+#if !defined(PLOT) && !defined(MCPLOT)
 #ifdef AIX
   iterm = 1
 #endif
@@ -172,6 +166,7 @@
   ntasks = max(1,ntasks)    ! threads
   ktask  = 0                ! threads

+
   ! Set up default values for unspecified file names.
   do i = 1,ndef
     if( isub(i)==' ' .and. iname/=' ' .and. i<=10 )  then
diff  -Naurd MCNP5/Source/src/mapmaz.F90 MCNP5_new/Source/src/mapmaz.F90
--- MCNP5/Source/src/mapmaz.F90      2003-04-30 20:11:26.000000000 -0600
+++ MCNP5_new/Source/src/mapmaz.F90  2004-07-22 15:14:41.000000000 -0600
@@ -8,6 +8,9 @@

   use mcnp_global
   use mcnp_debug
+  use qttyin_mod, only:qttyin
+  use erprnt_mod, only: erprnt
+

   implicit real(dknd) (a-h,o-z)

@@ -91,9 +94,6 @@
 130 continue
   if( ie(lv)<ne(lv) )  go to 70
   kp = (100*mz)/nmaz
-#ifdef PCDOS
-  if( lockl)  call pttyin
-#endif /*def.pcdos*/
  if( irup/=0 )  call qttyin(kp,'  "print table 128.",i7, "%"')
   lv = lv-1
   if( lv>=0 )  go to 130
diff  -Naurd MCNP5/Source/src/mbody.F90 MCNP5_new/Source/src/mbody.F90
--- MCNP5/Source/src/mbody.F90       2003-04-30 20:11:26.000000000 -0600
+++ MCNP5_new/Source/src/mbody.F90   2004-07-22 15:14:42.000000000 -0600
@@ -8,6 +8,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/mbodyo.F90 MCNP5_new/Source/src/mbodyo.F90
--- MCNP5/Source/src/mbodyo.F90      2003-04-30 20:11:28.000000000 -0600
+++ MCNP5_new/Source/src/mbodyo.F90  2004-07-22 15:14:42.000000000 -0600
@@ -8,6 +8,7 @@

   use mcnp_debug
   use mcnp_params, only: dknd
+  use erprnt_mod
   implicit real(dknd) (a-h,o-z)

   real(dknd) :: sc(33)
diff  -Naurd MCNP5/Source/src/mbodyr.F90 MCNP5_new/Source/src/mbodyr.F90
--- MCNP5/Source/src/mbodyr.F90      2003-04-30 20:11:28.000000000 -0600
+++ MCNP5_new/Source/src/mbodyr.F90  2004-07-22 15:14:42.000000000 -0600
@@ -6,6 +6,7 @@

   use mcnp_debug
   use mcnp_params, only: dknd
+  use erprnt_mod
   implicit real(dknd) (a-h,o-z)

   real(dknd) :: sc(33)
diff  -Naurd MCNP5/Source/src/mbodys.F90 MCNP5_new/Source/src/mbodys.F90
--- MCNP5/Source/src/mbodys.F90      2003-04-30 20:11:30.000000000 -0600
+++ MCNP5_new/Source/src/mbodys.F90  2004-07-22 15:14:42.000000000 -0600
@@ -7,6 +7,7 @@

   use mcnp_debug
   use mcnp_params, only: dknd
+  use erprnt_mod
   implicit real(dknd) (a-h,o-z)

   real(dknd) :: sc(33)
diff  -Naurd MCNP5/Source/src/mc.c MCNP5_new/Source/src/mc.c
--- MCNP5/Source/src/mc.c     2003-04-30 20:11:30.000000000 -0600
+++ MCNP5_new/Source/src/mc.c 2004-07-22 15:14:42.000000000 -0600
@@ -8,7 +8,7 @@
 #include <time.h>
 #include <sys/stat.h>
 #else
-#if defined(DEC) || defined(PCDOS)
+#if defined(DEC)
 #include <stdlib.h>
 #endif
 #endif
@@ -105,28 +105,6 @@
 #define XGQCF  __stdcall XGQCF
 #define XLNWID __stdcall XLNWID
 #endif
-#ifdef PCDOS
-/*  Lower case with appended underscore */
-#define XGOPWK xgopwk_
-#define XGCLWK xgclwk_
-#define XGACWK xgacwk_
-#define XGDAWK xgdawk_
-#define XGCLRW xgclrw_
-#define XGUWK  xguwk_
-#define XGPL   xgpl_
-#define XGTX   xgtx_
-#define XGSCHH xgschh_
-#define XGSCHU xgschu_
-#define XGSWN  xgswn_
-#define XGSWK  xgswk_
-#define XGRQLC xgrqlc_
-#define XGINST xginst_
```

```
-#define XGRQST xgrqst_
-#define XGFA   xgfa_
-#define XGSPLC xgsplc_
-#define XGQCF  xgqcf_
-#define XLNWID xlnwid_
-#endif
 #endif
 /* Define charater type from Fortran if not previously defined */
 #ifndef CDEF
diff  -Naurd MCNP5/Source/src/mcnp_data.F90 MCNP5_new/Source/src/mcnp_data.F90
--- MCNP5/Source/src/mcnp_data.F90   2003-04-30 20:11:30.000000000 -0600
+++ MCNP5_new/Source/src/mcnp_data.F90       2004-07-22 15:14:42.000000000 -0600
@@ -148,21 +148,21 @@
     &    'kz ', 'sq ', 'gq ', 'tx ', 'ty ', 'tz ', 'x  ', 'y  ', 'z  ', 'box',&
     &    'rpp', 'sph', 'rcc', 'rec', 'ell', 'trc', 'wed', 'arb', 'rhp' /)

-  character(len=58) :: rfq(15) = & != Partial formats for termination messages.
-    & (/'  "     seconds before job time limit."                   ', &
-    &   '  "when           particle histories were done."          ', &
-    &   '  "because     particles got lost."                       ', &
-    &   '  "by tty interrupt."                                     ', &
-    &   '  "when it had used      minutes of computer time."       ', &
-    &   '  "by sense switch 1."                                    ', &
-    &   '  "when       kcode cycles were done."                    ', &
-    &   '  "when            events were written to the ptrac file."', &
-    &   '  "because of bad trouble."                               ', &
-    &   '  "at end of surface-source input file."                  ', &
-    &   '  "because of fatal errors."                              ', &
-    &   '                                                          ', &
-    &   '                                                          ', &
-    &   '                                                          ', &
-    &   '                                                          '  /)
+  character(len=60) :: rfq(15) = & != Partial formats for termination messages.
+    & (/'  "     seconds before job time limit."                     ', &
+    &   '  "when             particle histories were done."          ', &
+    &   '  "because     particles got lost."                         ', &
+    &   '  "by tty interrupt."                                       ', &
+    &   '  "when it had used       minutes of computer time."        ', &
+    &   '  "by sense switch 1."                                      ', &
+    &   '  "when         kcode cycles were done."                    ', &
+    &   '  "when              events were written to the ptrac file."', &
+    &   '  "because of bad trouble."                                 ', &
+    &   '  "at end of surface-source input file."                    ', &
+    &   '  "because of fatal errors."                                ', &
+    &   '                                                            ', &
+    &   '                                                            ', &
+    &   '                                                            ', &
+    &   '                                                            '  /)

 end module mcnp_data
diff  -Naurd MCNP5/Source/src/mcnp_global.F90 MCNP5_new/Source/src/mcnp_global.F90
--- MCNP5/Source/src/mcnp_global.F90 2003-04-30 20:11:32.000000000 -0600
+++ MCNP5_new/Source/src/mcnp_global.F90     2004-07-22 15:14:42.000000000 -0600
@@ -164,6 +164,24 @@
     &  mazu(:),   &  != Universe/lattice map pointers. See App. E.
     &  mfl(:,:)      != Fill data for each cell.

+  ! Declaration of flags to check speed tally requirements
+  integer :: &
+    &  flag_speed_tally_ok = 0 , & != Overall flag to enable lattice speed tally
modifications
+                                  !   for lattices (modifications to tally.F90)
+                                  !   1 = criteria to use lattice speed tally has been
fulfilled.
+                                  !   0 = default (not tested yet or no lat=1 keyword
in geometry)
+                                  !  -1 = lat present, but criteria to use lattice
speed tally has NOT been fulfilled.
+    &  flag_speed_tally_force =0   != Overall flag to force use of lattice speed tally
modifications
+                                  !   for lattices (modifications to tally.F90)
+                                   !   1 = forced use of lattice speed tally   0 =
default (neither)
+                                   !  -1 = forced use of regualar tally routine (i.e.
slow)
+  integer, ALLOCATABLE :: &
+    &  flag_speed_tally_fm(:,:), & !  Flag to check if each f4 has corresponding fm card
present. 0=no 1=yes
+                                   !    the fm card should be present to enable the
lattice speed tally modifications
+    &  flag_speed_tally_de(:,:)    !  Flag to check if each f4 has corresponding de df
cards present. 0=no 1=yes
+                                   !    the de df cards should be present to enable the
lattice speed tally modifications
+                                   !  Both of these variables place icn (the user-
defined tally number) in the 2nd dimension.
+
   ! "VDAC" --  variable dynamically allocated common
   real(dknd),  ALLOCATABLE:: &
     &  ara(:),     &  != Areas of the surfaces in the problem.
@@ -176,10 +194,11 @@
   integer,    ALLOCATABLE :: &
     &  jfq(:,:), &  != Order for printing tally results.
     &  lfcl(:),  &  != Cells where fission is treated like capture.
-    &  npsw(:),  &  != For each surface source surface, the last history in
-                    !=  which a track crossed it.
     &  nsl(:,:), &  != Summary information for surface source file.
     &  ntbb(:,:)    != Counts of scores beyond the last bin.
+  integer(i8knd), ALLOCATABLE :: &
+    &  npsw(:)      != For each surface source surface, the last history in
+                    !=  which a track crossed it.

   ! Task "VDAC" (have l<var> and k<var> references).
   real(dknd),  ALLOCATABLE:: &
@@ -206,15 +225,16 @@
     &  wns(:,:),   &  != Actual frequencies of source sampling.
     &  wwfa(:)        != Weight window generator entering weight array.
   integer,    ALLOCATABLE :: &
-    &  isef(:,:), &  != Source position tries and rejections.
     &  laj(:),    &  != Cells on the other sides of the surfaces in LJA.
     &  lcaj(:),   &  != For each surface in LJA, a pointer into the list of
                      !=  other-side cells in LAJ.
     &  lse(:),    &  != Cells where source particles have appeared.
     &  maze(:),   &  != Universe/lattice map values. See App. E.
-    &  nbal(:),   &  != Number of histories processed by each task.
     &  ndpf(:,:), &  != Accounts of detector scores that failed.
-    &  ndr(:),    &  != List of discrete-reaction rejections.
+    &  ndr(:)        != List of discrete-reaction rejections.
+  integer(i8knd), ALLOCATABLE :: &
+    &  isef(:,:), &  != Source position tries and rejections.
+    &  nbal(:),   &  != Number of histories processed by each task.
     &  nhsd(:,:)     != Number in history score distribution which counts
                      !=  nonzero scores for statistical analysis.

@@ -309,4 +329,29 @@
     &  lsg(:),   &  != Kind of line to plot for each segment of curve.
     &  ncs(:)       != Number of curves where surface meets plot plane.

+  CONTAINS
+
+  subroutine mcnp_global_nullify
+
+    ! This routine ensures that the pointers declared above have
+    ! a defined association status.
+    implicit none
+
+    nullify( ncl )
+    nullify( nsfm )
+    nullify( kxs )
+    nullify( laf )
+    nullify( jun )
+    nullify( mat )
+    nullify( mazp )
+    nullify( mazu )
+    nullify( mfl )
+    nullify( xss )
+    nullify( exs )
+    nullify( zst )
+
+    return
+
+  end subroutine mcnp_global_nullify
+
 end module mcnp_global
diff  -Naurd MCNP5/Source/src/mcnp_input.F90 MCNP5_new/Source/src/mcnp_input.F90
--- MCNP5/Source/src/mcnp_input.F90   2003-11-05 17:23:12.000000000 -0700
+++ MCNP5_new/Source/src/mcnp_input.F90      2004-07-22 15:14:42.000000000 -0600
@@ -8,7 +8,7 @@
   integer,private :: i                  ! Local loop variable.

   ! Module Parameters:
-  integer,parameter :: nkcd   = 101     != Number of different types of input cards.
+  integer,parameter :: nkcd   = 102     != Number of different types of input cards.
   integer,parameter :: ntalmx = 1000    != Maximum number of tallies.
   integer,parameter :: mopts  = 7       != Number of M card options (gas, estep, etc.).

@@ -84,6 +84,8 @@
   integer :: nwc                        != Count of items on current input card.
   integer :: nxsc  = 0                  != Number of XSn cards.

+  integer(i8knd) :: i8itm               != 8 Byte Integer form of current item from
input card.
+
   character(len=5 ) ::  cnm(nkcd)
   character(len=80) ::  hitm
   character(len=80) ::  hlin
@@ -234,5 +236,5 @@
   data cnm(99), (krq(i,99),i=1,7) / 'rand ',0,0, 0,0, 0,  12,0 /
   data cnm(100),(krq(i,100),i=1,7)/ 'irmc ',1,0, 0,1, 2,  50,0 /
   data cnm(101),(krq(i,101),i=1,7)/ 'rrmc ',1,0, 0,1,-1,  50,0 /
-
+  data cnm(102),(krq(i,102),i=1,7)/ 'spdtl',0,0, 0,0, 0,   1,0 /
 end module mcnp_input
diff  -Naurd MCNP5/Source/src/mcnp_params.F90 MCNP5_new/Source/src/mcnp_params.F90
--- MCNP5/Source/src/mcnp_params.F90 2003-11-05 17:23:12.000000000 -0700
+++ MCNP5_new/Source/src/mcnp_params.F90     2004-07-22 15:14:42.000000000 -0600
@@ -55,7 +55,7 @@
     & mcnp_opt_multt    = mcnp_opt_omp,                      &
     & mcnp_opt_parallel = mcnp_opt_multp .or. mcnp_opt_multt

-  real(dknd),parameter :: huge = 1.0d+36      != A very large number.
+  real(dknd),parameter :: huge_float = 1.0d+36      != A very large number.

   ! array dimensions.
   integer,parameter :: lmrkp  = 651     != Minimum number of kcode cycles to plot
(mrkp).
@@ -137,11 +137,43 @@
     & euler  = .577215664901532861d+0,    & != Euler constant used in electron
transport.
     & fscon  = 137.0393d+0,               & != Inverse fine-structure constant.
     & hsll   = 1.0d-30,                   & != History score lower bin bound.
-    & one    = 1.0d+0,                    & != Floating-point constant 1. for arguments.
     & pie    = 3.1415926535898d+0,        & != pi.
     & planck = 4.135732d-13,              & != Planck constant.
-    & slite  = 299.7925d+0,               & != Speed of light.
-    & third  = one/3.0d+0,                & != Floating-point constant 1/3.
-    & zero   = 0.0d+0                       != Floating-point constant 0. for arguments.
+    & slite  = 299.7925d+0                  != Speed of light.
+   ! Frequently used real numbers:
+  real(dknd),parameter ::                 &
+    & one         = 1.0_dknd,             &
+    & two         = 2.0_dknd,             &
+    & three       = 3.0_dknd,             &
+    & four        = 4.0_dknd,             &
+    & five        = 5.0_dknd,             &
+    & six         = 6.0_dknd,             &
+    & seven       = 7.0_dknd,             &
+    & eight       = 8.0_dknd,             &
+    & nine        = 9.0_dknd,             &
+    & ten         = 10.0_dknd,            &
+    & twenty      = 20.0_dknd,            &
+    & thirty      = 30.0_dknd,            &
+    & forty       = 40.0_dknd,            &
+    & fifty       = 50.0_dknd,            &
+    & sixty       = 60.0_dknd,            &
+    & seventy     = 70.0_dknd,            &
+    & eighty      = 80.0_dknd,            &
+    & ninety      = 90.0_dknd,            &
+    & hundred     = 100.0_dknd,           &
+    & one_eighty  = 180.0_dknd,           &
+    & three_sixty = 360.0_dknd,           &
+    & thousand    = 1000.0_dknd,          &
+    & third       = one/three,            &
+    & half        = 0.5_dknd,             &
+    & tenth       = 0.1_dknd,             &
+    & hundredth   = 0.01_dknd,            &
+    & thousandth  = 0.001_dknd,           &
+    & zero        = 0.0_dknd
+
+  integer(i8knd), parameter :: i8limit = huge(1_i8knd)    != Max integer*8 ~1E20
+  integer,        parameter :: i4limit = huge(1_i4knd)    != Max integer*4

 end module mcnp_params
+
+
diff  -Naurd MCNP5/Source/src/mcplot_module.F90 MCNP5_new/Source/src/mcplot_module.F90
--- MCNP5/Source/src/mcplot_module.F90       2003-11-05 17:23:12.000000000 -0700
+++ MCNP5_new/Source/src/mcplot_module.F90   2004-07-22 15:14:42.000000000 -0600
@@ -244,9 +244,6 @@
     ! Make plots of tally data in response to input from the user.

     ! Modules:
-#ifdef LAHEY
-    use WINTERACTER
-#endif
     use dynamic_arrays

     hovr = 'mcplot'
@@ -305,9 +302,6 @@
         read(ifile,'(a80)',iostat=istat) klin
         if( istat==0 ) then
           continue
-#ifdef LAHEY
-          call IGrAreaClear()
-#endif
         else
           klin = ' '
           if( ifile/=itty ) then
@@ -1003,8 +997,8 @@
     if( kxsplt/=0 ) then
       call abvals( ab1(1:), 1 )
       call exord(2)
-      xyzmn(3) = huge
-      xyzmx(3) = -huge
+      xyzmn(3) = huge_float
+      xyzmx(3) = -huge_float
       call exord(1)
       ym = xyzmn(3)
       yx = xyzmx(3)
@@ -1234,8 +1228,8 @@
       enddo
     elseif( i>=36 .and. i<40 ) then
       ! Make criticality frequency plot.
-      ablo(1) = huge
-      abhi(1) = -huge
+      ablo(1) = huge_float
+      abhi(1) = -huge_float
       ki = i-35
       if( i==39)  ki = 20
       do j = k0,kcz
@@ -1307,9 +1301,9 @@
     ! Check log plot ordinate array for non-positive values.
 200 continue
     if( abs(lax-3)/=1) go to 260
-    ap = huge
-    am = huge
-    ax = -huge
+    ap = huge_float
+    am = huge_float
+    ax = -huge_float
     do i = 1,npt(1)
       am = min(am,ord(i))
       if( ord(i)>0.)  ap = min(ap,ord(i))
@@ -1349,8 +1343,8 @@
     ! Determine axis limits.
     if( jlim(1)==0) call gxlims(ablo(1),abhi(1),xyzmn(1),xyzmx(1),n,max(0,lax-2))
     if( jlim(2)/=0) go to 300
-    a = huge
-    b = -huge
+    a = huge_float
+    b = -huge_float
     do i = 1,npt(1)
       a = min(a,.95*ord(i))
       b = max(b,1.05*ord(i))
@@ -1399,8 +1393,8 @@
     if( washme )  ab2(0) = ablo(2)

     if( inform/=0)  return
-    xyzmn(3) = huge
-    xyzmx(3) = -huge
+    xyzmn(3) = huge_float
+    xyzmx(3) = -huge_float
     call exord(1)
     if( xyzmx(3)>xyzmn(3))  go to 380
     write(jtty,370)
@@ -1499,7 +1493,6 @@
     call gslwsc(1.)
     call gsln(1)
     if( mcolor>2 ) call gsplci( kcolor(ncolor+1),0)
-
     ! Add error bars if required.
     if( noerbr==0 .and. (itfc<=1 .or. itfc>=41 .and. itfc<=43 .or. &
       & itfc>=46 .and. itfc<=47 .or. itfc==52 .or. itfc==54 .or. &
@@ -2122,7 +2115,7 @@
     ablo(iv) = 0.
     if( ifree(iv)==7 .and. ktfile==1 ) then
       ! Energy bins from RUNTPE.
-      ablo(iv) = huge
+      ablo(iv) = huge_float
       j = jptal(3,mtal)
       do i = 1,mipt
         if( mod(j,2)/=0 )  ablo(iv) = min(ablo(iv),ecf(i))
@@ -2290,9 +2283,9 @@

     ! Find next minimum energy of given isotopes.
     ! Eo=-huge for photons.
-    eo = -huge
+    eo = -huge_float
     do
-      en = huge
+      en = huge_float
       do j = 1,kxsplt
         if( kxspen(j)<=0 .or. kxspxs(j)<=0 )  cycle
         if( kxspnx(mxe1+j)>kxspnx(j)       )  cycle
@@ -2302,7 +2295,7 @@
```

```
         k = j
         en = e
       enddo
-      if( en==huge )  exit
+      if( en==huge_float )  exit
       kxspnx(mxe1+k) = kxspnx(mxe1+k)+1
       if( en<=eo )  cycle
       eo = en
@@ -2338,15 +2331,15 @@

     i = 0
     if( ktfile==1 ) then
-      t = max(1,nps)
+      t = max(1_i8knd,nps)
       if( nsr==71  .and. kcz>=ikz    )  t = nsrck*(kcz-ikz)
       call ra_kcheck(kcheck)
       call ra_lcheck(lsav,ldif)
       if( kcheck>0 .and. kcz>=lsav )  t = nsrck*(kcz-lsav)
       if( kc8<0 )  t = t+nsrck-wt0*nsa
       if( knrm/=0 .or. (kcheck>0 .and. kcz<lsav) )  t = pax(1,1,1)
-      if( nsr==6 .and. nrrs>=nrss )  t = max(1,np1)
-      if( nsr==6 .and. nrrs<nrss )  t = max(1,npsr)
+      if( nsr==6 .and. nrrs>=nrss )  t = max(1_i8knd,np1)
+      if( nsr==6 .and. nrrs<nrss )  t = max(1_i8knd,npsr)
       fp = 1./max(t,one)
     endif
     j = jptal(5,mtal)
@@ -2609,11 +2602,6 @@
     ! Return 1 for any termination command, otherwise 0.
     ! Return ln=1 for end of line, otherwise ln=0.

-    ! Modules:
-#ifdef LAHEY
-    use WINTERACTER
-#endif
-
     character(len=80) :: ha
     character(len=8)  :: hm

@@ -3033,7 +3021,6 @@
         lax  = 3
         if( itfc==6 ) lax = 4
       endif
-
       return

     case( 33 )
@@ -3290,12 +3277,7 @@
     write(jtty,'(  " bad argument of ",a8,  " command:  ",a8)')hm,ha

 9020 continue
-#ifdef LAHEY
-    if( koplot/=0 )  call WindowClose
-    write(jtty,'(  " the rest of the command line is ignored.")')
-#else
     write(jtty,'(  " the rest of the command line is ignored.")')
-#endif
     ln = 1

 9030 continue
@@ -4154,7 +4136,7 @@
         fn = (ord(j)-ord(j-1))*f1+(ord(j+1)-ord(j))*f2
         fd = (ab1(j)-ab1(j-1))*f1+(ab1(j+1)-ab1(j))*f2
         t(j-i+1) = 1.
-        if( fn/=0. )  t(j-i+1) = huge*1e-10
+        if( fn/=0. )  t(j-i+1) = huge_float*1e-10
         if( fd/=0. )  t(j-i+1) = fn/fd
       enddo
       if( i==1 ) then
diff  -Naurd MCNP5/Source/src/mctalw.F90 MCNP5_new/Source/src/mctalw.F90
--- MCNP5/Source/src/mctalw.F90      2003-04-30 20:11:38.000000000 -0600
```

```
+++ MCNP5_new/Source/src/mctalw.F90   2004-07-22 15:14:42.000000000 -0600
@@ -22,23 +22,23 @@
   call unique(hm,jtty)
   open(iut,file=hm,status='new')
   write(ht,'(f16.0)')rnr
-  if( mct<0  )  write(iut,'(35x,i5,i11,a15)')knod,nps,ht(1:15)
-  if( mct>=0 )  write(iut,'(2a8,a19,i5,i11,a15)')kod,ver,probid,knod,nps,ht(1:15)
+  if( mct<0  )  write(iut,'(35x,i5,i12,a15)')knod,nps,ht(1:15)
+  if( mct>=0 )  write(iut,'(2a8,a19,i5,i12,a15)')kod,ver,probid,knod,nps,ht(1:15)
   write(iut,'(1x,a79)') aid
   if( npert==0 ) write(iut,'( "ntal",i6)')ntal
   if( npert/=0 ) write(iut,'( "ntal",i6, " npert",i6)')ntal,npert
   write(iut,'(16i5)') (jptal(1,i),i=1,ntal)

   ! calculate the tally normalization factor.
-  t = max(1,nps)
+  t = max(1_i8knd,nps)
   if( nsr==71 .and. kcz>=ikz)  t = nsrck*(kcz-ikz)
   call ra_kcheck( kcheck )
   call ra_lcheck( lsav, ldif )
   if( kcheck>0 .and. kcz>=lsav)  t = nsrck*(kcz-lsav)
   if( kc8<0)  t = t+nsrck-wt0*nsa
   if( knrm/=0 .or. (kcheck>0 .and. kcz<lsav) )  t = pax(1,1,1)
-  if( nsr==6 .and. nrrs>=nrss)  t = max(1,np1)
-  if( nsr==6 .and. nrrs<nrss)   t = max(1,npsr)
+  if( nsr==6 .and. nrrs>=nrss)  t = max(1_i8knd,np1)
+  if( nsr==6 .and. nrrs<nrss)   t = max(1_i8knd,npsr)
   fp = 1./max(t,one)

   ! determine if surfaces have macrobody facets.
diff  -Naurd MCNP5/Source/src/messages.F90 MCNP5_new/Source/src/messages.F90
--- MCNP5/Source/src/messages.F90     2003-11-05 17:23:12.000000000 -0700
+++ MCNP5_new/Source/src/messages.F90 2004-07-22 15:14:42.000000000 -0600
@@ -24,13 +24,11 @@
   !    msgZ_put(array, nitem)       for zero-based arrays
   !        (nstart is first element of the array  if lower bound is not 1.)
   !    msg_get and msgZ_get have same arguemnts as msg_put and msgZ_put
-
+  use mcnp_params, only: i4knd, i8knd, rknd, dknd, iuo, zero
+  use mcnp_iofiles, only: jtty
   implicit none

-  integer, parameter, PRIVATE ::   iknd = selected_int_kind( 9)
-  integer, parameter, PRIVATE ::  i8knd = selected_int_kind(18)
-  integer, parameter, PRIVATE ::   rknd = selected_real_kind(6,37)
-  integer, parameter, PRIVATE ::   dknd = selected_real_kind(15,307)
+  private i4knd, i8knd, rknd, dknd, iuo, jtty

   interface msg_get
     ! ==> specific routines used for generic subroutine:
@@ -38,7 +36,7 @@
     module procedure msg_get_i1, msg_get_i2, msg_get_i3, msg_get_i4
     module procedure msg_get_d1, msg_get_d2, msg_get_d3, msg_get_d4
     module procedure msg_get_d5
-    module procedure msg_get_i80,msg_get_i81
+    module procedure msg_get_i80,msg_get_i81,msg_get_i82
   end interface

   interface msg_put
@@ -47,7 +45,7 @@
     module procedure msg_put_i1, msg_put_i2, msg_put_i3, msg_put_i4
     module procedure msg_put_d1, msg_put_d2, msg_put_d3, msg_put_d4
     module procedure msg_put_d5
-    module procedure msg_put_i80,msg_put_i81
+    module procedure msg_put_i80,msg_put_i81,msg_put_i82
   end interface

 ! msgZ_get and msgZ_put to handle arrays with first lbound /=1.
@@ -55,11 +53,13 @@
      interface msgZ_get
        module procedure msgZ_get_d1, msgZ_get_d3
        module procedure msgZ_get_i1, msgZ_get_i2
+       module procedure msgZ_get_i81, msgZ_get_i82
      end interface

      interface msgZ_put
        module procedure msgZ_put_d1, msgZ_put_d3
        module procedure msgZ_put_i1, msgZ_put_i2
+       module procedure msgZ_put_i81, msgZ_put_i82
      end interface

 contains
@@ -186,7 +186,7 @@
      use dmmp
      implicit none
      !
-     integer(iknd), intent(out)  :: val
+     integer(i4knd), intent(out)  :: val
      integer :: rc
      !
      call dm_get(val,1,rc)
@@ -198,7 +198,7 @@
      use dmmp
      implicit none
      !
-     integer(iknd), intent(inout)   :: ptr(:)
+     integer(i4knd), intent(inout)   :: ptr(:)
      integer,       intent(in)      :: nstart,n
      integer :: rc
      !
@@ -211,7 +211,7 @@
      use dmmp
      implicit none
      !
-     integer,    intent(inout)   :: ptr(:,:)
+     integer(i4knd),    intent(inout)   :: ptr(:,:)
      integer,    intent(in)      :: nstart,n
      integer                    :: rc
 #if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFT) || defined(PGF90) || defined
(INTEL)
@@ -301,6 +301,29 @@
      if( rc/=0 ) call expire( 0, 'msg_get_i81', 'dm_get errors -' )
      return
   end subroutine msg_get_i81
+    !----------------------------------------------------------------------
+  subroutine msg_get_i82(ptr,nstart,n)
+     use dmmp
+     implicit none
+     !
+     integer(i8knd),  intent(inout)   :: ptr(:,:)
+     integer,         intent(in)      :: nstart,n
+     integer                          :: rc
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFT) || defined(PGF90) || defined
(INTEL)
+     integer, dimension(size(ptr)) :: tmp
+     if( n<size(ptr) ) then       !preserve remainder of array
+         tmp = reshape(ptr,shape(tmp))
+     else                            !whole array to be overwritten
+         tmp = 0
+     endif
+     call dm_get(tmp(nstart:nstart+n-1),n,rc)
+     ptr = reshape(tmp,shape(ptr))
+#else
+     call dm_get(ptr(nstart:,1),n,rc)
+#endif
+     return
+     if( rc/=0 ) call expire( 0, 'msg_get_i82', 'dm_get errors -' )
+  end subroutine msg_get_i82
   !-------------------------------------------------------------
   subroutine msg_get_c(val)
      use dmmp
@@ -414,7 +437,7 @@
```

```
      use dmmp
      implicit none
      !
-     integer(iknd), intent(in)  :: val
+     integer(i4knd), intent(in)  :: val
      integer :: rc
      !
      call dm_put(val,1,rc)
@@ -426,7 +449,7 @@
      use dmmp
      implicit none
      !
-     integer(iknd), intent(in) :: ptr(:)
+     integer(i4knd), intent(in) :: ptr(:)
      integer,       intent(in) :: nstart,n
      integer :: rc
      call dm_put(ptr(nstart:),n,rc)
@@ -438,7 +461,7 @@
      use dmmp
      implicit none
      !
-     integer,    intent(in)          :: ptr(:,:)
+     integer(i4knd),    intent(in)          :: ptr(:,:)
      integer,    intent(in)          :: nstart,n
      integer                         :: rc
 #if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFT) || defined(PGF90) || defined
(INTEL)
@@ -511,6 +534,24 @@
      if( rc/=0 ) call expire( 0, 'msg_put_i81', 'dm_put errors - ' )
      return
   end subroutine msg_put_i81
+    !--------------------------------------------------------
+  subroutine msg_put_i82(ptr,nstart,n)
+     use dmmp
+     implicit none
+     !
+     integer(i8knd),    intent(in)          :: ptr(:,:)
+     integer,    intent(in)          :: nstart,n
+     integer                         :: rc
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFT) || defined(PGF90) || defined
(INTEL)
+     integer,dimension(size(ptr)) :: tmp
+     tmp = reshape(ptr,shape(tmp))
+     call dm_put(tmp(nstart:nstart+n-1),n,rc)
+#else
+     call dm_put(ptr(nstart:,1),n,rc)
+#endif
+     if( rc/=0 ) call expire( 0, 'msg_put_i82', 'dm_put errors - ' )
+     return
+  end subroutine msg_put_i82
  !------------------------------------------------
   subroutine msg_put_c(val)
      use dmmp
@@ -568,9 +609,9 @@
      use dmmp
      implicit none
      !
-     integer,    intent(inout)   :: ptr(:)
-     integer,    intent(in)      :: n
-     integer                     :: rc
+     integer(i4knd), intent(inout)   :: ptr(:)
+     integer,        intent(in)      :: n
+     integer                         :: rc
      !
      call dm_get(ptr(:),n,rc)
      if( rc/=0 ) call expire( 0, 'msgZ_get_i1', 'dm_get errors -' )
@@ -581,9 +622,9 @@
      use dmmp
      implicit none
      !
-     integer, intent(inout)   :: ptr(:,:)
-     integer,    intent(in)   :: n
-     integer                  :: rc
+     integer(i4knd), intent(inout)   :: ptr(:,:)
+     integer,        intent(in)      :: n
+     integer                         :: rc
 #if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFT) || defined(PGF90) || defined
(INTEL)
      integer, dimension(size(ptr)) :: tmp
      if( n<size(ptr) ) then        !preserve remainder of array
@@ -599,6 +640,42 @@
      if( rc/=0 ) call expire( 0, 'msgZ_get_i2', 'dm_get errors -' )
      return
   end subroutine msgZ_get_i2
+  !-------------------------------------------------
+  subroutine msgZ_get_i81(ptr,n)
+     use dmmp
+     implicit none
+     !
+     integer(i8knd), intent(inout)   :: ptr(:)
+     integer,intent(in)              :: n
+     integer                         :: rc
+     !
+     call dm_get(ptr(:),n,rc)
+     if( rc/=0 ) call expire( 0, 'msgZ_get_i81', 'dm_get errors -' )
+     return
+  end subroutine msgZ_get_i81
+  !--------------------------------------------------------------
+  subroutine msgZ_get_i82(ptr,n)
+     use dmmp
+     implicit none
+     !
+     integer(i8knd), intent(inout)   :: ptr(:,:)
+     integer,intent(in)              :: n
+     integer                         :: rc
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFT) || defined(PGF90) || defined
(INTEL)
+     integer(i8knd), dimension(size(ptr)) :: tmp
+     if( n<size(ptr) ) then        !preserve remainder of array
+         tmp = reshape(ptr,shape(tmp))
+     else                             !whole array to be overwritten
+         tmp = 0
+     endif
+     call dm_get(tmp(:),n,rc)
+     ptr = reshape(tmp,shape(ptr))
+#else
+     call dm_get(ptr(:,1),n,rc)
+#endif
+     if( rc/=0 ) call expire( 0, 'msgZ_get_i82', 'dm_get errors -' )
+     return
+  end subroutine msgZ_get_i82

  !================================================
   subroutine msgZ_put_d1(ptr,n)
@@ -635,9 +712,9 @@
      use dmmp
      implicit none
      !
-     integer,    intent(in)         :: ptr(:)
-     integer,    intent(in)         :: n
-     integer                        :: rc
+     integer(i4knd), intent(in)         :: ptr(:)
+     integer,        intent(in)         :: n
+     integer                            :: rc
      call dm_put(ptr(:),n,rc)
      if( rc/=0 ) call expire( 0, 'msgZ_put_i1', 'dm_put errors - ' )
      return
@@ -648,9 +725,9 @@
      use dmmp
      implicit none
      !
-     integer, intent(in)          :: ptr(:,:)
-     integer,    intent(in)       :: n
-     integer                      :: rc
+     integer(i4knd), intent(in)       :: ptr(:,:)
+     integer,        intent(in)       :: n
+     integer                          :: rc
 #if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFT) || defined(PGF90) || defined
(INTEL)
      integer,dimension(size(ptr)) :: tmp
      tmp = reshape(ptr,shape(tmp))
@@ -661,6 +738,35 @@
      if( rc/=0 ) call expire( 0, 'msgZ_put_i2', 'dm_put errors - ' )
      return
   end subroutine msgZ_put_i2
-
+  ! ----------------------------------------------------
+  subroutine msgZ_put_i81(ptr,n)
+     use dmmp
+     implicit none
+     !
+     integer(i8knd), intent(in)     :: ptr(:)
+     integer, intent(in)            :: n
+     integer                        :: rc
+     call dm_put(ptr(:),n,rc)
+     if( rc/=0 ) call expire( 0, 'msgZ_put_i81', 'dm_put errors - ' )
+     return
+  end subroutine msgZ_put_i81
+  !---------------------------------------------------------
+  subroutine msgZ_put_i82(ptr,n)
+     use dmmp
+     implicit none
+     !
+     integer(i8knd), intent(in)     :: ptr(:,:)
+     integer,intent(in)             :: n
+     integer                        :: rc
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFT) || defined(PGF90) || defined
(INTEL)
+     integer(i8knd),dimension(size(ptr)) :: tmp
+     tmp = reshape(ptr,shape(tmp))
+     call dm_put(tmp(:),n,rc)
+#else
+     call dm_put(ptr(:,1),n,rc)
+#endif
+     if( rc/=0 ) call expire( 0, 'msgZ_put_i82', 'dm_put errors - ' )
+     return
+  end subroutine msgZ_put_i82

 end module messages
diff  -Naurd MCNP5/Source/src/mgcoln.F90 MCNP5_new/Source/src/mgcoln.F90
--- MCNP5/Source/src/mgcoln.F90      2003-04-30 20:11:40.000000000 -0600
+++ MCNP5_new/Source/src/mgcoln.F90  2004-07-22 15:14:42.000000000 -0600
@@ -4,6 +4,7 @@
 subroutine mgcoln
   ! calculate a neutron multigroup forward collision.
   use mcnp_global
+  use dxtran_mod
   use mcnp_debug

   implicit real(dknd) (a-h,o-z)
@@ -29,10 +30,7 @@
   l = jxs(1,iex)-1+nint(xss(jxs(13,iex)+1))+(jg-1)*jgm(2)
   ipt = 2
   ncp = 0
-  idx = 0
-  do i=1,ndx(2)
-    if( (xxx-dxx(2,1,i))**2+(yyy-dxx(2,2,i))**2+(zzz-dxx(2,3,i))**2 < dxx(2,5,i) )
idx=i
-  end do
+  idx = inside_dxtran_sphere()
   vel = slite
   st = totm
   do i=1,n
```

```
diff  -Naurd MCNP5/Source/src/mgimps.F90 MCNP5_new/Source/src/mgimps.F90
--- MCNP5/Source/src/mgimps.F90      2003-04-30 20:11:40.000000000 -0600
+++ MCNP5_new/Source/src/mgimps.F90  2004-07-22 15:14:42.000000000 -0600
@@ -5,6 +5,7 @@
   ! check multigroup tables and calculate multigroup arrays.
   use mcnp_global
   use mcnp_debug
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

@@ -112,7 +113,7 @@
     if( iplt==1 ) then
       do ip = 1,mipt
         do jg = 1,jgm(ip)
-          t = huge
+          t = huge_float
          do ia = 1,mxa
            if( fim(ip,ia)/=0. .and. &
              & gmg(ia+mxa*(jg-1+mgm(ip)))>0. )  t = min(t,gmg(ia+mxa*(jg-1+mgm(ip))))
diff  -Naurd MCNP5/Source/src/mgxspt.F90 MCNP5_new/Source/src/mgxspt.F90
--- MCNP5/Source/src/mgxspt.F90      2003-04-30 20:11:42.000000000 -0600
+++ MCNP5_new/Source/src/mgxspt.F90  2004-07-22 15:14:42.000000000 -0600
@@ -9,6 +9,7 @@
   use mcnp_global
   use mcnp_debug
   use dynamic_arrays
+  use erprnt_mod
   implicit real(dknd) (a-h,o-z)

   character hn*9,ht*10
diff  -Naurd MCNP5/Source/src/msgcon.F90 MCNP5_new/Source/src/msgcon.F90
--- MCNP5/Source/src/msgcon.F90      2003-04-30 20:11:42.000000000 -0600
+++ MCNP5_new/Source/src/msgcon.F90  2004-07-22 15:14:42.000000000 -0600
@@ -14,6 +14,8 @@
   use mcnp_debug
   use messages
   use fmesh_mod, only:fmesh_msgput,fmesh_msgcon
+  use qttyin_mod, only: qttyin
+  use erprnt_mod, only: erprnt, erprnt_i8_k1

   implicit real(dknd) (a-h,o-z)

@@ -22,13 +24,16 @@
   integer ::  mn
   save  np, ns, mn, nc
   real(dknd) :: rg(11)
-  integer,allocatable ::  nc(:,:)
+  integer(i8knd),allocatable ::  nc(:,:)
   character(len=130) :: hf
   character(len=69)  :: hm
   character(len=19)  :: hd
   character(len=6)   :: hs
+
+  integer(i8knd) :: np(2,6) = reshape( (/ (0,i=1,12) /),  (/ 2,6 /) )
+  integer(i8knd) :: ni8knd
+  integer(i8knd) :: n_i8
   integer :: lf = 5
-  integer :: np(2,6) = reshape( (/ (0,i=1,12) /),  (/ 2,6 /) )

   if( .not.mcnp_opt_multp ) then
     ! Just return, if not using MPI or PVM message-passing
@@ -86,18 +91,18 @@
     mn= ltasks
   endif
   allocate (nc(0:mn,1:4))
-  nc=0
+  nc=0_i8knd

   ! Initialize all possible rendezvous points.
   nbal(1:ltasks+1) = 0
   if( nsr/=71 ) then
-    if( prn>0. )  np(1,1) = int(prn)
-    if( dmp>0. )  np(1,2) = int(dmp)
-    if( krtm/=0 .and. freq>0. )  np(1,3) = int(freq)
+    if( prn>0. )  np(1,1) = int(prn,i8knd)
+    if( dmp>0. )  np(1,2) = int(dmp,i8knd)
+    if( krtm/=0 .and. freq>0. )  np(1,3) = int(freq,i8knd)
   endif
-  np(1,5) = 200
+  np(1,5) = 200_i8knd
   np(1,6) = npp
-  if( nsr==6  )  np(1,6) = nps+nint(min(one,1.1*snit)*(niss-nqss))
+  if( nsr==6  )  np(1,6) = nps+nint(min(one,1.1*snit)*(niss-nqss),i8knd)
   if( nsr==71 )  np(1,6) = nps+nsa
   np(2,1:6) = np(1,1:6)

@@ -106,20 +111,20 @@
   ! - if prdmp(5)=0  1/10th of job unless ddg(1,*)>0;
   ! - if prdmp(5)>0  set to prdmp(5).
   ns = 0
-  if( inpd<0 .or. (konrun/=0 .and. inpd==0) )  go to 80
+  if( inpd<0_i8knd .or. (konrun/=0 .and. inpd==0_i8knd) )  go to 80
     if( any( ddg(1,1:ndtt)>0) )  go to 70
     do i = 1,mipt
       if( any( ddx(i,1,1:ndx(i))>0 ) )  go to 70
     enddo
-    if( inpd>0 )  go to 80
-    n = max(0,npp-nps)
-    if( nsr==6  )  n = nint(min(one,snit)*(niss-nqss))
-    if( nsr==71 )  n = nsrck*(kct-kcy)
-    if( npp==0 .and. nsr/=6 .and. nsr/=71 )  ns = 1
-    npd = 1000*max(1,nint(one*n/10000.))
+    if( inpd>0_i8knd )  go to 80
+    n_i8 = max(0_i8knd,npp-nps)
+    if( nsr==6  )  n_i8 = nint(min(one,snit)*(niss-nqss))
+    if( nsr==71 )  n_i8 = nsrck*(kct-kcy)
+    if( npp==0_i8knd .and. nsr/=6 .and. nsr/=71 )  ns = 1
+    npd = 1000_i8knd*max(1_i8knd,nint(one*n_i8/10000._dknd,i8knd))
 70  continue
-    if( inpd>0)  call erprnt(1,2,1,npd,0,0,0,0,&
-      & '  "detector roulette data updated every",i10,  " particles."')
+    if( inpd>0_i8knd)  call erprnt_i8_k1(1,2,1,npd,0,0,0,0,&
+      & '  "detector roulette data updated every",i12,  " particles."')
 80 continue

   ! Send work-exists flag.
@@ -130,13 +135,18 @@
   write(jtty,*) "master sending static commons..."
   call dm_sndi
   call msg_put( gfixcm,  1,  nfixcm  )
-  call msg_put( jfixcm,  1,  lfixcm  )
   call msg_put( i8fixcm, 1,  l8fixcm )
+  call msg_put( jfixcm,  1,  lfixcm  )
+
   call msg_put( gvarcm,  1,  nvarcm  )
+  call msg_put( i8varcm, 1,  l8varcm )
   call msg_put( jvarcm,  1,  lvarcm  )
+
   call msg_put( rdum,    1,  n_rdum  )
   call msg_put( idum,    1,  n_idum  )
+
   call msg_put( gephcm,  1,  nephcm  )
+  call msg_put( i8ephcm, 1,  l8ephcm )
   call msg_put( jephcm,  1,  lephcm  )
   call dm_send(-1,4,i)

@@ -162,6 +172,7 @@
     call dm_sndi
     mr = (lchnk+nrcd-1)/nrcd
     call msg_put( ssb,  1,  nrcd)
+    ! QUESTION should n be i8
```

```
     do n = 2,nrss-nrrs
       read(iusr,end = 600)  (ssb(i),i = 1,nrcd)
       call msg_put( ssb,  1,  nrcd)
@@ -185,22 +196,22 @@

   ! Find the next rendezvous point.
   if( nst/=0  )  return
-  if( ns>0  )  npd = 1000*max(1,nint(ctme*60.*nps/max(one,cts)/10000.))
+  if( ns>0  )  npd =
1000_i8knd*max(1_i8knd,nint(ctme*60.*nps/max(one,cts)/10000._dknd,i8knd))
   if( ntal>0  )  np(1,4) = npd
   if( ns>0 .and. ntal==0 )  np(2,6) = nps+npd
   if( nsr==71 )  np(2,6) = nps+nsa
-  n = max( 100000, np(1,4), np(1,6)-nps )
+  n_i8 = max( 100000_i8knd, np(1,4), np(1,6)-nps )
   do i = 1,6
-    if( i<=4 .and. np(1,i)>0 .and. nps>=np(2,i) ) then
-      np(2,i) = np(2,i)+np(1,i)*((nps-np(2,i))/np(1,i)+1)
+    if( i<=4 .and. np(1,i)>0_i8knd .and. nps>=np(2,i) ) then
+      np(2,i) = np(2,i)+np(1,i)*((nps-np(2,i))/np(1,i)+1_i8knd)
     endif
-    if( nps<np(2,i) )  n = min(n,np(2,i)-nps)
+    if( nps<np(2,i) )  n_i8 = min(n_i8,np(2,i)-nps)
   enddo

   ! Determine the number of microtasks.
-  if( n<mf*ltasks .or. jtasks<0 ) then
+  if( n_i8<int(mf*ltasks,i8knd) .or. jtasks<0 ) then
     nm = ltasks
-  elseif( n<lf*ltasks ) then
+  elseif( n_i8<int(lf*ltasks,i8knd) ) then
     nm = min(mf*ltasks,mn)
   else
     nm = min(lf*ltasks,mn)
@@ -208,16 +219,16 @@

   ! Set nps ranges for each microtask.
   do i = 1,nm
-    nc(i,1) = nps+int(float(n*i)/nm)
-    nc(i,2) = 0
-    nc(i,3) = 0
+    nc(i,1) = nps+int(real(n_i8*int(i,i8knd),dknd)/nm,i8knd)
+    nc(i,2) = 0_i8knd
+    nc(i,3) = 0_i8knd
   enddo
   nc(0,1)  = nps
-  nc(nm,1) = nps+n
+  nc(nm,1) = nps+n_i8
   call getidt(hd)
-  write(jtty,'(" master set rendezvous nps =",i10,5x,a19)') nps+n, hd
+  write(jtty,'(" master set rendezvous nps =",i12,3x,a19)') nps+n_i8, hd
   if( mct>=0 )&
-   & write(iuo,'(" master set rendezvous nps =",i10,5x,a19)') nps+n, hd
+   & write(iuo,'(" master set rendezvous nps =",i12,3x,a19)') nps+n_i8, hd


   ! Send nps ranges for all microtasks.
@@ -236,6 +247,7 @@
   call msg_put(  ddn,   1,  24*ndnd)
   call msg_put(  dxd,   1,  mipt*24*mxdx)
   call msg_put( gvarcm, 1,  nvarcm)
+  call msg_put( i8varcm,1, l8varcm)
   call msg_put( jvarcm, 1,  lvarcm)
   call msg_put( nbal,   1,  size(nbal))
   call msg_put( rdum,   1,  n_rdum  )
@@ -260,7 +272,7 @@
   ! Clear the interrupt and check for an available host.
 200 continue
   if( irup/=0 ) then
-    call qttyin( max(1,nc(nk,1)),'  "dmmp microtask assignment, nps =",i9' )
+    call qttyin( max(1_i8knd,nc(nk,1)),'  "dmmp microtask assignment, nps =",i12' )
```

```
   endif

   ! Fault tolerance for PVM only.
@@ -363,6 +375,7 @@
     call dm_recv(-2,4,i)
     call dm_rcvi(i,nh)
     call msg_get( gtskcm,  1,  nvarsw)
+    call msg_get( i8tskcm,  1, l8varsw)
     call msg_get( jtskcm,  1,  lvarsw)
     call msg_get( ktskpt,  1, ltskpt-1)

@@ -385,13 +398,13 @@
     call fmesh_msgcon

     ! Receive output variables not updated in vtask.
-    call msg_get( n )
-    if( n/=0 ) then
+    call msg_get( ni8knd )
+    if( ni8knd/=0 ) then
        call msg_get( m )
        call msg_get( hs )
        call msg_get( hm )
-      if( kdbnps==0 ) then
-        kdbnps = n
+      if( kdbnps==0_i8knd ) then
+        kdbnps = ni8knd
         mmkdb  = m
         hsub   = hs
         hmes   = hm
@@ -435,10 +448,10 @@
                          nkrp = nkrp+n-m5
     call msg_get( n )
                          jtlx = jtlx+n-m6
-    call msg_get( n )
-                         nqsw = nqsw+n
-    call msg_get( n )
-                         nrsw = nrsw+n
+    call msg_get( ni8knd )
+                         nqsw = nqsw+ni8knd
+    call msg_get( ni8knd )
+                         nrsw = nrsw+ni8knd
     do k = 1,njss+nilw
       do j = 1,2+4*mipt
         call msg_get( n )
@@ -461,12 +474,12 @@
     enddo
     do j = 1,ndnd
       call msg_get( a )
-                         ddn(22,j) = ddn(22,j)+dble(nint(a))
+                         ddn(22,j) = ddn(22,j)+real(nint(a),dknd)
     enddo
     do k = 1,mxdx*min(1,nxnx)
       do j = 1,mipt
         call msg_get( a )
-                         dxd(j,22,k) = dxd(j,22,k)+dble(nint(a))
+                         dxd(j,22,k) = dxd(j,22,k)+real(nint(a),dknd)
       enddo
     enddo

@@ -504,7 +517,7 @@
   enddo DO_360

   ! Step through the microtasks in order - to preserve tracking.
-  nc(0,2) = 0
+  nc(0,2) = 0_i8knd
   DO_430: do mt = 1,nm
     nh       = nc(mt,2)
     nc(nh,3) = nc(nh,3)+1
@@ -572,16 +585,16 @@
     nn = nm
     nx = 1
     do nt = 1,ltasks
-      nn = min(nn,nc(nt,3))
-      nx = max(nx,nc(nt,3))
+      nn = min(int(nn,i8knd),nc(nt,3))
+      nx = max(int(nx,i8knd),nc(nt,3))
     enddo
-    lf = nint((float(nx)/float(max(1,nn)))**3)
+    lf = nint((real(nx)/real(max(1,nn)))**3)
     lf = min(lf,30)
     lf = max(lf,mf)
   endif

   ! Undo nst=4 and reset appropriately.
-  if( (nps<npp .or. npp==0) .and. mod(nst/4,2)/=0 )  nst = nst-4
+  if( (nps<npp .or. npp==0_i8knd) .and. mod(nst/4,2)/=0 )  nst = nst-4
   ! check for max time used
   if( ctme/=0.0 .and. cts>=ctme*60. )  nst = nst + 32
   if( krtm/=0 .and. nst/=0 )  mrm = 1
diff  -Naurd MCNP5/Source/src/msgtsk.F90 MCNP5_new/Source/src/msgtsk.F90
--- MCNP5/Source/src/msgtsk.F90      2003-11-05 17:23:12.000000000 -0700
+++ MCNP5_new/Source/src/msgtsk.F90  2004-07-22 15:14:42.000000000 -0600
@@ -20,8 +20,8 @@

   integer            ::  mn
   real(dknd)         ::  rg(11)
-  integer,allocatable ::  nc(:,:)
-  integer(i8knd)     ::  i8a, i8b, i8c, i8d
+  integer(i8knd),allocatable ::  nc(:,:)
+  integer(i8knd)     ::  i8a, i8b, i8c, i8d, i8i
   character(len=130) ::  hf
   logical            ::  file_open
 #ifdef AIX
@@ -46,13 +46,18 @@
   call dm_recv(-1,4,i)
   call secnd(t1)
   call msg_get( gfixcm,  1, nfixcm  )
-  call msg_get( jfixcm,  1, lfixcm  )
   call msg_get( i8fixcm, 1, l8fixcm )
+  call msg_get( jfixcm,  1, lfixcm  )
+
   call msg_get( gvarcm,  1, nvarcm  )
+  call msg_get(i8varcm,  1, l8varcm )
   call msg_get( jvarcm,  1, lvarcm  )
+
   call msg_get( rdum,    1, n_rdum  )
   call msg_get( idum,    1, n_idum  )
+
   call msg_get( gephcm,  1, nephcm  )
+  call msg_get(i8ephcm,  1, l8ephcm )
   call msg_get( jephcm,  1, lephcm  )

   if(jtasks > 0) then
@@ -96,7 +101,7 @@
     mt = 1
   endif
   allocate( gbnk( 1:mbnk*mtasks ) )
-  allocate( ibnk( 1:(nbmx*(lpblcm+2*abs(iunr))+1)*mtasks ) )
+  allocate( ibnk( 0:(nbmx*(lpblcm+2*abs(iunr))+1)*mtasks ) )
  allocate( tal(  1:(nmxf*mxf+ktls)*mt ) )
   gbnk = 0.0
   ibnk = 0
@@ -169,6 +174,7 @@
   call msg_get(  ddn,    1, 24*ndnd)
   call msg_get(  dxd,    1, mipt*24*mxdx)
   call msg_get( gvarcm,  1, nvarcm )
+  call msg_get( i8varcm, 1, l8varcm )
   call msg_get( jvarcm,  1, lvarcm )
   call msg_get( nbal,    1, size(nbal))
   call msg_get( rdum,    1, n_rdum  )
@@ -203,9 +209,10 @@
   endif
   ! Clear the swept part of varcom.
-  gvarcm(1:nvarsw) = 0.
-  jvarcm(1:lvarsw) = 0
-  wgts(1) = huge
+  gvarcm(1:nvarsw)   = 0.
+  i8varcm(1:l8varsw) = 0_i8knd
+  jvarcm(1:lvarsw)   = 0
+  wgts(1) = huge_float

   call RN_init_problem( new_standard_gen = RN_gen_input,     &
     &                   new_seed         = RN_seed_input,    &
@@ -234,8 +241,8 @@
       if( mt==nm )  kbp = 0
     endif
     ! Skip over starters from previous chunks.
-    do i = nps+1,nc(mt-1,1)
-      nps  = nps+1
+    do i8i = nps+1_i8knd,nc(mt-1,1)
+      nps  = nps+1_i8knd
       call RN_init_particle( int(nps,i8knd) )
       if( nsr==6 ) then
         call sursrc(1)
@@ -323,6 +330,7 @@
   ! Send task common data stored in varcom.
   call dm_sndi
   call msg_put( gvarcm, 1, nvarsw)
+  call msg_put( i8varcm, 1, l8varsw)
   call msg_put( jvarcm, 1, lvarsw)
   call msg_put( ktskpt, 1, ltskpt-1)
   call dm_send(0,4,i)
diff  -Naurd MCNP5/Source/src/namrsd.F90 MCNP5_new/Source/src/namrsd.F90
--- MCNP5/Source/src/namrsd.F90      2003-04-30 20:11:44.000000000 -0600
+++ MCNP5_new/Source/src/namrsd.F90  2004-07-22 15:14:42.000000000 -0600
@@ -15,6 +15,7 @@
   ! Modules used:
   use mcnp_global
   use mcnp_debug
+  use erprnt_mod
   implicit real(dknd) (a-h,o-z)
   character(len=*)   :: hh
   character(len=120) :: hs
diff  -Naurd MCNP5/Source/src/newcd1.F90 MCNP5_new/Source/src/newcd1.F90
--- MCNP5/Source/src/newcd1.F90      2003-04-30 20:11:46.000000000 -0600
+++ MCNP5_new/Source/src/newcd1.F90  2004-07-22 15:14:42.000000000 -0600
@@ -10,6 +10,8 @@
   use mcnp_debug
   use mcnp_input
   use dynamic_arrays
+  use qttyin_mod, only: qttyin
+  use erprnt_mod, only: erprnt

  implicit real(dknd) (a-h,o-z)

@@ -76,9 +78,6 @@
   return
 100 continue
  ndup(min(ica,3)) = ndup(min(ica,3))+1
-#ifdef PCDOS
-  if(lockl)  call pttyin
-#endif /*def.pcdos*/
  if(irup /= 0) call qttyin(0,'"reading input cards.     '//ich//'"')

  ! Do the processing necessary for the particular card type.
@@ -124,8 +123,9 @@

      if((ich == 'de'.or.ich == 'df').and.icn/=0) return
      if (ich == 'fm') return  !The 'fm' card is also used by the mesh tallies
-
+
      if( ich=='dd' .and. (icn==1 .or. icn==2) )  return
+
       i = icn+1000*nqw
       do ital_tmp=0,ntal
         ital = ital_tmp
@@ -209,6 +209,17 @@
     case (96)  !===> third spare card type                          zc
       continue

+    case (102) !===> force use of lattice speed tally modifications  spdtl
+      call nxtsym(hlin,' =',1,istart,ifinish,1)            ! Find card name
+      call nxtsym(hlin,' =',ifinish+1,istart,ifinish,1) ! Find keyword
+      if  ( ifinish == 0 ) call erprnt(2,1,0,0,0,0,0,0,&
+        & '"spdtl card must have exactly one keyword, either force or off."')
+      if ( hlin(istart:ifinish) /= 'force' .and. hlin(istart:ifinish) /= 'off'  ) &
+        & call erprnt(2,1,0,0,0,0,0,0,'"spdtl card must have exactly one keyword, either
force or off."')
+      if ( hlin(istart:ifinish) == 'force' ) flag_speed_tally_force = 1
+      if ( hlin(istart:ifinish) == 'off' ) flag_speed_tally_force = -1
+      continue
+
     case default  !===> default - do nothing
       continue

diff  -Naurd MCNP5/Source/src/newcrd.F90 MCNP5_new/Source/src/newcrd.F90
--- MCNP5/Source/src/newcrd.F90       2003-04-30 20:11:46.000000000 -0600
+++ MCNP5_new/Source/src/newcrd.F90   2004-07-22 15:14:42.000000000 -0600
@@ -8,6 +8,8 @@
   use mcnp_debug
   use mcnp_input
   use fmesh_mod
+  use qttyin_mod, only: qttyin
+  use erprnt_mod, only: erprnt

   implicit real(dknd) (a-h,o-z)
   character(len=8) :: ht
@@ -87,9 +89,6 @@
   call erprnt(2,1,0,0,0,0,0,1,'"'//ich//' is not a legal data symbol."')
   return
 80 continue
-#ifdef PCDOS
-  if(lockl)call pttyin
-#endif
   if( irup/=0 )  call qttyin(0,'"reading input cards.     "'//ich)
   if( krq(2,ica)/=0 .and. nqw==0 ) then
     call erprnt(2,1,0,0,0,0,0,1,'  "particle-type designator is missing."')
@@ -189,6 +188,9 @@
           & '"dxtran cell probabilities without '//hnp(i)//' dxtran spheres."')
         endif
       end do
+      if ( flag_speed_tally_ok == 1 ) flag_speed_tally_ok= -1
+      if ( flag_speed_tally_force == 1 )  call erprnt(2,3,0,0,0,0,0,0,&
+        & '"lattice speed tally conflicts with dxc card."')
       go to 9010

     case( 17 )
@@ -259,7 +261,45 @@
         return
       endif
 350   continue
-      !
+
+      ! Checks for lattice speed tally criteria
+      ! Print warning about conflict if spdtl force card is used
+      iy = mod(icn,10)
+      if ((ich=='ft' .or. ich=='cf' .or. ich=='sf' .or. ich=='t'  .or. &
+        &  ich=='e'  .or. ich=='tm' .or. ich=='cm' .or. ich=='em' .or. &
+        &  ich=='sf' .or. ich=='c'  .or. ich=='fs')  .and. iy/=5 ) then
+        if ( flag_speed_tally_ok == 1 ) flag_speed_tally_ok = -1
+        if ( flag_speed_tally_force == 1 )  call erprnt(2,3,1,icn,0,0,0,0,&
+          & '"lattice speed tally conflicts with '//ich(1:3)//' card for tally",i4,"."')
+      endif
```

```
+
+      if ( ich == 'f' ) then
+        ! Set flags for fm and de,df cards.
+        flag_speed_tally_fm(ital,2)=icn
+        flag_speed_tally_de(ital,2)=icn
+        ! Exclude all tallies except F4 for lattice speed tally
+        if ( iy==1 .or. iy==2 .or. iy==3 .or. &
+           & iy==6 .or. iy==7 .or. iy==8) then
+          if ( flag_speed_tally_ok == 1 ) flag_speed_tally_ok = -1
+          if ( flag_speed_tally_force == 1 )  call erprnt(2,3,1,icn,0,0,0,0,&
+          & '"lattice speed tally conflicts with tally",i4,"."')
+        endif
+        if ( iy==4 ) then
+          ! exclude *F4  tallies for lattice speed tally
+          if ( icx /= 0 ) then
+            if ( flag_speed_tally_ok == 1) flag_speed_tally_ok = -1
+            if ( flag_speed_tally_force == 1 )  call erprnt(2,3,1,icn,0,0,0,0,&
+              & '"lattice speed tally conflicts with *f4 tally",i4,"."')
+          endif
+        ! check for [] on f4 tally line for lattice speed tally
+        i = 0
+        i = index(klin,'[')
+        if ( i  <= 3 .and. flag_speed_tally_ok == 1) flag_speed_tally_ok = -1
+        if ( i  <= 3 .and. flag_speed_tally_force == 1 )  &
+          & call erprnt(2,3,1,icn,0,0,0,0,&
+          & '"lattice speed tally needs lattice index range [] for tally",i4,"."')
+        endif
+      endif      !
       if( ich=='fq' ) then
         do i=1,8
           jfq(i,ital) = i
@@ -355,6 +395,7 @@
       !  >>>>>   tally multiplier                                          fm
       !  m3c = 0   tally multiplier card
       !      = 1   mesh tally  multiplier card
+
       m3c=1
       ital=0
       do ital_tmp=1,ntal
@@ -363,7 +404,7 @@
           exit
         endif
       enddo
-
+
       if(m3c==1) then
         nfm=nfm+1
         meshfm(nfm)=icn
@@ -376,6 +417,8 @@
         ipnt(1,krq(3,ica),ital) = ipl+1
       endif

+      ! Set flag for spdtl that de df cards for tally ital exist
+      if (m3c == 0) flag_speed_tally_fm(ital,1)=icn

     case( 35,36)
       !  >>>>>  response functions                                       de,df
@@ -406,8 +449,20 @@
         ipnt(1,krq(3,ica),ital) = ipl+1
       endif

+      ! Set flag for spdtl that de df cards for tally ital exist
+      if( ital==0 ) then
+        flag_speed_tally_de(ital,1)=1000  ! flag for fm0 cards
+      else
+        flag_speed_tally_de(ital,1)=icn
+      endif
+
     case( 47 )
       !  >>>>>  dxtran parameters                                           dxt
+
+      if ( flag_speed_tally_ok == 1 ) flag_speed_tally_ok= -1
+      if ( flag_speed_tally_force == 1 )  call erprnt(2,3,0,0,0,0,0,0,&
+        & '"lattice speed tally conflicts with dxt card."')
+
       if( nqp(3)/=0 ) then
         call erprnt(2,1,0,0,0,0,0,1,&
           & '  "dxtran is not available for electrons."')
@@ -467,9 +522,19 @@
         call erprnt(2,1,0,0,0,0,0,0,&
           & '  "ksrc and sdef cards are incompatible."')
       endif
+
+    case( 71 )
+      !  >>>>>  weight-window generator parameters                          wwg
+      if ( flag_speed_tally_ok == 1 ) flag_speed_tally_ok = -1
+      if ( flag_speed_tally_force == 1 )  call erprnt(2,3,0,0,0,0,0,0,&
+        & '"lattice speed tally conflicts with wwg card."')

     case( 72 )
       !  >>>>>  energy bins for weight-window generator                    wwge
+      if ( flag_speed_tally_ok == 1 ) flag_speed_tally_ok = -1
+      if ( flag_speed_tally_force == 1 )  call erprnt(2,3,0,0,0,0,0,0,&
+        & '"lattice speed tally conflicts with wwge card."')
+
       if( iwwg==0 ) then
         call erprnt(2,1,0,0,0,0,0,0,&
           & '"'//ich//'card without any wwg card."')
@@ -537,6 +602,9 @@
         call erprnt(2,2,0,0,0,0,0,0,&
           & '  "this perturbation is not consistent with the problem mode."')
       endif
+      if ( flag_speed_tally_ok == 1 ) flag_speed_tally_ok = -1
+      if ( flag_speed_tally_force == 1 )  call erprnt(2,3,0,0,0,0,0,0,&
+        & '"lattice speed tally conflicts with pert card."')

     case( 86 )
       !  >>>>>  superimposed weight window generator mesh                  mesh
diff  -Naurd MCNP5/Source/src/nextit.F90 MCNP5_new/Source/src/nextit.F90
--- MCNP5/Source/src/nextit.F90      2003-04-30 20:11:48.000000000 -0600
+++ MCNP5_new/Source/src/nextit.F90  2004-07-22 15:14:42.000000000 -0600
@@ -8,6 +8,7 @@
   use mcnp_debug
   use mcnp_input
   use fmesh_mod
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)
   character(len=75) :: ht
@@ -475,7 +476,7 @@
      !  >>>>>  tally multiplier                                             fm
      !  m3c = 0   tally multiplier card
      !      = 1   mesh tally  multiplier card
-           ks = index('():#<[],u=',hitm(1:1))
+     ks = index('():#<[],u=',hitm(1:1))
      if( m3c==1 ) then
         if( kitm/=0 ) then
        fmtmp(nwc,nfm) = ritm
@@ -489,6 +490,14 @@
            if( ks/=0 )  rtp(ipl+nwc) = 1000000+ks
        endif
     endif
+     ! Disable lattice speed tally if more than one entry on fm card.
+     if( mod(icn,10) == 4 .and. m3c == 0 ) then
+       if( nwc > 1 .and. flag_speed_tally_ok == 1 ) flag_speed_tally_ok = -1
+       if( nwc > 1 .and. flag_speed_tally_force == 1 ) call erprnt(2,3,1,icn,0, &
+         & 0,0,0, &
+         & '"lattice speed tally conflicts with multiple entries on fm",i4,", card."')
+     endif
+

   case( 35,36 )
```

```
      !  >>>>>  response function                                         de,df
@@ -513,8 +522,10 @@
               else
                  intrpol(ndf) = intrpol(ndf)+ 1
               endif
+              nwc = nwc-1
            endif
            if( m2c==0 .or. icn==0 ) then
+              if ( m2c == 1 ) nwc = nwc+1
               rtp(ipl+nwc) = -1.
            endif
         else if(hitm=='log')then
@@ -540,7 +551,7 @@
         return
       endif
       do k = 1,mipt
-        if( ddx(k,i,j)==huge )  ddx(k,i,j) = ritm
+        if( ddx(k,i,j)==huge_float )  ddx(k,i,j) = ritm
       enddo
     endif
     if( kitm/=0 ) then
@@ -757,16 +768,16 @@
     if( nwc==5 .and. ritm==0.)  swtm = -1.
     if( nqp(1)/=0 .and. nwc==1 .and. ritm/=0. ) then
       do i = 2,mipt
-        if( tco(i)==.001d0*huge)  tco(i) = tco(1)
+        if( tco(i)==.001d0*huge_float)  tco(i) = tco(1)
       enddo
     endif

   case( 60 )
     !  >>>>>  source particle cutoff number                              nps
-    if( nwc==1 )  npp = iitm
-    if( nwc==2 .and. (konrun==0 .or. iitm/=0) )  npsmg = iitm
-    if( konrun/=0 .and. nwc==2 .and. iitm>nps )&
-      & call erprnt(2,2,1,npsmg-nps,0,0,0,0,&
+    if( nwc==1 )  npp = i8itm
+    if( nwc==2 .and. (konrun==0 .or. i8itm/=0_i8knd) )  npsmg = i8itm
+    if( konrun/=0 .and. nwc==2 .and. i8itm>nps )&
+      & call erprnt_i8_k1(2,2,1,npsmg-nps,0,0,0,0,&
       & ' "adding ",i5,  " more source contributions to image."')

   case( 61 )
@@ -787,8 +798,8 @@
     if( nwc==2 )  dmp = ritm
     if( nwc==3 )  mct = iitm
     if( nwc==4 )  ndmp = iitm
-    if( nwc==5 .and. iitm>0 )  npd = iitm
-    if( nwc==5 .and. mcnp_opt_multp )  inpd = iitm
+    if( nwc==5 .and. i8itm>0_i8knd )  npd = i8itm
+    if( nwc==5 .and. mcnp_opt_multp )  inpd = i8itm

   case( 65 )
     !  >>>>>  termination and print control for lost particles          lost
@@ -973,7 +984,7 @@
     case( 3 )
       if( nwc==-m2c+2 ) then
        ktr(m1c) = trf(2,mxtr)
-       trf(2,mxtr) = huge
+       trf(2,mxtr) = huge_float
        mxtr = mxtr-1
      else
        call trfmat(mxtr)
@@ -1050,7 +1061,7 @@
      if( nwc==-m2c+2 ) then
        if( m3c==0)  mfl(3,m1c) = trf(2,mxtr)
        if( m3c/=0)  laf(3+mlaf+3,m3c) = trf(2,mxtr)
-       trf(2,mxtr) = huge
+       trf(2,mxtr) = huge_float
        mxtr = mxtr-1
      else
```

```
         call trfmat(mxtr)
@@ -1118,6 +1129,8 @@
         ptr(iptra(m1c)) = m3c
         if( kitm==1)  ptr(m2c) = ritm
         if( kitm==2)  ptr(m2c) = iitm
+        if( kitm==2 .and. hptr(m1c)=='nps' ) ptr(m2c) = ritm
+        if( kitm==2 .and. hptr(m1c)=='max' ) ptr(m2c) = ritm
       endif
     else
       if(index(',=',hitm(1:1))/=0)  return
@@ -1448,6 +1461,13 @@
     !  >>>>>  real quantities for Recursive Monte Carlo                   rrmc
     rrmc(nwc) = ritm

+  case( 102 )
+    !  >>>>>  force lattice speed tally modifications (override appropriate) spdtl
+    if ( flag_speed_tally_ok == 0 )   &
+      & call erprnt(2,2,0,0,0,0,0,0,'"spdtl card present, but no lattice. spdtl
ignored."')
+    flag_speed_tally_used = -1
+    continue
+
   end select
   return
 end subroutine nextit
diff  -Naurd MCNP5/Source/src/norma.F90 MCNP5_new/Source/src/norma.F90
--- MCNP5/Source/src/norma.F90        2003-04-30 20:11:48.000000000 -0600
+++ MCNP5_new/Source/src/norma.F90    2004-07-22 15:14:42.000000000 -0600
@@ -6,6 +6,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/normh.F90 MCNP5_new/Source/src/normh.F90
--- MCNP5/Source/src/normh.F90        2003-04-30 20:11:48.000000000 -0600
+++ MCNP5_new/Source/src/normh.F90    2004-07-22 15:14:42.000000000 -0600
@@ -10,6 +10,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod
   implicit real(dknd) (a-h,o-z)

   ! Trim the distribution if it is an energy distribution.
diff  -Naurd MCNP5/Source/src/nsourc.F90 MCNP5_new/Source/src/nsourc.F90
--- MCNP5/Source/src/nsourc.F90       2003-04-30 20:11:50.000000000 -0600
+++ MCNP5_new/Source/src/nsourc.F90   2004-07-22 15:14:42.000000000 -0600
@@ -24,6 +24,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit double precision (a-h,o-z)
   character(len=80) :: hh
diff  -Naurd MCNP5/Source/src/nxtit1.F90 MCNP5_new/Source/src/nxtit1.F90
--- MCNP5/Source/src/nxtit1.F90       2003-04-30 20:11:50.000000000 -0600
+++ MCNP5_new/Source/src/nxtit1.F90   2004-07-22 15:14:42.000000000 -0600
@@ -471,6 +471,8 @@
   case( 77 )
     !  >>>>>  lattice type                                                 lat
    if( iitm/=0 )  nlat = nlat+1
+    ! Enable lattice speed tally modifications for hexahedra lattices
+    if( iitm==1 )  flag_speed_tally_ok = 1

   case( 78 )
     !  >>>>>  cell-filling universes, with transformations                fill
diff  -Naurd MCNP5/Source/src/oldcd1.F90 MCNP5_new/Source/src/oldcd1.F90
```

```
--- MCNP5/Source/src/oldcd1.F90      2003-04-30 20:11:52.000000000 -0600
+++ MCNP5_new/Source/src/oldcd1.F90  2004-07-22 15:14:42.000000000 -0600
@@ -14,7 +14,7 @@
   ! space needed for input coefficients for each surface type.
   integer :: ns(39) = &
     & (/ 4, 1, 1, 1, 1, 4, 2, 2, 2, 3, 3, 3, 1, 1, 1, 5, 5, 5, 3, 3, &
-    &    3,10,10, 7, 7, 7, 0, 0, 0,24,15, 4,18,18,10,18,20,24,32 /)
+    &    3,10,10, 7, 7, 7, 0, 0, 0,24,24, 4,18,18,10,18,20,24,32 /)

   select case( ica )

diff  -Naurd MCNP5/Source/src/oldcrd.F90 MCNP5_new/Source/src/oldcrd.F90
--- MCNP5/Source/src/oldcrd.F90      2003-04-30 20:11:52.000000000 -0600
+++ MCNP5_new/Source/src/oldcrd.F90  2004-07-22 15:14:42.000000000 -0600
@@ -8,8 +8,10 @@
   use mcnp_debug
   use mcnp_input
   use fmesh_mod
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)
+  integer(i8knd) :: k8

   ! correct number of input coefficients for each surface type.
   integer :: ns(39) = &
@@ -195,7 +197,6 @@
     !  >>>>>  cell volumes for tallies                                   vol
     if( novol==0 .or. nwc/=0 )  go to 500
     return
-
   case( 8 )
     !  >>>>>  surface areas for tallies                                 area
     if( nwc/=mxj )  call erprnt(2,2,2,nwc,mxj,0,0,0,&
@@ -439,7 +440,7 @@
     !  >>>>>  various cell data          vol,pwt,ext,fcl,wwn,pd,dxc,tmp,u,lat
     if( nwc/=mxa )  call erprnt(2,2,2,nwc,mxa,0,0,0,&
       & 'i5,  " entries not equal to number of cells =",i5, "."')
-    return
+        return

   case( 72 )
    !  >>>>>  energy bins for weight-window generator                   wwge
@@ -553,9 +554,9 @@
       cycle DO_700
 660   continue
       if( n>1 )  go to 650
-      k = nint(ptr(j+1))
-      if(hptr(i) == 'max'.and.k < 0)iptr=-iptr
-      iptra(i) = abs(k)
+      k8 = nint(ptr(j+1),i8knd)
+      if(hptr(i) == 'max'.and. k8 < 0)iptr=-iptr
+      iptra(i) = abs(k8)
       cycle DO_700
 670   continue
       if( n>2 )  go to 650
@@ -840,6 +841,7 @@
   end select
   return

+
   ! a common complaint
 9010 continue
   call erprnt(2,1,0,0,0,0,0,0,'  "entries are not monotonically increasing."')
diff  -Naurd MCNP5/Source/src/output.F90 MCNP5_new/Source/src/output.F90
--- MCNP5/Source/src/output.F90      2003-04-30 20:11:52.000000000 -0600
+++ MCNP5_new/Source/src/output.F90  2004-07-22 15:14:42.000000000 -0600
@@ -10,6 +10,7 @@
   use mcnp_debug
   use mcplot_module
   use crit1_mod
+  use qttyin_mod, only:qttyin
```

```
   use ra2_mod

   implicit real(dknd) (a-h,o-z)
@@ -22,12 +23,12 @@

   ! Keyboard interrupt.
   if( lockl   ) call pttyin
-  if( irup/=0 ) call qttyin(nps,'  "output rendezvous, nps =",i9')
+  if( irup/=0 ) call qttyin(nps,'  "output rendezvous, nps =",i12')


-  if( ntal/=0 .and. npp>=0 )  then
+  if( ntal/=0 .and. npp>=0_i8knd )  then
     ! Update detector and dxtran russian roulette criteria.
-    if( nps==200 .or. jtfc/=0 )  then
+    if( nps==200_i8knd .or. jtfc/=0 )  then
       ddn(24,1:ndnd) = ddn(20,1:ndnd)/nps
       dxd(1:mipt,24,1:mxdx) = dxd(1:mipt,20,1:mxdx)/nps
     endif
@@ -91,14 +92,14 @@
    endif

   ! Write restart (runtpe) if required.
-  if( mdc+nst/=0 .and. npp>=0 ) then
+  if( mdc+nst/=0 .and. npp>=0_i8knd ) then
     call tpefil(4)
     tdc = cp1
     if( issw/=0 .and. nst/=0 )  call wrwssa
   endif

   ! Write mctal file if required.
-  if( mct/=0 .and. (nst/=0 .or. npp<0) )  then
+  if( mct/=0 .and. (nst/=0 .or. npp<0_i8knd) )  then
     hf = mctal
     call mctalw(hf)
     mctal = hf
diff  -Naurd MCNP5/Source/src/pconst.F90 MCNP5_new/Source/src/pconst.F90
--- MCNP5/Source/src/pconst.F90      2003-04-30 20:11:56.000000000 -0600
+++ MCNP5_new/Source/src/pconst.F90  2004-07-22 15:14:42.000000000 -0600
@@ -17,7 +17,7 @@
   ! Write physical contstants table.
   write(iuo,'(  "1physical constants",85x,  "print table 98",2/)')
   write(iuo,'(3x, "name",20x, "value",5x,  "description"/)')
-  write(iuo,'(3x, "huge",1p1e25.13,5x, "infinity")')huge
+  write(iuo,'(3x, "huge",1p1e25.13,5x, "infinity")')huge_float
   write(iuo,'(4x, "pie",1p1e25.13,5x, "pi")')pie
   write(iuo,'(2x, "euler",1p1e25.13,5x,  "euler constant")')euler
   write(iuo,'(1x, "avogad",1p1e25.13,5x,&
@@ -66,21 +66,9 @@
 #ifdef MCPLOT
   write(iuo,'(5x, "mcplot")')
 #endif /*def.mcplot*/
-#ifdef GKSSIM
-  write(iuo,'(5x, "gkssim")')
-#endif /*def.gkssim*/
 #ifdef XLIB
   write(iuo,'(5x, "xlib")')
 #endif /*def.xlib*/
-#ifdef PCDOS
-  write(iuo,'(5x, "pcdos")')
-#endif /*def.pcdos*/
-#ifdef LAHEY
-  write(iuo,'(5x, "lahey")')
-#endif /*def.lahey*/
-#ifdef QWIN
-  write(iuo,'( "qwin")')
-#endif /*def.qwin*/
 #ifdef LINUX
   write(iuo,'(5x, "linux")')
 #endif /*def.linux*/
diff  -Naurd MCNP5/Source/src/plotg.F90 MCNP5_new/Source/src/plotg.F90
```

```
--- MCNP5/Source/src/plotg.F90        2003-04-30 20:12:00.000000000 -0600
+++ MCNP5_new/Source/src/plotg.F90    2004-07-22 15:14:42.000000000 -0600
@@ -165,15 +165,9 @@
     if( jvp==0) sl = xhom
     if( jvp==0) tl = yhom
 #endif
-#ifdef LAHEY
-    call WindowOutString(100,5000,hl)
-#else
     call gtx(sl,tl,hl)
-#endif
     if( mcolor>2) call gstxci(kcolor(ncolor+1),0)
-#ifndef LAHEY
     call gxhome(real(xhom),real(yhom))
-#endif
     go to 60
   endif

diff  -Naurd MCNP5/Source/src/pltdat.F90 MCNP5_new/Source/src/pltdat.F90
--- MCNP5/Source/src/pltdat.F90       2003-04-30 20:12:00.000000000 -0600
+++ MCNP5_new/Source/src/pltdat.F90   2004-07-22 15:14:42.000000000 -0600
@@ -133,9 +133,9 @@
       if( nv<=3 )  return
       f = 0.
       if( nv==4 )  then
-        f = 1./max(nps,1)
-        if( nsr==6 .and. nrrs>=nrss )  f = 1./max(np1,1)
-        if( nsr==6 .and. nrrs<nrss  )  f = 1./max(npsr,1)
+        f = 1./max(nps,1_i8knd)
+        if( nsr==6 .and. nrrs>=nrss )  f = 1./max(np1,1_i8knd)
+        if( nsr==6 .and. nrrs<nrss  )  f = 1./max(npsr,1_i8knd)
       endif
       if( nv==5 .and. pac(ip,9,iap)/=0.  )  f = 1./pac(ip,9,iap)
       if( nv==6 .and. pac(ip,10,iap)/=0. )  f = 1./pac(ip,10,iap)
diff  -Naurd MCNP5/Source/src/pltmsh.F90 MCNP5_new/Source/src/pltmsh.F90
--- MCNP5/Source/src/pltmsh.F90       2003-04-30 20:12:02.000000000 -0600
+++ MCNP5_new/Source/src/pltmsh.F90   2004-07-22 15:14:42.000000000 -0600
@@ -248,7 +248,7 @@
           elseif(kk==3 ) then
           if( km(1)==1 )  cycle DO_620
           if( ih==0 )  cycle DO_610
-          cc(1,3) = sin(wr)
+          cc(1,3) =-sin(wr)
           cc(3,1) = cc(1,3)
           cc(1,4) = cos(wr)
           cc(4,1) = cc(1,4)
diff  -Naurd MCNP5/Source/src/prhpdf.F90 MCNP5_new/Source/src/prhpdf.F90
--- MCNP5/Source/src/prhpdf.F90       2003-04-30 20:12:06.000000000 -0600
+++ MCNP5_new/Source/src/prhpdf.F90   2004-07-22 15:14:42.000000000 -0600
@@ -16,6 +16,7 @@

   real(dknd) :: pd(59), pl(59), sb(60), sp(59)
   integer    :: np(59)
+  integer(i8knd) :: ns,ni
   character(len=100) :: da = &
     & '--------10--------20--------30--------40--------50&
       &--------60--------70--------80--------90-------100'
@@ -27,7 +28,7 @@
   character(len=7)   :: he
   character(len=8)   :: ha, hf(2)
   character(len=9)   :: hb
-
+
   ! set the upper(nu) and lower(nl) grid bin limits for score prd.
   it = ital+iper*ntal
   nu = nhsd(nsp+4,it)-1
@@ -35,145 +36,145 @@
   if( nu-nl<=0 ) then
     write(iuo,10)hsb(nl),hsb(nl+1)
 10  format(  " print table 161 cannot be done for this tally",&
-      &   " because all tallies were between",1pe12.5, " and",e12.5,&
+      &   " because all tallies were between",es12.5, " and",es12.5,&
       &   " which is one tally bin.")
     return
   endif
-
+
   ! set flag no for tallies above hsb(nsp-1) and/or below hsb(1).
   no = 0
-  if( nhsd(1,it)>0   )  no=1
-  if( nhsd(nsp,it)>0 )  no=no+2
+  if( nhsd(1,it)>0_i8knd   )  no=1
+  if( nhsd(nsp,it)>0_i8knd )  no=no+2
   kp = 59
   if( no>0 )  kp=58
-
+
   ! calculate information arrays for the distribution tables.
   ! nc is the number of bins to collapse to keep table on one page.
   nc = (nu-nl)/kp+1
-  pg = -huge
-  ps = huge
+  pg = -huge_float
+  ps = huge_float
   sb(1) = hsb(nl)
-
+
   ! loop over all of the tally distribution information.
   k = 0
-  do i=nl,nu,nc
+  do i=nl,nu,nc
     k = k+1
     np(k) = 0
-    pd(k) = 0.
-    pl(k) = 0.
-    sp(k) = 0.
-
+    pd(k) = zero
+    pl(k) = zero
+    sp(k) = zero
+
     ! collapse statistical tally information if necessary.
-    do j=1,nc
+    do j=1,nc
       if( i+j-1>nu )  exit
       np(k) = np(k)+nhsd(i+j,it)
       sp(k) = sp(k)+shsd(i+j,it)
-    end do
-
+    end do
+
     ! calculate number density in tally bin pd and take the log10.
     if( np(k)/=0 ) then
       pd(k) = np(k)*fpi/(hsb(i+j-1)-hsb(i))
-      if( pd(k)>0. )  pl(k)=log10(pd(k))
+      if( pd(k)>zero )  pl(k)=log10(pd(k))
       ps = min(ps,pl(k))
       pg = max(pg,pl(k))
     endif
     sb(k+1) = hsb(i+j-1)
-  end do
-
+  end do
+
   ! set up constants to give from 1 to 100 characters on a line.
-  d = 100./(1.01*(pg-ps))
-  pt = 1.01*ps-.01*pg
-
+  d = hundred/(1.01_dknd*(pg-ps))
+  pt = 1.01_dknd*ps-hundredth*pg
+
   ! create a character line with a 'd' approximately every decade.
   j = 1
   ld = nint(d)
   ls = ld
-  do i=2,100
+  do i=2,100
     dl(i) = '-'
     l1(i) = ' '
-    if( mod(i,ld)/=0 )  cycle
+    if( mod(i,ld)/=0 )  cycle
     dl(i) = 'd'
     if( ls>6 .or. mod(j,3)==0 )  l1(i)='|'
     j = j+1
     ld = nint(j*d)
-  end do
-
+  end do
+
   ! set some character and other variables for later use.
-  t5 = min(10.+zero,tfc(5,nn,it))
-  if( t5<=1. .and. t5/=0. )  t5=10.
+  t5 = min(ten,tfc(5,nn,it))
+  if( t5<=one .and. t5/=zero )  t5=ten
   t6 = tfc(6,nn,it)
   tb = shsd(nspt,it)
-  if( t5==0. .or. t5==10. )  tb=hsb(nsp-1)
+  if( t5==zero .or. t5==ten )  tb=hsb(nsp-1)
   hb = ',s=slope)'
   if( tb==hsb(nsp-1) )  hb=')        '
   he = '        '
-  if( hsb(nsp)<0. )  he='[f(-x)]'
+  if( hsb(nsp)<zero )  he='[f(-x)]'
   j1 = jptal(1,ital)
-
+
   ! print the history tally probability density function plot.
   sm = hsb(nsp)*shsd(nsp+2,it)/nhsd(nsp+2,it)
   write(iuo,80) j1,he,sm,nps,t5,(dl(j),j=1,100)
 80 format(/ "1unnormed tally density for tally",i4,1x,a7,4x, "nonzero",&
-    & " tally mean(m) =",1pe10.3,3x, "nps =",i10,4x,  "print table 161",&
+    & " tally mean(m) =",es10.3,3x, "nps =",i12,2x,  "print table 161",&
     & 2/," abscissa",13x, " ordinate",2x,  " log plot of tally",&
     & " probability density function in tally fluctuation chart bin",&
-    & "(d=decade,slope=",0pf4.1, ")"/&
+    & "(d=decade,slope=",f4.1, ")"/&
     & "  tally  number num den log den:",100a1)
-
+
   ! loop over k table lines and check for mean(m) line.
   ny = 0
   DO_160: do i=1,k
     n = 0
-    if( np(i)>0. )  n=nint((pl(i)-pt)*d)
+    if( np(i)>0 )  n=nint((pl(i)-pt)*d)
     ch = '*'
     if( sm>sb(i) .and. sm<=sb(i+1) ) ch='m'
-
+
     ! delete the e from two floating point numbers for tight print.
-    write(hb,'(1pe9.2)')  sb(i+1)
+    write(hb,'(es9.2)')  sb(i+1)
     hf(1) = hb(1:5)//hb(7:9)
-    write(hb,'(1pe9.2)')  pd(i)
+    write(hb,'(es9.2)')  pd(i)
     hf(2) = hb(1:5)//hb(7:9)
-
+
     ! create a line for the print plot for the ith line.
-    do j=1,100
+    do j=1,100
       l2(j) = l1(j)
-    end do
-    do j=1,n
+    end do
+    do j=1,n
       if( l2(j)/='|' )  l2(j)=ch
-    end do
-
+    end do
+
     ! set location of s for printed pareto tail plot at average sb.
-    ! log point for a pareto is t5*log10(1.+t6*(avg sb)/(t5-1.))
+    ! log point for a pareto is t5*log10(one+t6*(avg sb)/(t5-one))
     ! location nz=ny-d*(log10(sb(i+1)>tb)-log10(first sb(i+1)>tb)).
     if( tb<sb(i+1) ) then
       if( ny==0 )then
         ny = min(n+10,100)
         if( l2(ny)=='|' )  ny=ny-1
         l2(ny) = 's'
-        ty = ny+d*t5*log10(t6*(1.+t6*(0.7*sb(i)+0.3*sb(i+1))/(t5-1.)))
+        ty = ny+d*t5*log10(t6*(one+t6*(0.7_dknd*sb(i)+0.3_dknd*sb(i+1))/(t5-one)))
       else
-        nz = nint(ty-d*t5*log10(t6*(1.+t6*(0.7*sb(i)+0.3*sb(i+1))/(t5-1.))))
+        nz = nint(ty-d*t5*log10(t6*(one+t6*(0.7_dknd*sb(i)+0.3_dknd*sb(i+1))/(t5-one))))
         if( nz>1 .and. nz<=100 )  l2(nz)='s'
       endif
     endif
-
+
     ! write a line of the printed plot of the history tally pdf.
     if( np(i)<=999999 ) then
       write(iuo,130)  hf(1),np(i),hf(2),pl(i),(l2(j),j=1,100)
 130   format(a8,i7,a8,f8.3,1x,100a1)
     else
-      write(ha,'(1pe8.1)')  real(np(i))
+      write(ha,'(es8.1)')  real(np(i))
       he = ha(1:4)//ha(6:8)
       write(iuo,150)  hf(1),he,hf(2),pl(i),(l2(j),j=1,100)
 150   format(a8,a7,a8,f8.3,1x,100a1)
     endif
   end do DO_160
-
+
   ! print the total line for the printed history tally pdf plot.
   ns = nhsd(nsp+2,it)-nhsd(1,it)-nhsd(nsp,it)
-  write(hb,'(1pe9.2)')  float(ns)
+  write(hb,'(es9.2)')  real(ns)
   hf(1) = hb(1:5)//hb(7:9)
-  write(hb,'(1pe9.2)')  ns*fpi
+  write(hb,'(es9.2)')  ns*fpi
   hf(2) = hb(1:5)//hb(7:9)
   if( ns<=9999999 ) then
     write(iuo,170) ns,hf(2),(dl(j),j=1,100)
@@ -182,23 +183,23 @@
     write(iuo,180) hf(1),hf(2),(dl(j),j=1,100)
 180 format(2x, "total",2a8,9x,100a1)
   endif
-
+
   ! write any history tallies that were outside the values in hsb.
   kp = 26
   if( no/=0 ) then
    kp = 25
    if( no==1 )  write(iuo,190) nhsd(1,it)
-190 format(  " a total of",i10,&
+190 format(  " a total of",i12,&
      &   " tallies were below the score grid bin boundaries.")
    if( no==2 )  write(iuo,200) nhsd(nsp,it)
-200 format(  " a total of",i10,&
+200 format(  " a total of",i12,&
     &   " tallies were above the score grid bin boundaries.")
   if( no==3 )  write(iuo,210) nhsd(1,it),nhsd(nsp,it)
-210 format(  " a total of",i10,  " tallies were below and",i10,&
+210 format(  " a total of",i12,  " tallies were below and",i12,&
       &   " tallies were above the score grid bin boundaries.")
   endif
   if( ink(162)==0 )  return
-
+
   ! calculate and print the cumulative tally number table.
   ch = '1'
   if( k<=kp ) then
@@ -208,40 +209,40 @@
   endif
   write(iuo,250) ch,j1,he,sm,nps
 250 format(a1,  "cumulative tally number for tally",i4,1x,a7,3x,&
-    &   "nonzero tally mean(m) =",1pe10.3,3x, "nps =",i10,4x,&
+    &   "nonzero tally mean(m) =",es10.3,3x, "nps =",i12,2x,&
     &   "print table 162")
   write(iuo,260)da
 260 format(/  "   abscissa",4x,  " cum   ordinate",3x,  " plot of the",&
     &   " cumulative number of tallies in the tally fluctuation chart",&
    &   " bin from 0 to 100 percent"/  "    tally      number  cum pct:",&
    & a100)
-
+
   ! loop over k table lines and check for mean(m) line.
   n = nhsd(1,it)-nhsd(nsp+1,it)
-  do i=1,k
+  do i=1,k
     n = n+np(i)
     ch = '*'
     if( sm>sb(i) .and. sm<=sb(i+1) )  ch='m'
-    do j=1,100
+    do j=1,100
       l2(j) = h(j)
-    end do
-    l = nint(100.*n/ni)
-    do j=1,l
+    end do
+    l = nint(hundred*n/ni)
+    do j=1,l
      if( l2(j)/='|' )  l2(j)=ch
-    end do
+    end do
    if( n<=9999999 ) then
-      write(iuo,290) sb(i+1),n,100.*n/ni,(l2(j),j=1,100)
-290   format(1pe12.5,i9,0pf9.3, "|",100a1)
+      write(iuo,290) sb(i+1),n,hundred*n/ni,(l2(j),j=1,100)
+290   format(es12.5,i9,f9.3, "|",100a1)
     else
-      write(hb,'(1pe9.2)') float(n)
+      write(hb,'(es9.2)') real(n)
      hf(1) = hb(1:5)//hb(7:9)
-      write(iuo,310) sb(i+1),hf(1),100.*n/ni,(l2(j),j=1,100)
-310   format(1pe12.5,1x,a8,0pf9.3, "|",100a1)
+      write(iuo,310) sb(i+1),hf(1),hundred*n/ni,(l2(j),j=1,100)
+310   format(es12.5,1x,a8,f9.3, "|",100a1)
    endif
-  end do
-  write(iuo,330) ns,100.*n/nhsd(nsp+2,it),da
+  end do
+  write(iuo,330) ns,hundred*n/nhsd(nsp+2,it),da
 330 format(4x, "total",i12,f9.3, ":",a100)
-
+
   ! write any history tallies that were outside the values in hsb.
   kp = 15
   if( no/=0 ) then
@@ -250,7 +251,7 @@
    if( no==2 )  write(iuo,200) nhsd(nsp,it)
    if( no==3 )  write(iuo,210) nhsd(1,it),nhsd(nsp,it)
  endif
-
```

```
+
   ! calculate and print the cumulative history tally table.
   ch = '1'
   if( k<=kp ) then
@@ -259,42 +260,42 @@
   endif
   write(iuo,360) ch,j1,he,sm,nps
 360 format(/,a1,  "cumulative unnormed tally for tally",i4,1x,a7,1x,&
-    &   "nonzero tally mean(m) =",1pe10.3,3x, "nps =",i10,4x,&
+    &   "nonzero tally mean(m) =",es10.3,3x, "nps =",i10,4x,&
     &   "print table 162")
   write(iuo,370) da
 370 format(/  "  abscissa",4x, " cum",3x, " ordinate",15x, " plot of",&
     &   " the cumulative tally in the tally fluctuation chart bin from",&
    &   " 0 to 100 percent"/  "   tally    tally/nps  cum pct:",a100)
-
+
   ! loop over k table lines and check for mean(m) line.
-  do i=1,k
+  do i=1,k
     bg = bg+sp(i)*fpi
     ch = '*'
     if( sm>sb(i) .and. sm<=sb(i+1) )  ch='m'
-    do j=1,100
+    do j=1,100
       l2(j) = h(j)
-    end do
-    l = nint(100.*bg/dn)
-    do j=1,l
+    end do
+    l = nint(hundred*bg/dn)
+    do j=1,l
      if( l2(j)/='|' )  l2(j)=ch
-    end do
-    write(iuo,410) sb(i+1),bg,bg/dn,(l2(j),j=1,100)
-  end do
-410 format(1pe10.3,e11.3,1x,2pf8.3, "|",100a1)
+    end do
+    write(iuo,410) sb(i+1),bg,hundred*bg/dn,(l2(j),j=1,100)
+  end do
+410 format(es10.3,es11.3,1x,f8.3, "|",100a1)
   write(iuo,420) (shsd(nsp+2,it)-shsd(1,it)-&
-    & shsd(nsp,it))*fpi,bg/dn,da
-420 format(3x, "total",1pe13.5,1x,2pf8.3, ":",a100)
-
+    & shsd(nsp,it))*fpi,hundred*bg/dn,da
+420 format(3x, "total",es13.5,1x,f8.3, ":",a100)
+
   ! write any history tallies that were outside the values in hsb.
   if( no==0 )  return
  if( no==1 )  write(iuo,430) hsb(nsp)*shsd(1,it)*fpi
-430 format(  " a total tally/nps of",1pe12.5,&
+430 format(  " a total tally/nps of",es12.5,&
    &   " was below the score grid bin boundaries.")
  if( no==2 )  write(iuo,440) hsb(nsp)*shsd(nsp,it)*fpi
-440 format(  " a total tally/nps of",1pe12.5,&
+440 format(  " a total tally/nps of",es12.5,&
   &   " was above the score grid bin boundaries.")
  if( no==3 )  write(iuo,450) hsb(nsp)*shsd(1,it)*fpi,hsb(nsp)*shsd(nsp,it)*fpi
-450 format(  " total tallies/nps of",1pe12.5,  " were below and",&
-    & e12.5,  " were above the score grid bin boundaries.")
+450 format(  " total tallies/nps of",es12.5,  " were below and",&
+    & es12.5,  " were above the score grid bin boundaries.")
  return
 end subroutine prhpdf
diff  -Naurd MCNP5/Source/src/prinv.F90 MCNP5_new/Source/src/prinv.F90
--- MCNP5/Source/src/prinv.F90       2003-04-30 20:12:08.000000000 -0600
+++ MCNP5_new/Source/src/prinv.F90   2004-07-22 15:14:42.000000000 -0600
@@ -8,6 +8,7 @@
  use mcnp_global
  use mcnp_input
```

```
   use mcnp_debug
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)
   real(dknd) ::  nt(20)
diff  -Naurd MCNP5/Source/src/prlost.F90 MCNP5_new/Source/src/prlost.F90
--- MCNP5/Source/src/prlost.F90      2003-04-30 20:12:08.000000000 -0600
+++ MCNP5_new/Source/src/prlost.F90  2004-07-22 15:14:42.000000000 -0600
@@ -14,9 +14,9 @@
   character(len=8)  :: hw(2) = (/'newcel  ','track   ' /)
   integer :: ip(29) = (/0,1,2,3,4,0,1,2,3,0,0,0,1,2,3,0,0,0,1,2,3,0,0,1,2,3,1,2,3/)

-  !$OMP ATOMIC
+  !$OMP CRITICAL     (UPDATE_VARCOM)
   nerr = nerr+1
-
+  !$OMP END CRITICAL (UPDATE_VARCOM)
   !$OMP CRITICAL (PRINT_OUTPUT)

   ! once only, send geometry error message to the tty.
@@ -49,7 +49,7 @@
   if( kfq/=0 )  write(hf,'(i6, ".",i1)') jq,kfq
   write(iuo,50) nerr,npstc,hnp(ipt),hf,ncl(icl)
 50 format(  "1   lost particle no.",i3,5x,  "no cell found in",&
-    &   " subroutine newcel",5x,  "history no.",i9,2/," the ",a8,&
+    &   " subroutine newcel",5x,  "history no.",i12,2/," the ",a8,&
     &   " currently being tracked has reached surface",a8, "; there"/&
     &   " appears to be no cell on the other side of the surface from",&
     &  " cell",i6/  " at that point.")
@@ -93,7 +93,7 @@
 160 format(  " the distance to surface",a8,  " from the last event is",&
     & 1pe12.5/  " the distance to collision from the last event is",&
     & e12.5/  " the number of ",a8,&
-    &   " collisions so far in this history is",i6, ".")
+    &   " collisions so far in this history is",i12, ".")

   ! print a list of the rejected cells.
   do i1 = abs(lca(icl)),nlja
@@ -142,7 +142,7 @@
 230 continue
     write(iuo,240) nerr,npstc
 240 format(  "1   lost particle no.",i3,5x,  "no intersection found in",&
-    &   " subroutine track",5x,  "history no.",i9)
+    &   " subroutine track",5x,  "history no.",i12)
   if( j1==0 )  write(iuo,250) hnp(ipt),ncl(icl),hnp(ipt)
 250 format(/ " the ",a8,  " currently being tracked is in cell",i5, ".",&
     &   "  there appears"/  " to be no surface of the cell in the",&
@@ -162,7 +162,7 @@
     &   " u,v,w direction cosines:",3e15.5/&
     &  " energy =",e12.5,4x, "weight =",e12.5,4x, "time =",e12.5/&
     &   " the distance to collision from the last event is",e12.5/&
-    &   " the number of collisions so far in this history is",i6, ".")
+    &   " the number of collisions so far in this history is",i12, ".")

   ! **********************  check for cone vertex.  ***********************

diff  -Naurd MCNP5/Source/src/prstat.F90 MCNP5_new/Source/src/prstat.F90
--- MCNP5/Source/src/prstat.F90      2003-04-30 20:12:10.000000000 -0600
+++ MCNP5_new/Source/src/prstat.F90  2004-07-22 15:14:42.000000000 -0600
@@ -14,11 +14,13 @@

   implicit real(dknd) (a-h,o-z)
   character he*7,hf*8,hg*5
+  integer(i8knd) :: j_i8knd
+  integer(i8knd) :: na, nb, ni, nl, np, nq, nu, nx

   ! check if the tally fluctuation chart (tfc) bin has any tallies.
   it = ital+iper*ntal
   nt = nhsd(nsp+2,it)
-  if( nt<=0 ) then
+  if( nt<=0_i8knd ) then
```

```
     write(iuo,10) jptal(1,ital)
 10  format(/  " there are no nonzero tallies in the tally fluctuation",&
       &   " chart bin for tally",i4)
@@ -121,7 +123,7 @@
    ! print tally information in the tally fluctuation chart bin.
   write(iuo,50) jptal(1,ital),nps
 50 format(/ "1analysis of the results in the tally fluctuation",&
-    &   " chart bin (tfc) for tally",i4,  " with nps =",i10,4x,&
+    &   " chart bin (tfc) for tally",i4,  " with nps =",i12,2x,&
     &   "print table 160")
   if( hsb(nsp)<0. )  write(iuo,60)
 60 format(  " the empirical history score probability density",&
@@ -160,8 +162,8 @@
     &   " change as follows:")
   j = nps-nint(pax(1,1,1))
   if( j>0 .and. nsr==71 )  write(iuo,110) nint(pax(1,1,1)),ikz,j
-110 format( " nps =",i10,  " for this table because",i4,  " keff cycles",&
-    &  " and",i10,  " histories were skipped before tally accumulation.")
+110 format( " nps =",i12,  " for this table because",i4,  " keff cycles",&
+    &  " and",i12,  " histories were skipped before tally accumulation.")
   if( mct>=0 )  write(iuo,120) t1,t1*(1.+tc),tc,t2,rm,sqrt(1/rr)-1.,t4,&
     & vm,vr,ac,am,am/ac-1.,t3,t3*rr,rr-1.
   if( mct<0 )  write(iuo,120) t1,t1*(1.+tc),tc,t2,rm,sqrt(1/rr)-1.,t4,&
@@ -211,14 +213,14 @@
   ! see if extreme f(x) values decline monotonically last 5 cycles.
   ! skip the check if f(x) appears bounded or has only one bin.
   if( t5==10. .and. np==0 .or. nu==nl )  go to 260
-  do j = nu-min(5,nu-nl-1),nu
-    if( nhsd(j,it)/=0 )  go to 210
+  do j = nu-min(5_i8knd,nu-nl-1_i8knd),nu
+    if( nhsd(j,it)/=0_i8knd )  go to 210
   enddo
   go to 260
 210 continue
   nb = nhsd(j,it)
   do k = j+1,nu+1
-    if( nhsd(k,it)==0 )  cycle
+    if( nhsd(k,it)==0_i8knd )  cycle
    if( nb>=nhsd(k,it) )  go to 220
    if( nb+nint(6.*sqrt(float(nb)))<nhsd(k,it) )  go to 240
 220 continue
@@ -233,7 +235,7 @@

   ! check for holes in the last 5 bins of the high tally tail.
 260 continue
-  do j = nu-min(5,nu-nl-1),nu
+  do j = nu-min(5_i8knd,nu-nl-1),nu
    if( nhsd(j,it)==0 )  go to 290
  enddo
  write(iuo,280)
diff  -Naurd MCNP5/Source/src/prtfcc.F90 MCNP5_new/Source/src/prtfcc.F90
--- MCNP5/Source/src/prtfcc.F90       2003-04-30 20:12:10.000000000 -0600
+++ MCNP5_new/Source/src/prtfcc.F90   2004-07-22 15:14:42.000000000 -0600
@@ -9,9 +9,11 @@
   use mcnp_global
   use mcnp_debug
   use ra2_mod
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)
   character ch,fd(4)*3,mc(3)*8,sr(10)*4
+  integer(i8knd) :: nw

   ! set tfc quantities for later use.
   it = ital+iper*ntal
@@ -22,7 +24,7 @@
   if( t5<=1. .and. t5/=0. )  t5 = 10.

   ! set the statistical check parameters to pass all of the tests.
-  nw = 0
+  nw = 0_i8knd
```

```
   do i = 1,4
     fd(i) = 'yes'
   enddo
@@ -50,7 +52,7 @@
     nk = nskk
     do i=1,nn
       ! Never use the first tfc bin even if npc(1)>nskk (nh always >1)
-      if( npc(i)>nskk )  exit
+      if( npc(i)>int(nskk,i8knd) )  exit
     enddo
     nh = i + (nn-i)/2 + 1
   endif
@@ -72,8 +74,8 @@
       if( tfc(1,j+1,it)>tfc(1,j,it) .and. is==1 .or. &
         & tfc(1,j+1,it)<tfc(1,j,it) .and. is==2 )  go to 80
     enddo
-    nw = nw+1
-    nhsd(nsp+8,it) = 2
+    nw = nw+1_i8knd
+    nhsd(nsp+8,it) = 2_i8knd
     sr(1)='  no'
     mc(1)='decrease'
     if(is == 2)mc(1)='increase'
@@ -85,10 +87,10 @@
 80 continue
   if( t2==0. )  go to 140
   if( t2>=rd ) then
-    nw = nw+1
+    nw = nw+1_i8knd
     sr(2) = '  no'
-    nhsd(nsp+8,it) = 3
-    if( jptal(2,ital)==5 )  nhsd(nsp+8,it) = 4
+    nhsd(nsp+8,it) = 3_i8knd
+    if( jptal(2,ital)==5 )  nhsd(nsp+8,it) = 4_i8knd
   endif

   ! check for a decreasing relative error for last half of problem.
@@ -97,7 +99,7 @@
   do i = nb+1,nn
     if( tfc(2,i-1,it)==0.)  nr = nr+1
     if( tfc(2,i,it)<=tfc(2,i-1,it) )  cycle
-    t = sqrt(float(npc(i-1)-nk)/float(npc(i)-nk))
+    t = sqrt(real(npc(i-1)-int(nk,i8knd),dknd)/real(npc(i)-int(nk,i8knd),dknd))
    s = npc(i-1)-nk-4*nsrck*(mcheck-ikz)
    if( kcheck>0 .and. kcz>mcheck .and. mcheck>ikz .and. s>0. )&
      & t = sqrt(s/(s+npc(i)-npc(i-1)))
@@ -106,15 +108,15 @@
   enddo
   go to 120
 110 continue
-  nw = nw+1
-  if( nw==1 )  nhsd(nsp+8,it) = 5
+  nw = nw+1_i8knd
+  if( nw==1_i8knd )  nhsd(nsp+8,it) = 5_i8knd
   sr(3)='  no'
   fd(1)=' no'
   go to 130

   ! check 1/sqrt(nps) rel error for max(5,last half of problem).
 120 continue
-  t = sqrt(float(npc(nr)-nk)/(nps-nk))
+  t = sqrt(real(npc(nr)-int(nk,i8knd),dknd)/(nps-int(nk,i8knd)))
  s = npc(nr)-nk-4*nsrck*(mcheck-ikz)
  if( kcheck>0 .and. kcz>mcheck .and. mcheck>ikz .and. s>0.)&
    & t = sqrt(.5*s/pax(1,1,1))
@@ -123,8 +125,8 @@
   if( max(r,1./r)<=max(1.05+zero,sqrt(1.+5.*sqrt(t7)+12.5*t7)))&
     & go to 140
 130 continue
-  nw = nw+1
-  if( nw==1 )  nhsd(nsp+8,it) = 6
```

```
+  nw = nw+1_i8knd
+  if( nw==1_i8knd )  nhsd(nsp+8,it) = 6_i8knd
   sr(4)='  no'
   fd(2)=' no'

@@ -132,8 +134,8 @@
 140 continue
   if( t4==0. )  go to 200
   if( t4>=0.1 ) then
-    nw = nw+1
-    if( nw==1 )  nhsd(nsp+8,it) = 7
+    nw = nw+1_i8knd
+    if( nw==1_i8knd )  nhsd(nsp+8,it) = 7_i8knd
     sr(5)='  no'
   endif

@@ -145,8 +147,8 @@
   enddo
   go to 180
 170 continue
-  nw = nw+1
-  if( nw==1 )  nhsd(nsp+8,it) = 8
+  nw = nw+1_i8knd
+  if( nw==1_i8knd )  nhsd(nsp+8,it) = 8_i8knd
   sr(6)='  no'
   fd(3)=' no'
   go to 190
@@ -161,8 +163,8 @@
   r = t*tfc(4,nr,it)/t4
   if( max(r,1./r)<1.5 )  go to 200
 190 continue
-  nw = nw+1
-  if( nw==1 )  nhsd(nsp+8,it) = 9
+  nw = nw+1_i8knd
+  if( nw==1_i8knd )  nhsd(nsp+8,it) = 9_i8knd
   sr(7)='  no'
   fd(4)=' no'

@@ -172,8 +174,8 @@
     r = tfc(3,nb,it)/tfc(3,nn,it)
     t7 = max(tfc(4,nb,it),t4)
     if( max(r,1./r)>=max(1.02+zero,min(1.5+zero, 1.+5.*sqrt(t7)+12.5*t7)) ) then
-      nw = nw+1
-      if( nw==1 )  nhsd(nsp+8,it) = 10
+      nw = nw+1_i8knd
+      if( nw==1_i8knd )  nhsd(nsp+8,it) = 10_i8knd
      sr(8) = '  no'
      mc(2) = 'decrease'
      if(r < 1.)mc(2)='increase'
@@ -187,8 +189,8 @@
        if( tfc(3,j+1,it)>tfc(3,j,it) .and. is==1 .or. &
          & tfc(3,j+1,it)<tfc(3,j,it) .and. is==2 )  go to 230
      enddo
-      nw = nw+1
-      if( nw==1 )  nhsd(nsp+8,it) = 11
+      nw = nw+1_i8knd
+      if( nw==1_i8knd )  nhsd(nsp+8,it) = 11_i8knd
      sr(9) = '  no'
      mc(3) = 'decrease'
      if(is == 2)mc(3)='increase'
@@ -198,18 +200,18 @@
   ! check for a steep enough slope in the extreme history tallies.
 230 continue
   if( t5<3. ) then
-    nw = nw+1
+    nw = nw+1_i8knd
    sr(10) = '  no'
-    if( nw<=1 ) then
-      if( t5/=0.)  nhsd(nsp+8,it) = 12
-      if( t5==0. .and. nhsd(nsp+5,it)==0) nhsd(nsp+8,it) = 13
-      if( t5==0. .and. nhsd(nsp+5,it)>0)  nhsd(nsp+8,it) = 14
```

```
+    if( nw<=1_i8knd ) then
+      if( t5/=0.)  nhsd(nsp+8,it) = 12_i8knd
+      if( t5==0. .and. nhsd(nsp+5,it)==0) nhsd(nsp+8,it) = 13_i8knd
+      if( t5==0. .and. nhsd(nsp+5,it)>0)  nhsd(nsp+8,it) = 14_i8knd
     endif
   endif

   ! write the table for the 10 statistical checks.
   nhsd(nsp+7,it) = nw
-  if( nw==0 )  nhsd(nsp+8,it) = 15
+  if( nw==0 )  nhsd(nsp+8,it) = 15_i8knd
   if( nn-nb<=3 )  write(iuo,250) nn-nb+1
 250 format(2/," ***** the nps-dependent tfc bin check results are",&
     &   " suspect because there are only",i3,  " nps tally values to",&
@@ -236,7 +238,7 @@
     write(iuo,270) nw
 270 format( 2/, " warning.  the tally in the tally fluctuation chart",&
       &   " bin did not pass",i3,  " of the 10 statistical checks.")
-    call erprnt(1,2,2,jptal(1,ital),nw,0,0,-1,'"tally",i4,&
+    call erprnt_i8_k2(1,2,2,jptal(1,ital),nw,0,0,-1,'"tally",i4,&
       & " tfc bin did not pass",i3,  " of 10 statistical checks."')
     return
   endif
diff  -Naurd MCNP5/Source/src/psurf.F90 MCNP5_new/Source/src/psurf.F90
--- MCNP5/Source/src/psurf.F90       2003-04-30 20:12:12.000000000 -0600
+++ MCNP5_new/Source/src/psurf.F90   2004-07-22 15:14:42.000000000 -0600
@@ -6,6 +6,7 @@
   use mcnp_global
   use mcnp_input
   use mcnp_debug
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/ptfc.F90 MCNP5_new/Source/src/ptfc.F90
--- MCNP5/Source/src/ptfc.F90 2003-04-30 20:12:12.000000000 -0600
+++ MCNP5_new/Source/src/ptfc.F90     2004-07-22 15:14:42.000000000 -0600
@@ -6,6 +6,7 @@

   use mcnp_global
   use mcnp_debug
+  use erprnt_mod, only: erprnt

   implicit real(dknd) (a-h,o-z)
   real(dknd) ::  t5(3)
@@ -163,7 +164,7 @@
       write(iuo,230) ('tally',jptal(1,i),i=jt,la)
 230   format(/23x,3(a5,i5,32x))
       write(iuo,240)('mean','error','vov','slope','fom',i=jt,la)
-240   format(7x, "nps",3(6x,a4,5x,a5,3x,a3,2x,a5,4x,a3))
+240   format(10x, "nps",3(6x,a4,5x,a5,3x,a3,2x,a5,4x,a3))
       !
       ! print the npc, tfc table.
       do l=1,nn
@@ -180,7 +181,7 @@
         end do
         write(iuo,270)  npc(l),((tfc(j,l,i+k),j=1,2),tfc(4,l,i+k),&
          & t5(i-jt+1),ha(i-jt+1),i=jt,la)
-270     format(i10,3(2x,1pe11.4,0p2f7.4,f5.1,a8))
+270     format(1x,i12,3(2x,1pe11.4,0p2f7.4,f5.1,a8))
       end do
     end do
   end do
diff  -Naurd MCNP5/Source/src/ptrak.F90 MCNP5_new/Source/src/ptrak.F90
--- MCNP5/Source/src/ptrak.F90       2003-04-30 20:12:14.000000000 -0600
+++ MCNP5_new/Source/src/ptrak.F90   2004-07-22 15:14:42.000000000 -0600
@@ -6,6 +6,7 @@

   use mcnp_global
   use mcnp_debug
+  use erprnt_mod, only: erprnt
```

```
   implicit real(dknd) (a-h,o-z)

@@ -15,8 +16,8 @@
     &                     19,23,24,25,-2,-3,-7,-8,-10,-11,-12,-13 /)
   integer :: il(6)   = (/ 8, 7, 6,11, 3, 5 /)
   integer :: in(5)   = (/ 0, 0, 0, 0, 0 /)
-  integer :: ip(nptr+1) = (/ (0,j=1,nptr+1) /)
-  integer :: iq(3,5) = reshape( (/ 1, (0,j=1,14) /),  (/3,5/) )
+  integer(i8knd) :: ip(nptr+1) = (/ (0,j=1,nptr+1) /)
+  integer(i8knd) :: iq(3,5) = reshape( (/ 1, (0,j=1,14) /),  (/3,5/) )
   integer :: iv(5)   = (/ 0, 0, 2, 9,10 /)
   integer :: iw(6)   = (/ 1, 2, 3, 4, 5, 6 /)
   integer :: iz(25)  = (/1,0,0,1,0,0,1,1,1,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0/)
@@ -25,7 +26,7 @@
   character(len=74) :: hq
   character(len=14) :: hf
   save hs,ht,ih,il,in,ip,iq,iv,iz,nn
-
+  integer(i8knd) :: l,lp
   select case( m )

   case( 0 )
@@ -127,11 +128,11 @@
       if( kpt(i)/=0 )  k = i
       n = n+kpt(i)
     enddo
-    if( n==1 )  ip(11) = 1
+    if( n==1 )  ip(11) = 1_i8knd
     if( n/=1 )  k = 0
     i1 = 6+iptra(13)
     i2 = 3**iptra(13)
-    ip(13) = i1+i2
+    ip(13) = int(i1+i2,i8knd)
     do i = 1,5
       iq(1,i) = i1-ip(11)-iq(1,i)
       iq(2,i) = i2-iq(2,i)
@@ -152,7 +153,7 @@
     !  >>>>>  m=1 through 5 -- filter and write the ptrac events.
     ! m=1 source; m=2 bank; m=3 surface; m=4 collision; m=5 term.
     im = 1
-    if( m==5 )  ip(5) = ip(5)+1
+    if( m==5 )  ip(5) = ip(5)+1_i8knd

     ! apply the various event filters - bank events are special.
     DO_290: do i = 1,6
@@ -191,7 +192,7 @@
           endif
         enddo
       case( 2, 3 )
-        if( ip(il(i))<lp )  cycle DO_290
+        if( ip(il(i))<int(lp,i8knd) )  cycle DO_290
         if( m/=2 .or. il(i)/=5 )  return
         im = -1
       case( 1 )
@@ -204,14 +205,14 @@
     enddo DO_290

     ! eliminate useless bank events, increment the various counters.
-    if( iptra(3)/=0 .or. m/=2 .or. ip(3)/=2 ) then
-      ip(6) = ip(6)+1
-      ip(7) = ip(7)+1
-      ip(1) = ip(1)+1
+    if( iptra(3)/=0 .or. m/=2 .or. ip(3)/=2_i8knd ) then
+      ip(6) = ip(6)+1_i8knd
+      ip(7) = ip(7)+1_i8knd
+      ip(1) = ip(1)+1_i8knd
     endif

     ! write the appropriate event data to the buffer.
-    lp = ip(13)*ip(1)
```

```
+    lp = int(ip(13)*ip(1),i4knd)
     pts(lp+1) = 9000.
     pts(lp+2) = node
     l = 0
@@ -244,7 +245,7 @@
       lp = lp+1
     end select
     pts(lp+4) = ipt
-    if( ip(11)==0 )  lp = lp+1
+    if( ip(11)==0_i8knd )  lp = lp+1
     pts(lp+4) = ncl(icl)
     pts(lp+5) = mat(icl)
     pts(lp+6) = ncp
@@ -257,15 +258,15 @@
         pts(lp+5+i) = udt(i,0)
       enddo
     endif
-    pts(ip(13)*(ip(1)-1)+1) = float(im*(m*1000+l))
+    pts(int(ip(13)*(ip(1)-1)+1,i4knd)) = float(im*(m*1000+l))
     ip(3) = m
     if( ip(1)<=iptra(1) )  return

     ! when full, move the buffer to the scratch file.
-    if( ip(4)==0 )  ip(4) = nint(pts(1))
+    if( ip(4)==0_i8knd )  ip(4) = nint(pts(1),i8knd)
     rp(1) = pts(1)
     do i = 1,iptra(1)
-      lp = ip(13)*i
+      lp = int(ip(13)*i,i4knd)
       n = iq(3,nint(abs(rp(1))))/1000
       do j = 1,n
         rp(j) = pts(lp+j)
@@ -275,8 +276,8 @@

     ! move the last event to the front of the buffer.
     pts(1) = rp(1)
-    do j = 1,ip(13)
-      pts(ip(13)+j) = pts(lp+ip(13)+j)
+    do j = 1,int(ip(13),i4knd)
+      pts(int(ip(13),i4knd)+j) = pts(lp+int(ip(13),i4knd)+j)
     enddo
     ip(1) = 1

@@ -286,7 +287,7 @@
     do i = 3,4
       l = iv(i)
       lp = abs(iptra(l))
-      if( lp==0 .or. ip(l)/=0 )  go to 470
+      if( lp==0 .or. ip(l)/=0_i8knd )  go to 470
       do j = 1,nint(ptr(lp))
         if( i/=4 .and. nint(ptr(lp+j))==k   )  go to 460
         if( i==4 .and. ptr(lp+j)==k+kfq*0.1 )  go to 460
@@ -302,13 +303,13 @@

   case( 7 )
     !  >>>>>  m=7 -- filter the history and move events to the output file.
-    if( ip(7)==0     )  go to 690
+    if( ip(7)==0_i8knd     )  go to 690
     if( abs(iptr)==1 )  go to 570

     if( iptra(2)/=0 ) then
       ! check the cell filters.
-      if( ip(2)==0    )  go to 680
-      ip(2) = ncl(ip(2))
+      if( ip(2)==0_i8knd  )  go to 680
+      ip(2) = ncl(int(ip(2),i4knd))
     endif
     if( iptra(9)/=0 ) then
       ! check the surface filters.
@@ -334,9 +335,9 @@
         if( it<=0 ) then
```

```
           ! find the tfc bin value and the appropriate multiplier.
           do l = 1,20
-            if( npc(l)==0 )  exit
+            if( npc(l)==0_i8knd )  exit
           enddo
-          l = max(1,l-1)
+          l = max(1_i8knd,l-1_i8knd)
           tv = tfc(1,l,k)*10.
           if( tv==0. )  cycle
           if( lv/=0  )  tv = tfc(1,l,k)*ptr(lv+i)
@@ -353,8 +354,8 @@

     ! write the nps line to the output file.
 570 continue
-    ip(8) = ip(8)+1
-    if( ip(4)==0 )  ip(4) = nint(pts(1))
+    ip(8) = ip(8)+1_i8knd
+    if( ip(4)==0 )  ip(4) = nint(pts(1),i8knd)
     iw(1) = npstc
     iw(2) = ip(4)
     ! check for surface facet for write.
@@ -462,9 +463,9 @@
     ip(6) = ip(6)-ip(7)
 690 continue
     rewind(iupc)
-    ip(1:5) = 0
-    ip(7) = 0
-    ip(9) = 0
+    ip(1:5) = 0_i8knd
+    ip(7) = 0_i8knd
+    ip(9) = 0_i8knd
     if( iptr<0 .and. ip(6)>=iptra(6) )  nst = nst+256

   case( 8 )
@@ -473,7 +474,7 @@
     close(iupw)
     write(jtty,720) ht(iptra(4)),ptrac,ip(6),ip(8)
     write(iuo, 720) ht(iptra(4)),ptrac,ip(6),ip(8)
-720 format(1x,a6," file ",a8," written with ",i7," events from ",i7," histories.")
+720 format(1x,a6," file ",a8," written with ",i12," events",/29x," from ",i12,"
histories.")

   end select

diff  -Naurd MCNP5/Source/src/qttyin.F90 MCNP5_new/Source/src/qttyin.F90
--- MCNP5/Source/src/qttyin.F90      2003-04-30 20:12:16.000000000 -0600
+++ MCNP5_new/Source/src/qttyin.F90  2004-07-22 15:14:43.000000000 -0600
@@ -1,73 +1,167 @@
-!+ $Id: qttyin.F90,v 1.7 2002/12/03 19:30:08 ljcox Exp $
+!+ $Id: qttyin.F90,v 1.2 2004/04/01 23:48:02 jgoorley Exp $
 ! Copyright LANL/UC/DOE - see file COPYRIGHT_INFO

-subroutine qttyin(ia,hm)
-  ! Description:
-  ! Process tty interrupts if irup=1 flag on.  hm=message.
-  ! The allowed interrupts are
-  !   (ctrl-c) status (s) (or nothing) -- return program status.
-  !   (ctrl-c) mcplot (m) -- call mcplot.  mcrun only.
-  !   (ctrl-c) quit (q) -- terminate at history end.  mcrun only.
-  !   (ctrl-c) kill (k) -- unconditional kill.
+module qttyin_mod

-  ! Modules:
-  use dmmp, only: dm_term
-  use mcnp_global
-  use mcnp_debug
-  use gxsub, only : gxquit
+  interface qttyin
+    ! ==> specific routines used for generic subroutine:
+    module procedure qttyin_i4, qttyin_i8
+  end interface
```

```
-  implicit real(dknd) (a-h,o-z)
-  character(len=*)  :: hm
-  character(len=60) :: ha
-  character(len=8)  :: hg
-  character(len=6)  :: is(4) = (/'status','mcplot','quit  ','kill  '/)
-  character(len=2)  :: js(4) = (/'is','im','iq','ik'/)
+  contains

-  ! Get the message from the tty.
-  irup = 0
-  write(jtty,'(  " **** interrupt.  enter s (status), m (mcplot), ",'&
-    & //'  "q (quit), k (kill)")')
-  read(itty,'(a8)')  hg
-  call nxtsym(hg,' ',1,i,j,2)
-  do mi = 1,4
-    if( hg==is(mi)(1:1) .or. hg==is(mi) .or. hg==js(mi) .or. hg==' ' )  exit
-  enddo
-  if( (mi==2 .or. mi==3) .and. iovr/=4 )  mi = 1
+  subroutine qttyin_i4(ia,hm)
+    ! Description:
+    ! Process tty interrupts if irup=1 flag on.  hm=message.
+    ! The allowed interrupts are
+    !   (ctrl-c) status (s) (or nothing) -- return program status.
+    !   (ctrl-c) mcplot (m) -- call mcplot.  mcrun only.
+    !   (ctrl-c) quit (q) -- terminate at history end.  mcrun only.
+    !   (ctrl-c) kill (k) -- unconditional kill.

-  select case( mi )
+    ! Modules:
+    use dmmp, only: dm_term
+    use mcnp_global
+    use mcnp_debug
+    use gxsub, only : gxquit

-    case( 1 )
-      !  >>>>>  mi=1 -- return the status.
-      call secnd(t)
-      ha = '( " time =",f9.2,5x,'//hm//')'
-      if( ia<0  )  write(jtty,ha) (t)/60.,nps,nch(1)+nch(2)+nch(3)
-      if( ia==0 )  write(jtty,ha) (t)/60.
-      if( ia>0  )  write(jtty,ha) (t)/60.,ia
-      return
+    implicit none

-    case( 2 )
-      !  >>>>>  mi=2 -- call mcplot.
-      kmplot = 1
-      write(jtty,'(  " will call plotter after history",i10)')  nps
-      return
+    integer(i4knd), intent(in) :: ia
+    character(len=*), intent(in)  :: hm

-    case( 3 )
-      !  >>>>>  mi=3 -- terminate run after this history.
-      nst = nst+16
-      write(jtty,'(  " will quit after history",i10)')  nps
-      return
+    integer :: i,j,mi
+    real(dknd) :: t

-    case( 4 )
-      !  >>>>>  mi=4 -- unconditional kill.
-      call gxquit
-      if( mcnp_opt_multp ) then
-        if( ltasks>1 .and. iovr==4 )  call msgcon(3)
-        if( ltasks>=0 )  call dm_term
-      endif
+    character(len=60) :: ha
+    character(len=8)  :: hg
+    character(len=6)  :: is(4) = (/'status','mcplot','quit  ','kill  '/)
+    character(len=2)  :: js(4) = (/'is','im','iq','ik'/)

-      stop
+    ! Get the message from the tty.
+    irup = 0
+    write(jtty,'(  " **** interrupt.  enter s (status), m (mcplot), ",'&
+      & //'  "q (quit), k (kill)")')
+    read(itty,'(a8)')  hg
+    call nxtsym(hg,' ',1,i,j,2)
+    do mi = 1,4
+      if( hg==is(mi)(1:1) .or. hg==is(mi) .or. hg==js(mi) .or. hg==' ' )  exit
+    enddo
+    if( (mi==2 .or. mi==3) .and. iovr/=4 )  mi = 1

-  end select
+    select case( mi )

-  return
-end subroutine qttyin
+      case( 1 )
+        !  >>>>>  mi=1 -- return the status.
+        call secnd(t)
+        ha = '( " time =",f9.2,5x,'//hm//')'
+        if( ia  < 0 )  write(jtty,ha) t/sixty,nps,sum(nch)
+        if( ia == 0 )  write(jtty,ha) t/sixty
+        if( ia  > 0 )  write(jtty,ha) t/sixty,ia
+        return
+
+      case( 2 )
+        !  >>>>>  mi=2 -- call mcplot.
+        kmplot = 1
+        write(jtty,'(  " will call plotter after history",i12)')  nps
+        return
+
+      case( 3 )
+        !  >>>>>  mi=3 -- terminate run after this history.
+        nst = nst+16
+        write(jtty,'(  " will quit after history",i12)')  nps
+        return
+
+      case( 4 )
+        !  >>>>>  mi=4 -- unconditional kill.
+        call gxquit
+        if( mcnp_opt_multp ) then
+          if( ltasks>1 .and. iovr==4 )  call msgcon(3)
+          if( ltasks>=0 )  call dm_term
+        endif
+
+        stop
+
+    end select
+
+    return
+  end subroutine qttyin_i4
+
+  subroutine qttyin_i8(ia,hm)
+    ! Description:
+    ! Process tty interrupts if irup=1 flag on.  hm=message.
+    ! The allowed interrupts are
+    !   (ctrl-c) status (s) (or nothing) -- return program status.
+    !   (ctrl-c) mcplot (m) -- call mcplot.  mcrun only.
+    !   (ctrl-c) quit (q) -- terminate at history end.  mcrun only.
+    !   (ctrl-c) kill (k) -- unconditional kill.
+
+    ! Modules:
+    use dmmp, only: dm_term
+    use mcnp_global
+    use mcnp_debug
+    use gxsub, only : gxquit
+
+    implicit none
```

```
+
+    integer(i8knd), intent(in) :: ia
+    character(len=*), intent(in)  :: hm
+
+    integer :: i,j,mi
+    real(dknd) :: t
+
+    character(len=60) :: ha
+    character(len=8)  :: hg
+    character(len=6)  :: is(4) = (/'status','mcplot','quit  ','kill  '/)
+    character(len=2)  :: js(4) = (/'is','im','iq','ik'/)
+
+    ! Get the message from the tty.
+    irup = 0
+    write(jtty,'(  " **** interrupt.  enter s (status), m (mcplot), ",'&
+      & //'  "q (quit), k (kill)")')
+    read(itty,'(a8)')  hg
+    call nxtsym(hg,' ',1,i,j,2)
+    do mi = 1,4
+      if( hg==is(mi)(1:1) .or. hg==is(mi) .or. hg==js(mi) .or. hg==' ' )  exit
+    enddo
+    if( (mi==2 .or. mi==3) .and. iovr/=4 )  mi = 1
+
+    select case( mi )
+
+      case( 1 )
+        !  >>>>>  mi=1 -- return the status.
+        call secnd(t)
+        ha = '( " time =",f9.2,5x,'//hm//')'
+        if( ia  < 0_i8knd )  write(jtty,ha) t/sixty,nps,sum(nch)
+        if( ia == 0_i8knd )  write(jtty,ha) t/sixty
+        if( ia  > 0_i8knd )  write(jtty,ha) t/sixty,ia
+        return
+
+      case( 2 )
+        !  >>>>>  mi=2 -- call mcplot.
+        kmplot = 1
+        write(jtty,'(  " will call plotter after history",i12)')  nps
+        return
+
+      case( 3 )
+        !  >>>>>  mi=3 -- terminate run after this history.
+        nst = nst+16
+        write(jtty,'(  " will quit after history",i12)')  nps
+        return
+
+      case( 4 )
+        !  >>>>>  mi=4 -- unconditional kill.
+        call gxquit
+        if( mcnp_opt_multp ) then
+          if( ltasks>1 .and. iovr==4 )  call msgcon(3)
+          if( ltasks>=0 )  call dm_term
+        endif
+
+        stop
+
+    end select
+
+    return
+  end subroutine qttyin_i8
+
+end module qttyin_mod
diff  -Naurd MCNP5/Source/src/ra2_mod.F90 MCNP5_new/Source/src/ra2_mod.F90
--- MCNP5/Source/src/ra2_mod.F90     2003-04-30 20:12:20.000000000 -0600
+++ MCNP5_new/Source/src/ra2_mod.F90 2004-07-22 15:14:43.000000000 -0600
@@ -6,6 +6,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod
```

```
   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/rdprob.F90 MCNP5_new/Source/src/rdprob.F90
--- MCNP5/Source/src/rdprob.F90      2003-04-30 20:12:20.000000000 -0600
+++ MCNP5_new/Source/src/rdprob.F90  2004-07-22 15:14:43.000000000 -0600
@@ -7,6 +7,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)
   character(len=80) :: hl
@@ -17,6 +18,12 @@
     if( krq(6,i)==-1 )  krq(6,i) = -mxa
   enddo

+  ! Allocate flag arrays for fm, de df card checks for spdtl
+  allocate ( flag_speed_tally_fm(0:max(1,ntal),1:2) )
+  allocate ( flag_speed_tally_de(0:max(1,ntal),1:2) )
+  flag_speed_tally_fm = 0
+  flag_speed_tally_de = 0
+
   ! process 1 or 3 data blocks, which are separated by blank lines.
   rewind iu1
   jui = iui
@@ -114,5 +121,68 @@
       if( kl<=1 ) exit
     enddo
   enddo DO_80
+
+  ! Issue statement about lattice speed tally modifications, if necessary.
+  do i=1,ntal
+    ! Check presence of fm#4 tally cards for spdt
+    if ( flag_speed_tally_fm(i,1) == 0 .and. &
+      &  flag_speed_tally_fm(i,2) /= 0 ) then
+      if ( flag_speed_tally_ok == 1) flag_speed_tally_ok = -1
+      if ( flag_speed_tally_force == 1 )  call erprnt(1,3,1,&
+        & flag_speed_tally_fm(i,2),0,0,0,0,&
+        & '"need fm card with tally",i4," for lattice speed tally."')
+    endif
+    ! Check presence of de#4 df#4 tally cards for spdtl
+    if ( flag_speed_tally_de(i,1) == 0 .and. &  ! check for de# card
+      &  flag_speed_tally_de(0,1) == 0 .and. &  ! check for de0 card
+      &  flag_speed_tally_de(i,2) /= 0) then
+      if ( flag_speed_tally_ok == 1) flag_speed_tally_ok = -1
+      if ( flag_speed_tally_force == 1 )  call erprnt(1,3,1,&
+        & flag_speed_tally_de(i,2),0,0,0,0,&
+        & '"need de df cards with tally",i4," for lattice speed tally."')
+    endif
+  enddo
+  !
+  if ( flag_speed_tally_ok == 1 .and. &
+    & (flag_speed_tally_force == 0 .or. flag_speed_tally_force == 1) ) then
+    call erprnt(1,2,0,0,0,0,0,0,&
+      & '"**********************************************************************"')
+    call erprnt(1,2,0,0,0,0,0,0,'"Using lattice speed tally modifications."')
+    call erprnt(1,2,0,0,0,0,0,0,&
+      & '"  User should review input deck and verify the following are true:"')
+    call erprnt(1,2,0,0,0,0,0,0,&
+      & '"  1) Nested lattices are not tallied over."')
+    call erprnt(1,2,0,0,0,0,0,0,&
+      & '"  2) A cell with the fill keyword does not reference its own universe."')
+    call erprnt(1,2,0,0,0,0,0,0,&
+      & '"  3) Lattice index range on tally must match corresponding fill range."')
+    call erprnt(1,2,0,0,0,0,0,0,&
+      & '" Failure to meet these criteria may result in silent wrong answers."')
+    call erprnt(1,2,0,0,0,0,0,0,&
+      & '"  See the Lattice Speed Tally Enhancement report: LA-UR-04-3400"')
+    call erprnt(1,2,0,0,0,0,0,0,&
+      & '"**********************************************************************"')
+    flag_speed_tally_used=1
+  endif
+  if ( flag_speed_tally_ok == 1 .and. flag_speed_tally_force == -1 ) then
+    call erprnt(1,3,0,0,0,0,0,0,&
+      & '"lattice speed tally modifications ok to use, but have been turned off."')
+    flag_speed_tally_used=-1
+  endif
+  if ( flag_speed_tally_ok == -1 .and. &
+    & (flag_speed_tally_force == -1 .or. flag_speed_tally_force == 0) ) then
+    call erprnt(1,3,0,0,0,0,0,0,&
+      & '"lattice speed tally modifications will not be used."')
+    flag_speed_tally_used=-1
+  endif
+  if( flag_speed_tally_ok == -1 .and. flag_speed_tally_force == 1 ) then
+    call erprnt(1,2,0,0,0,0,0,0, &
+      & '"using lattice speed tally even though not appropriate."')
+    call erprnt(1,2,0,0,0,0,0,0, &
+      & '"    Silent wrong answers or crash may result."')
+    flag_speed_tally_used=1
+  endif
+
+
   return
 end subroutine rdprob
diff  -Naurd MCNP5/Source/src/regula.F90 MCNP5_new/Source/src/regula.F90
--- MCNP5/Source/src/regula.F90      2003-04-30 20:12:22.000000000 -0600
+++ MCNP5_new/Source/src/regula.F90  2004-07-22 15:14:43.000000000 -0600
@@ -10,6 +10,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_plot
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/rhoden.F90 MCNP5_new/Source/src/rhoden.F90
--- MCNP5/Source/src/rhoden.F90      2003-04-30 20:12:22.000000000 -0600
+++ MCNP5_new/Source/src/rhoden.F90  2004-07-22 15:14:43.000000000 -0600
@@ -6,6 +6,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)
   character(len=82) :: ht
diff  -Naurd MCNP5/Source/src/ronge.F90 MCNP5_new/Source/src/ronge.F90
--- MCNP5/Source/src/ronge.F90       2003-04-30 20:12:24.000000000 -0600
+++ MCNP5_new/Source/src/ronge.F90   2004-07-22 15:14:43.000000000 -0600
@@ -15,6 +15,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_landau
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)
   integer, parameter :: km = 5
diff  -Naurd MCNP5/Source/src/runtpr.F90 MCNP5_new/Source/src/runtpr.F90
--- MCNP5/Source/src/runtpr.F90      2003-04-30 20:12:26.000000000 -0600
+++ MCNP5_new/Source/src/runtpr.F90  2004-07-22 15:14:43.000000000 -0600
@@ -16,7 +16,7 @@
   ie = 0

   read(iu,end = 10) hk,hv,hl,hi,hc,hp
-  read(iu,end = 10) avarcm,gvarcm,jvarcm,rdum,idum
+  read(iu,end = 10) avarcm,gvarcm,i8varcm,jvarcm,rdum,idum
   call vdac_read(iu)
   call fmesh_runtpr(iu)
   return
diff  -Naurd MCNP5/Source/src/runtpw.F90 MCNP5_new/Source/src/runtpw.F90
--- MCNP5/Source/src/runtpw.F90      2003-04-30 20:12:28.000000000 -0600
```

```
+++ MCNP5_new/Source/src/runtpw.F90   2004-07-22 15:14:43.000000000 -0600
@@ -12,7 +12,7 @@
   implicit real(dknd) (a-h,o-z)

   write(iu) kod,ver,loddat,idtm,chcd,probid
-  write(iu) avarcm,gvarcm,jvarcm,rdum,idum
+  write(iu) avarcm,gvarcm,i8varcm,jvarcm,rdum,idum
   call vdac_write(iu)
   call fmesh_runtpw(iu)

diff  -Naurd MCNP5/Source/src/setdas.F90 MCNP5_new/Source/src/setdas.F90
--- MCNP5/Source/src/setdas.F90      2003-11-05 17:23:12.000000000 -0700
+++ MCNP5_new/Source/src/setdas.F90   2004-07-22 15:14:43.000000000 -0600
@@ -465,7 +465,7 @@
   allocate( rho(   1:mxa1 ) )
   rho = 0.0
   allocate( isef(  1:2,   1:mxa1*mt ) )    !***** check
-  isef = 0
+  isef = 0_i8knd
   allocate( jfq(   1:8,   0:ntal1 ) )
   jfq = 0
   allocate( laj(   1:(mlaj1+mxa1)*mtasks ) )
@@ -560,7 +560,7 @@
   allocate(  stt(   1:ntp,   1:ntal1*(npert1+1)*mt ) )
   stt = 0
   allocate(  nhsd(  1:nsp12, 1:ntal1*(npert1+1)*mt ) )
-  nhsd = 0
+  nhsd = 0_i8knd

   ! Call routine to allocate mesh tally arrays
   if( nmesh>0 ) then
@@ -969,10 +969,10 @@
   ! call msg_put( wwfa,   1,   (mgww1+mipt)*nwwma1)
   ! call msg_put(swwfa,   1,   (mgww1+mipt)*nwwma1)

-  call msg_put(  pac,   1, mipt*10*mxa1)
-  call msg_put(  pan,   1, 3*8*npn1)
+  call msg_put(  pac,   lpac+1, mipt*10*mxa1)
+  call msg_put(  pan,   lpan+1, 3*8*npn1)
   ! call msg_put(  pcc,   1,   3*mxa1*kpt1(2))
-  call msg_put(  pwb,   1, mipt*22*mxa1)
+  call msg_put(  pwb,   lpwb+1, mipt*22*mxa1)

   ! if( nsr==71 ) then
   !  call msg_put( sump,   1,   npert1)
@@ -1234,12 +1234,12 @@
      call msg_put( sump,   1,   npert1)
      sump(1:npert1)=zero
   endif
-
+
   call msg_put(  wns,   1,   2*(mxxs1/4))
   wns(1:2,1:(mxxs1/4))=zero

   call msg_put( isef,   1,   2*mxa1)
-  isef(1:2,1:mxa1)=0
+  isef(1:2,1:mxa1)=0_i8knd

   call msg_put( maze,   1,   3*nmaz1*sum(kpt))
   maze(1:3*nmaz1*sum(kpt))=0
@@ -1249,7 +1249,7 @@

   call msg_put(  ndr,   1,   mxe1)
   ndr(1:mxe1)=0
-
+
   if( ntal>0 ) then
    call msg_put( shsd,   1,   nspt*ntal*(npert+1))
    shsd(1:nspt,1:ntal*(npert+1))=zero
@@ -1258,7 +1258,7 @@
    stt(1:ntp,1:ntal*(npert+1))=zero
    call msg_put( nhsd,   1,   nsp12*ntal*(npert+1))
-   nhsd(1:nsp12,1:ntal*(npert+1))=0
+   nhsd(1:nsp12,1:ntal*(npert+1))=0_i8knd
   endif

   return
diff  -Naurd MCNP5/Source/src/sfiles.F90 MCNP5_new/Source/src/sfiles.F90
--- MCNP5/Source/src/sfiles.F90      2003-04-30 20:12:32.000000000 -0600
+++ MCNP5_new/Source/src/sfiles.F90  2004-07-22 15:14:43.000000000 -0600
@@ -6,6 +6,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/shade.F90 MCNP5_new/Source/src/shade.F90
--- MCNP5/Source/src/shade.F90       2003-04-30 20:12:34.000000000 -0600
+++ MCNP5_new/Source/src/shade.F90   2004-07-22 15:14:43.000000000 -0600
@@ -8,6 +8,7 @@
   use mcnp_debug
   use mcnp_plot
   use gkssim
+  use erprnt_mod
   implicit real(dknd) (a-h,o-z)

   real :: x(4),y(4)
@@ -33,6 +34,7 @@
     case ('tmp')
       sx = maxval(tmp)
       sn = minval(tmp,MASK = tmp.NE.0.0)
+
     case default
       sx = 1.0
       sn = 0.0
@@ -122,7 +124,7 @@
     il = ic
 95  continue
     s2 = extent(1)
-    if( dls<huge )  s2 = min( s+dls, real(extent(1),dknd) )
+    if( dls<huge_float )  s2 = min( s+dls, real(extent(1),dknd) )
     if( mat(ic)==0 .or. lc==1 )  go to 100
     if( lev/=levplt .and. levplt>=0 )  go to 100
     !
@@ -141,6 +143,7 @@
     case ('tmp')
       ik = min(nshades,int(nshades*(tmp(ic)-sn)/(sd))+1)
       ik = nshades + 1 - ik
+
     end select
     call gsfaci(ik,color_mode)
     !
diff  -Naurd MCNP5/Source/src/smmp.F90 MCNP5_new/Source/src/smmp.F90
--- MCNP5/Source/src/smmp.F90 2003-04-30 20:12:36.000000000 -0600
+++ MCNP5_new/Source/src/smmp.F90     2004-07-22 15:14:43.000000000 -0600
@@ -22,7 +22,7 @@

   ! Modules used:
   use mcnp_debug
-  use mcnp_params,   only: dknd, IDEF
+  use mcnp_params,   only: dknd,i8knd, IDEF

   implicit none
   public                               ! Default PUBLIC for this module.
@@ -36,7 +36,7 @@
   integer, private, parameter :: TWICE    =   2 !  locked twice.

   ! Private Variables:
-  integer,   private,allocatable :: lock_var(:) !  the lock variable.
```

```
+  integer(i8knd),   private,allocatable :: lock_var(:) !  the lock variable.
   integer,   private             :: nlocks=0    !  number of OMP locks.
   real(dknd),private,dimension(MAXTASKS) :: btme = 0.0
                                                 !  time waiting at barriers.
diff  -Naurd MCNP5/Source/src/sourcb.F90 MCNP5_new/Source/src/sourcb.F90
--- MCNP5/Source/src/sourcb.F90       2003-04-30 20:12:38.000000000 -0600
+++ MCNP5_new/Source/src/sourcb.F90   2004-07-22 15:14:43.000000000 -0600
@@ -523,7 +523,7 @@

   !$OMP CRITICAL (PRINT_OUTPUT)
   write(iuo,590) k,isef(kise+1,ji),ncl(ji)
-590 format(/ " only",i8,  " successes in",i10,  " tries in source cell",i6)
+590 format(/ " only",i8,  " successes in",i12,  " tries in source cell",i6)
   write(jtty,595) ic(1),ksd(1,ivdis(1)),hbln(1,1),xxx,yyy,zzz
   write( iuo,595) ic(1),ksd(1,ivdis(1)),hbln(1,1),xxx,yyy,zzz
 595 format(/," entry",i5," on SI",i3," for independant variable",a3,&
diff  -Naurd MCNP5/Source/src/sprob.F90 MCNP5_new/Source/src/sprob.F90
--- MCNP5/Source/src/sprob.F90        2003-04-30 20:12:40.000000000 -0600
+++ MCNP5_new/Source/src/sprob.F90    2004-07-22 15:14:43.000000000 -0600
@@ -8,6 +8,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)
   character ht*75
diff  -Naurd MCNP5/Source/src/srcsrf.F90 MCNP5_new/Source/src/srcsrf.F90
--- MCNP5/Source/src/srcsrf.F90       2003-04-30 20:12:42.000000000 -0600
+++ MCNP5_new/Source/src/srcsrf.F90   2004-07-22 15:14:43.000000000 -0600
@@ -8,6 +8,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/sread.F90 MCNP5_new/Source/src/sread.F90
--- MCNP5/Source/src/sread.F90        2003-04-30 20:12:42.000000000 -0600
+++ MCNP5_new/Source/src/sread.F90    2004-07-22 15:14:43.000000000 -0600
@@ -6,6 +6,9 @@

   use mcnp_global
   use mcnp_debug
+  use qttyin_mod, only: qttyin
+  use erprnt_mod, only: erprnt
+

   implicit real(dknd) (a-h,o-z)
   real(dknd) ::  aw(0:16)
@@ -26,9 +29,6 @@
     if( i>=5  )  read(ht,'(bn,i10)') ly(i-4)
   end do
   call zaid(2,ht,ixl(1,iex))
-#ifdef PCDOS
-  if(lockl)call pttyin
-#endif
   if( irup/=0 )  call qttyin(0,'"reading nuclide '//ht//'"')

   if( ly(1)/=2 ) then
diff  -Naurd MCNP5/Source/src/startp.F90 MCNP5_new/Source/src/startp.F90
--- MCNP5/Source/src/startp.F90       2003-04-30 20:12:44.000000000 -0600
+++ MCNP5_new/Source/src/startp.F90   2004-07-22 15:14:43.000000000 -0600
@@ -7,6 +7,7 @@

   ! Modules:
   use mcnp_global
+  use dxtran_mod
   use mcnp_debug
   use rmc_mod
```

```
@@ -142,7 +143,6 @@
     endif
   endif

-
   ! Report, count, and resample the source if energy > emx.
   if( erg>emx(ipt) ) then
     if( nsr==6 .or. nsr==71 ) then
@@ -159,6 +159,7 @@

   ! Print the initial parameters for the first fifty histories.

+160 continue
   !$OMP CRITICAL (PRINT_OUTPUT)
   ! Facet of macrobody surface source not printed to save space.
   if( npstc<=50 .and. ink(110)/=0 .and. krflg/=2 ) then
@@ -220,14 +221,7 @@
   endif

   ! Mark the particle if it is born inside a dxtran sphere.
-  idx = 0
-  do i = 1,ndx(ipt)
-    if( (dxx(ipt,1,i)-xxx)**2+(dxx(ipt,2,i)-yyy)**2+&
-      & (dxx(ipt,3,i)-zzz)**2<dxx(ipt,5,i) ) then
-      idx = i
-      exit
-    endif
-  enddo
+  idx = inside_dxtran_sphere()

   ! Increment source accounts and save some initial variables.
   paxtc(2,1,ipt) = paxtc(2,1,ipt)+wgt
diff  -Naurd MCNP5/Source/src/stuff.F90 MCNP5_new/Source/src/stuff.F90
--- MCNP5/Source/src/stuff.F90        2003-04-30 20:12:46.000000000 -0600
+++ MCNP5_new/Source/src/stuff.F90    2004-07-22 15:14:43.000000000 -0600
@@ -7,15 +7,14 @@
   use mcnp_debug
   use mcnp_input
   use fmesh_mod
+  use qttyin_mod, only: qttyin
+  use erprnt_mod, only: erprnt

   implicit real(dknd) (a-h,o-z)

   character hs*10,ht*10,hb*30,hc*181

   ! change mat from material names to material indexes.
-#ifdef PCDOS
-  if( lockl)  call pttyin
-#endif /*def.pcdos*/
   if( irup/=0 )  call qttyin(0,'  "beginning material processing."')
   DO_20: do i = 1,mxa
     do j = 1,nmat
@@ -65,8 +64,13 @@
         enddo
       enddo
       do i = 1,nmesh
-        if( fm(i)%mat /= nmt(mn) ) cycle
-        if( fm(i)%ipt == m ) goto 150
+        if( fm(i)%mat < 0 ) then
+          do j=1,fm(i)%nreact,2
+             if ( fm(i)%react(j) == mn .and. fm(i)%ipt == m ) goto 150
+          enddo
+        else if( fm(i)%mat == mn .and. fm(i)%ipt == m ) then
+          goto 150
+        endif
       enddo
       cycle DO_230

diff  -Naurd MCNP5/Source/src/sumary.F90 MCNP5_new/Source/src/sumary.F90
```

```
--- MCNP5/Source/src/sumary.F90      2003-04-30 20:12:46.000000000 -0600
+++ MCNP5_new/Source/src/sumary.F90  2004-07-22 15:14:43.000000000 -0600
@@ -6,15 +6,17 @@
   ! Print summaries of problem activity.

   use mcnp_global
+  use dxtran_mod
   use mcnp_debug
   use crit1_mod
   use fmesh_mod, only:  nmesh, fmesh_print
   use ra1_mod
   use ra2_mod
+  use erprnt_mod, only: erprnt, erprnt_i8_k2

   implicit real(dknd) (a-h,o-z)
   integer(i8knd)    :: i8_stride, i8_period
-  character(len=88) :: hy
+  character(len=92) :: hy
   character(len=26) :: ha(2)
   character(len=16) :: hg
   character(len=14) :: ht(4)
@@ -67,16 +69,16 @@
   if( nsr==6 .and. nrrs>=nrss )  t = np1
   if( nsr==6 .and. nrrs<nrss  )  t = npsr
   fpi = 1./max(one,t)
-  write(hy,'(20x,  "source particle weight for summary table normalization =",f12.2)')t
-  if( nint(t)==nps )  hy=' '
+  write(hy,'(20x,  "source particle weight for summary table normalization =",f16.2)')t
+  if( nint(t,i8knd)==nps )  hy=' '
   if( kc8<0 .or. kcheck/=0 )  hy(1:20)='(active cycles only)'
-  write(iuo,'(  "1problem summary ",a20,14x,a68/)')  hy(1:20),hy(21:88)
+  write(iuo,'(  "1problem summary ",a20,14x,a72/)')  hy(1:20),hy(21:92)

   ! If the run is finished, print the reason.
   call getidt(idtm)
   if( nst/=0 ) then
     write(rfq(1)(4:7),'(i4)')    ksr
-    write(rfq(2)(8:17),'(i10)')  nps
+    write(rfq(2)(8:19),'(i12)')  nps
     write(rfq(3)(11:14),'(i4)')  nerr
     if( ctme>0.0 .and. ctme<10. ) then
       write(rfq(5)(20:24),'(f5.1)') ctme
@@ -85,7 +87,7 @@
     endif
     write(rfq(7)(8:13),'(i6)')   kcy
     if( kcheck/=0 ) write(rfq(7)(8:15),'(i6, " a")')  kct
-    write(rfq(8)(9:18),'(i10)')  iptra(6)
+    write(rfq(8)(9:20),'(i12)')  iptra(6)
     do i = 1,11
       if( mod(nst/2**i,2)==0 )  cycle
       write(iuo,'(6x,  "run terminated ",'//rfq(i)//')')
@@ -227,14 +229,14 @@
     &   "maximum number ever in bank",i10/  " computer time in mcrun",&
     & f19.2, " minutes",12x,  "bank overflows to backup file",i8/&
     &   " source particles per minute",1pe22.4,/," random numbers generated",&
-    & 0pf26.0,tl1, " ",11x,  "most random numbers used was",i9,&
-    &   " in history",i9)
+    & 0pf26.0,tl1, " ",11x,  "most random numbers used was",i12,&
+    &   " in history",i12)
   endif

   ! Print conditional comments.
   call RN_query( stride = i8_stride,  period = i8_period )
-  if( nrnh(1)>0 ) then
-    call erprnt(1,2,2,int(i8_stride),nrnh(1),0,0,0, &
+  if( nrnh(1)>0_i8knd ) then
+    call erprnt_i8_k2(1,2,2,int(i8_stride),nrnh(1),0,0,0, &
       &  '" random number stride",i9," exceeded",i9," times."')
   endif
   if( nps*i8_stride>i8_period ) then
@@ -244,8 +246,8 @@
```

```
 240 format(/  " range of sampled source weights =",1pe11.4," to",e11.4)
   ie = 0
   do m = 1,mxa
-    if( isef(1,m)/=0 )  ie = 1
-    if( isef(1,m)/=0 )  write(iuo,260)  1.-isef(2,m)/(isef(1,m)+zero),ncl(m)
+    if( isef(1,m)/=0_i8knd )  ie = 1
+    if( isef(1,m)/=0_i8knd )  write(iuo,260)  1.-
real(isef(2,m),dknd)/real(isef(1,m),dknd),ncl(m)
   enddo
 260 format(/  " source efficiency =",f7.4,"  in cell",i6)
   je = 0
@@ -301,7 +303,7 @@
       else
         write(iuo,390) (nbal(i),i = 1,ltasks+1)
       endif
-390   format(/," number of histories processed by each task"/(10i10))
+390   format(/," number of histories processed by each task"/(10i12))
     endif
   endif

@@ -348,7 +350,7 @@

   ! Print information about results.
   call ra_result( mcheck )
-  if( nsr==71 .and. (nst/=0 .or. npp<0) .and. mcheck==0 ) then
+  if( nsr==71 .and. (nst/=0 .or. npp<0_i8knd) .and. mcheck==0 ) then
     call kprint
   endif

@@ -363,15 +365,15 @@
   endif

   ! Generate weight-window cards at the end of the run.
-  if( iwwg>0 .and. (nst/=0 .or. npp<0) ) then
+  if( iwwg>0 .and. (nst/=0 .or. npp<0_i8knd) ) then
     call avrout
   endif
-  if( icw/=0 .and. (nst/=0 .or. npp<0) ) then
+  if( icw/=0 .and. (nst/=0 .or. npp<0_i8knd) ) then
     call gmgww
   endif

   ! Print rnr if required, for quick evaluation of test problems.
-  if( dbcn(12)==0. .or. nst==0 .and. npp>=0 .or. nps<=0 )  return
+  if( dbcn(12)==0. .or. nst==0 .and. npp>=0_i8knd .or. nps<=0_i8knd )  return
   write(ha(1)(1:16),'(f16.0)')  dbcn(12)
   write(ha(2)(1:16),'(f16.0)')  rnr
   write(jtty,420)  ha(1)(1:15),ha(2)(1:15)
diff  -Naurd MCNP5/Source/src/sursrc.F90 MCNP5_new/Source/src/sursrc.F90
--- MCNP5/Source/src/sursrc.F90      2003-04-30 20:12:48.000000000 -0600
+++ MCNP5_new/Source/src/sursrc.F90  2004-07-22 15:14:43.000000000 -0600
@@ -9,6 +9,7 @@
   use mcnp_debug

   implicit real(dknd) (a-h,o-z)
+  integer(i8knd) :: nr, n2

   nr = 0
   nt = 0
@@ -124,7 +125,7 @@
     if( n2/=npsr )  then
       n2 = npsr
       if( rang()>snit ) then
-        n2 = -1
+        n2 = -1_i8knd
         nssi(8) = nssi(8)+1
         go to 340
       endif
@@ -200,7 +201,7 @@
   if( nt==0 ) then
     ! save the first accepted particle to start the history with.
     nt = 1
-    ntss = ntss+1
+    ntss = ntss+1_i8knd
     if( mcnp_opt_multp .and. ns/=0 )  go to 330
     npb = 1
     call savpar(1,0)
@@ -232,12 +233,12 @@
     if( npsr==abs(ssb(1)) )  go to 10

     ! all tracks associated with this history have been read.
-    nqss = nqss+1
+    nqss = nqss+1_i8knd
     n2 = 0
     if( nt==0 ) go to 10
   else
     ! terminate the problem when all tracks are read.
-    nqss = nqss+1
+    nqss = nqss+1_i8knd
     if( twss<=0. ) then
       call expirx(0,'sursrc','no weight is on the '//rssa//' file.')
       return
@@ -248,7 +249,7 @@
     endif
     if( nt==0 ) then
       nomore = 1
-      nps = nps-1
+      nps = nps-1_i8knd
     endif
   endif

diff  -Naurd MCNP5/Source/src/tallmg.F90 MCNP5_new/Source/src/tallmg.F90
--- MCNP5/Source/src/tallmg.F90      2003-04-30 20:12:48.000000000 -0600
+++ MCNP5_new/Source/src/tallmg.F90  2004-07-22 15:14:43.000000000 -0600
@@ -13,6 +13,7 @@
   ! Modules:
   use mcnp_global
   use mcnp_debug
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/talloc.F90 MCNP5_new/Source/src/talloc.F90
--- MCNP5/Source/src/talloc.F90      2003-04-30 20:12:48.000000000 -0600
+++ MCNP5_new/Source/src/talloc.F90  2004-07-22 15:14:43.000000000 -0600
@@ -7,6 +7,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)
   integer, parameter ::  ki(8,3) = &
diff  -Naurd MCNP5/Source/src/tallyd.F90 MCNP5_new/Source/src/tallyd.F90
--- MCNP5/Source/src/tallyd.F90      2003-04-30 20:12:50.000000000 -0600
+++ MCNP5_new/Source/src/tallyd.F90  2004-07-22 15:14:43.000000000 -0600
@@ -10,7 +10,7 @@
   use mcnp_debug

   implicit real(dknd) (a-h,o-z)
-  integer(i8knd) :: i8_count
+  integer(i8knd) :: i8_count,itmp2
   integer        :: ik(8)
   logical        :: test_td

@@ -244,12 +244,12 @@
       if( jptal(8,ital)<4 ) then
         if( ddg(1,idet)<0. ) then
          s = -ddg(1,idet)/(wgt*psc)
-        elseif( ddg(1,idet)>0. .and. npstc>200 ) then
+        elseif( ddg(1,idet)>0. .and. npstc>200_i8knd ) then
          s =  ddg(1,idet)*ddn(24,idet)/(wgt*psc)
         endif
         if( ipsc/=11 )  s = s*2.*pie*dd**2
       else
-        if( jptal(8,ital)==4 .and. npstc>npsmg .and. npsmg/=0 .and. &
+        if( jptal(8,ital)==4 .and. npstc>npsmg .and. npsmg/=0_i8knd .and. &
           & (ipsc==3 .or. ipsc==10 .or. ipsc==11 .or. ipsc==12 .or. &
           & ipsc==101) ) then
           amfp = 0.
@@ -366,7 +366,7 @@
         ca = tme+tds(ll+1)
         cd = tme+tds(ll+2)
         ibt = 1
-        b = -huge
+        b = -huge_float
       endif
 320   continue
       a = b
@@ -439,10 +439,10 @@
             itmp2 = npsout
           else
             itmp1 = mxf
-            itmp2 = nps - 1
+            itmp2 = nps - 1_i8knd
           endif
 
-          test_td = jptal(8,ital)==4 .and. npstc>npsmg .and. npsmg/=0 .and. &
+          test_td = jptal(8,ital)==4 .and. npstc>npsmg .and. npsmg/=0_i8knd .and. &
             &      (ipsc==3 .or. ipsc==10 .or. ipsc==11 .or. ipsc==12 .or. ipsc==101)
 
           is = ibs
diff  -Naurd MCNP5/Source/src/tally.F90 MCNP5_new/Source/src/tally.F90
--- MCNP5/Source/src/tally.F90        2003-04-30 20:12:50.000000000 -0600
+++ MCNP5_new/Source/src/tally.F90    2004-07-22 15:14:43.000000000 -0600
@@ -18,6 +18,9 @@
   ! return if kcode problem is not settled.
   if( kc8>0 )  return
 
+  ! Check for speed tally for lattice usage
+  if ( flag_speed_tally_used == 1 ) goto 800
+
   ! warn if tally made by both dxtran and non-dxtran particles.
   if( idx+1/=itds(lo-1) ) then
     if( itds(lo-1)>=0 ) then
@@ -66,7 +69,7 @@
       l = jxs(1,mgegbt(1))+2*jgm(1)-1
       do ibu_tmp = 1,iptal(3,4,ital)-1
         ibu = ibu_tmp
-        if( abs(tds(iptal(3,1,ital)+ibu)-xss(l+jgp))<=.001*xss(l+jgp) )  go to 60
+        if( abs(tds(iptal(3,1,ital)+ibu)-xss(l+jgp))<=.001_dknd*xss(l+jgp) )  go to 60
       enddo
       ibu = ibu_tmp
       go to 700
@@ -89,7 +92,7 @@
           t2 = t1**2+rang()**2
           if( t2<=1. ) exit
         enddo
-        erg = erg + .60056120439322*(tds(l+1)+tds(l+2)*sqrt(max(zero,&
+        erg = erg + .60056120439322_dknd*(tds(l+1)+tds(l+2)*sqrt(max(zero,&
           & erg+tds(l+3)*erg**2)))*sqrt(-log(t2)/t2)*t1
       endif
 
@@ -214,7 +217,7 @@
       mk = mat(icl)
       if( mcal==0 ) then
         do m = jmd(mk),jmd(mk+1)-1
-          h2 = huge
+          h2 = huge_float
           i = lme(1,m)
           if( jxs(21,i)==0 )  cycle
           l = jxs(21,i)
@@ -231,7 +234,7 @@
```

```
           if( npert/=0 )  call pertc(2,mk,m,-6,lx,h2)
           h = h+h2
         enddo
-        if( npert/=0 .and. h2==huge )  call pertc(2,mk,jmd(mk+1)-1,-6,lx,zero)
+        if( npert/=0 .and. h2==huge_float )  call pertc(2,mk,jmd(mk+1)-1,-6,lx,zero)
       else
         ! multigroup fission heating.
         do m = jmd(mk),jmd(mk+1)-1
@@ -299,7 +302,7 @@
     enddo
     db = min(dr,dti(k))
     if( db==dr )  go to 470
-    dti(k) = huge
+    dti(k) = huge_float
     do j = 1,n
       if( abs(lja(abs(lca(ig))+j-1))==iti(k) )  exit
     enddo
@@ -322,7 +325,7 @@
     ca = tme+tds(ll+1)
     cd = tme+tds(ll+2)
     ibt = 1
-    b = -huge
+    b = -huge_float
     if( jptal(2,ital)>=4 ) then
       dt = (dc-da)/vel
       tt = ta*vel
@@ -512,4 +515,27 @@
   tme = tv
   erg = es
   return
+!   this is the special speed tally treatment for lattices
+800 lx=lo+2                         ! From: do all tallies that include this cell or
surface
+    ta=wgt                          !   include weight of particle in tally
+    j=-mfl(1,nint(udt(7,lev-1)))    ! effectively from do surface or cell bins
+    n8=nint(udt(8,lev-1))
+    n9=nint(udt(9,lev-1))
+    n10=nint(udt(10,lev-1))
+    do ml=1,itds(lo)                ! From: do all tallies that include this cell or
surface
+      ital=itds(lx-1)
+      n1=itds(lx)
+      ir=n8-laf(j+1,1)+laf(j+1,2)*(n9-&
+        & laf(j+2,1)+laf(j+2,2)*(n10-laf(j+3,1)))+1
+      j7=ktal+jptal(5,ital)+iptal(2,5,ital)*(itds(lx+ir)-1)+1
+      tb=dosef(ta)                  ! Include DE, DF cards into tally result
+      td=tb*dr*tds(iptal(5,2,ital)+1)
+      if(tal(j7) .eq. 0._dknd)then
+       jtls=jtls+1
+       if(jtls .le. ktls) tal(ktal+nmxf*mxf+jtls)=j7
+      endif
+      tal(j7)=tal(j7)+td
+    lx=lx+n1+2
+   enddo
+   return
 end subroutine tally
diff  -Naurd MCNP5/Source/src/tallyh.F90 MCNP5_new/Source/src/tallyh.F90
--- MCNP5/Source/src/tallyh.F90      2003-04-30 20:12:50.000000000 -0600
+++ MCNP5_new/Source/src/tallyh.F90  2004-07-22 15:14:43.000000000 -0600
@@ -5,6 +5,7 @@
   ! print the tally heading.
   use mcnp_global
   use mcnp_debug
+  use qttyin_mod

   implicit real(dknd) (a-h,o-z)

@@ -51,7 +52,7 @@
   if( nps==0 )  write(iuo,10) k
 10 format(/"1tally",i4,94x,  "print table 30")
   if( nps>0 )  write(iuo,20) k,nps
```

```
-20 format(/"1tally",i4,8x, "nps =",i9)
+20 format(/"1tally",i4,8x, "nps =",i12)
   if( l/=0 ) then
     n = itds(l)
     hv = '+'
@@ -69,9 +70,6 @@
   iy = jptal(2,ital)
   js = min(jptal(4,ital),1)
   if( iy==5 )  hk(js+1,5)(20-2*js:24-2*js) = hd(jptal(8,ital))
-#ifdef PCDOS
-  if( lockl) call pttyin
-#endif /*def.pcdos*/
   if( irup/=0 )  call qttyin(iy,'  "processing tally",i5')
   hv = ' '
   if( nps==0 .or. iptal(5,2,ital)/=0 .or. &
diff  -Naurd MCNP5/Source/src/tallyp.F90 MCNP5_new/Source/src/tallyp.F90
--- MCNP5/Source/src/tallyp.F90      2003-04-30 20:12:52.000000000 -0600
+++ MCNP5_new/Source/src/tallyp.F90  2004-07-22 15:14:43.000000000 -0600
@@ -26,7 +26,7 @@

   ! find how many lines are in the tfc charts.
   do i=1,20
-    if( npc(i)/=0 )  nn=i
+    if( npc(i)/=0_i8knd )  nn=i
   end do

   ! do all of the tallies in the problem.
@@ -49,9 +49,9 @@
       do i=1,8
         ip(i) = i
       end do
-      nhsd(nsp+8,it) = 1
+      nhsd(nsp+8,it) = 1_i8knd
       do i=4,7
-        nhsd(nsp+5+i,it) = 0
+        nhsd(nsp+5+i,it) = 0_i8knd
       end do
       ac = 0.1
       if( iy==5 )  ac=.05
@@ -146,13 +146,13 @@
                         tpp(5+i) = 0.
                         vv(i) = 0.
                         vv(i+5) = 0.
-                        if( tal(j)==0. )  nhsd(nsp+10,it)=nhsd(nsp+10,it)+1
+                        if( tal(j)==0. )  nhsd(nsp+10,it)=nhsd(nsp+10,it)+1_i8knd
                         if( tal(j)==0. .or. tal(mxf+j)==0. )  cycle DO_80

                         ! calculate estimated relative error using first two moments.
                         t = tal(j)
                         tpp(5+i) = sqrt(max(zero,min(tal(mxf+j)/t**2-fpi,one)))
-                        if( tpp(5+i)>=ac )  nhsd(nsp+11,it)=nhsd(nsp+11,it)+1
+                        if( tpp(5+i)>=ac )  nhsd(nsp+11,it)=nhsd(nsp+11,it)+1_i8knd

                         ! assume a very small rel error is round off and set to zero.
                         if( tpp(5+i)<=1.e-7 )  tpp(5+i)=0.
@@ -176,7 +176,7 @@
                         if( t2<=0. .or. t1<=0. .or. tal(mxf+j)<=0 )  cycle DO_80
                         vv(i+5) = max(zero,min(t2/t1**2,one))
                         if( t2<1.e-30 .and. vv(i+5)==one )  vv(i+5)=0.
-                        if( vv(i+5)>=0.1 )  nhsd(nsp12,it)=nhsd(nsp12,it)+1
+                        if( vv(i+5)>=0.1 )  nhsd(nsp12,it)=nhsd(nsp12,it)+1_i8knd

                         ! calculate asymmetric confidence interval shift(w/scale
factor).
                         ! see a-1/87-749 for the confidence interval center shift.
@@ -200,13 +200,13 @@
         end do DO_110

         ! print detector diagnostics with the detector if possible.
-        if( ip(1)==1 .and. iy==5 .and. nps/=0 )  call dddiag(i1,0)
+        if( ip(1)==1 .and. iy==5 .and. nps/=0_i8knd )  call dddiag(i1,0)
```

```
       end do DO_120

       ! otherwise print detector diagnostics at the end of the tally.
       if( ip(1)==1 )  go to 140
 125   continue
-      if( iy==5 .and. nps/=0 ) then
+      if( iy==5 .and. nps/=0_i8knd ) then
         do i1=1,iptal(1,3,ital)
           call dddiag(i1,1)
         end do
@@ -242,7 +242,7 @@
       endif
     end do
   end do
-  if( nps<=0 )  return
+  if( nps<=0_i8knd )  return

   ! print tally fluctuation charts.
   call ptfc(nn)
diff  -Naurd MCNP5/Source/src/talshf.F90 MCNP5_new/Source/src/talshf.F90
--- MCNP5/Source/src/talshf.F90      2003-04-30 20:12:54.000000000 -0600
+++ MCNP5_new/Source/src/talshf.F90  2004-07-22 15:14:43.000000000 -0600
@@ -9,6 +9,7 @@
   ! Modules used:
   use mcnp_global
   use mcnp_debug
+  use hpsort_mod, only: hpsort

   implicit real(dknd) (a-h,o-z)

@@ -35,24 +36,24 @@
       j = max(1,min(nsp,int(10.*log10(abs(t))+nhb)))
       if ( t<=0 ) then
         j = 1
-        nhsd(knhs+nsp+1,it) = nhsd(knhs+nsp+1,it)+1
+        nhsd(knhs+nsp+1,it) = nhsd(knhs+nsp+1,it)+1_i8knd
        shsd(kshs+nsp+1,it) = shsd(kshs+nsp+1,it)+t
       end if
-      nhsd(knhs+j,it) = nhsd(knhs+j,it)+1
-      nhsd(knhs+nsp+2,it) = nhsd(knhs+nsp+2,it)+1
+      nhsd(knhs+j,it) = nhsd(knhs+j,it)+1_i8knd
+      nhsd(knhs+nsp+2,it) = nhsd(knhs+nsp+2,it)+1_i8knd
       shsd(kshs+j,it) = shsd(kshs+j,it)+t
       shsd(kshs+nsp+2,it) = shsd(kshs+nsp+2,it)+t

       ! save the ntp largest,smallest history tallies for each tfc bin.
       if( t<=shsd(kshs+nsp+ntp+6,it) )  cycle DO_10_1
       if( nhsd(knhs+nsp+6,it)==ntp ) then
-        nhsd(knhs+nsp+7,it) = nhsd(knhs+nsp+7,it)+1
+        nhsd(knhs+nsp+7,it) = nhsd(knhs+nsp+7,it)+1_i8knd
         stt(kstt+nhsd(knhs+nsp+7,it),it) = t
-        if( nhsd(knhs+nsp+7,it)==ntp ) then
+        if( nhsd(knhs+nsp+7,it)==int(ntp,i8knd) ) then
           call hpsort(it,knhs+nsp+5, kshs+nsp+5,ntp,ntp)
         endif
       else
-        nhsd(knhs+nsp+6,it) = nhsd(knhs+nsp+6,it)+1
+        nhsd(knhs+nsp+6,it) = nhsd(knhs+nsp+6,it)+1_i8knd
         shsd(kshs+nsp+5+nhsd(knhs+nsp+6,it),it) = t
       endif
     end do DO_10_1
diff  -Naurd MCNP5/Source/src/tekdvr.F90 MCNP5_new/Source/src/tekdvr.F90
--- MCNP5/Source/src/tekdvr.F90      2003-04-30 20:12:56.000000000 -0600
+++ MCNP5_new/Source/src/tekdvr.F90  2004-07-22 15:14:43.000000000 -0600
@@ -25,9 +25,6 @@
   data np/0,1,3*0,1,4*0,1,1,0,1,4*0,1/,nn/9*0,1,1,0,1,5*0,1/
   !
   ! specify the window and the character expansion factor.
-#ifdef LAHEY
-  call IGrAreaClear(0.0,0.0,1.0,1.0)
-#endif /*def.lahey*/
```

```
   et(1) = extent(1)*(1.+.2*(1-jvp))
   et(2) = extent(2)*(1.+.2*(1-jvp))
   call gswn(1,-et(1)*(5-2*jvp)/3.,et(1),-et(2),et(2))
@@ -110,14 +107,11 @@
   ! write the legend if the viewport is rectangular.
   yv = .96*et(2)
   yd = 1.5*sch*et(2)
-#ifdef LAHEY
-  if( jgf==0 )  yd = yd*2.0
-#endif /*def.lahey*/
   call gschh(real(sch)*et(2))
   if( jvp/=0 )  go to 110
   call getidt(idtm)
   xv = -1.64*et(1)
-  call gtx(xv,yv,idtm)
+    call gtx(xv,yv,idtm)
   i = 33
   do iz = 1,33
     if(aid(iz:iz) == ' ')i=iz
@@ -171,9 +165,7 @@
 110 continue
     xhom = -.33*(5-2*jvp)*et(1)
   yhom = yv-yd
-#ifndef LAHEY
   call gxhome(real(xhom),real(yhom))
-#endif /*-def.lahey*/
   return
 end subroutine tekdvr
 #endif /*def.plot*/
diff  -Naurd MCNP5/Source/src/tpefil.F90 MCNP5_new/Source/src/tpefil.F90
--- MCNP5/Source/src/tpefil.F90      2003-04-30 20:12:58.000000000 -0600
+++ MCNP5_new/Source/src/tpefil.F90  2004-07-22 15:14:43.000000000 -0600
@@ -52,11 +52,11 @@
   ! ---------------
   ! mm=3 -- Write the fixed data and the first dump.
   case( 3 )
-    write(iur) aplace,kplace,lplace,gfixcm,jfixcm,i8fixcm
+    write(iur) aplace,kplace,lplace,gfixcm,i8fixcm,jfixcm
    call fdac_write(iur)

    ! Zero the tfc bin history score arrays and tally blocks.
-    nhsd( 1:nsp12, 1:ntal*(npert+1) ) = 0
+    nhsd( 1:nsp12, 1:ntal*(npert+1) ) = 0_i8knd
    shsd( 1:nspt,  1:ntal*(npert+1) ) = 0.
    tal(  1:nmxf*mxf+ktls ) = 0.

@@ -87,9 +87,9 @@
    write(jtty,70) knod,runtpe,nps,hp(1:15),cts/60.,hp(17:31)

 60  format(/1x,119('*')/ " dump no.",i5, " on file ",a8,5x, &
-      & "nps =",i10,5x, "coll =",a15,5x, "ctm =",f8.2,5x, "nrn =",a15/)
-70  format( " dump",i5, " on file ",a8,3x, "nps =",i10,4x, &
-      &  "coll =",a15/ 30x, "ctm =",f10.2,5x, "nrn =",a15)
+      & "nps =",i12,5x, "coll =",a15,5x, "ctm =",f12.2,3x, "nrn =",a18/)
+70  format( " dump",i5, " on file ",a8,3x, "nps =",i12,4x, &
+      &  "coll =",a15/ 30x, "ctm =",f12.2,3x, "nrn =",a18)

  ! ---------------
  ! mm=5 -- Open existing runtpe and read it for continue run.
@@ -115,7 +115,7 @@
      read(iur )  id
    enddo

-    read(iur) aplace,kplace,lplace,gfixcm,jfixcm,i8fixcm
+    read(iur) aplace,kplace,lplace,gfixcm,i8fixcm,jfixcm

    ! set up dynamic storage
    if( allocated( gbnk ) )  deallocate( gbnk )
@@ -127,7 +127,7 @@
      mt = 1
    endif
```

```
     allocate( gbnk( 1:mbnk*mtasks ) )
-    allocate( ibnk( 1:(nbmx*(lpblcm+2*abs(iunr))+1)*mtasks ) )
+    allocate( ibnk( 0:(nbmx*(lpblcm+2*abs(iunr))+1)*mtasks ) )
     allocate( tal(  1:(nmxf*mxf+ktls)*mt ) )
     gbnk = 0.0
     ibnk = 0
@@ -175,10 +175,10 @@
     ! Print comments about starting from the restart dump.
      write(iuo,150) knod,runtpe,nps,cts/60.,probid
 150  format(/  " starting from dump no.",i5,  " from file ",a8,5x,&
-      &  "nps =",i10,5x, "ctm =",f8.2,5x, "probid = ",a19/)
+      &  "nps =",i12,5x, "ctm =",f12.2,5x, "probid = ",a19/)
     write(jtty,160) knod,runtpe,nps,cts/60.,probid
-160  format(  " starting from dump",i5, "  file ",a8, " nps =",i10,&
-      &  " ctm =",f8.2/  " probid = ",a19)
+160  format(  " starting from dump",i5, "  file ",a8, " nps =",i12,&
+      &  " ctm =",f12.2/  " probid = ",a19)

   ! ---------------
   ! mm=6 -- Make sure of write access to runtpe.
@@ -213,7 +213,7 @@
     do i = 1,mxe
       read(iur )  id
     enddo
-    read(iur) aplace,kplace,lplace,gfixcm,jfixcm,i8fixcm
+    read(iur) aplace,kplace,lplace,gfixcm,i8fixcm,jfixcm

     call dyn_allocate

diff  -Naurd MCNP5/Source/src/track.F90 MCNP5_new/Source/src/track.F90
--- MCNP5/Source/src/track.F90       2003-04-30 20:12:58.000000000 -0600
+++ MCNP5_new/Source/src/track.F90   2004-07-22 15:14:43.000000000 -0600
@@ -11,18 +11,16 @@

   use mcnp_global
   use mcnp_debug
+  use qttyin_mod, only: qttyin

   implicit real(dknd) (a-h,o-z)
   real(dknd) :: dl(0:mxlv)
   integer    :: jp(0:mxlv)

   ! set up to do the cell in the current level.
-#ifdef PCDOS
-  if( lockl )  call pttyin
-#endif /*def.pcdos*/
   if( irup/=0 ) then
     if( .not.mcnp_opt_multt ) then
-      call qttyin(-1,' "nps =",i9,5x,  "collisions =",i6')
+      call qttyin(-1,' "nps =",i12,5x,  "collisions =",i6')
     endif
   endif
   ll = lev
@@ -30,7 +28,7 @@

   ! find the distance to boundary, dl, in this cell.
 10 continue
-  dl(ll) = huge
+  dl(ll) = huge_float
   nlt = 0

   ! calculate all intersections with all surfaces of this cell.
@@ -318,11 +316,11 @@
           if( dti(m)<dti(n1) )  n1 = m
         enddo
         dl(ll) = dti(n1)
-        if( dl(ll)==huge ) exit
+        if( dl(ll)==huge_float ) exit
         do j1 = j2,j3
           if( abs(lja(j1))==iti(n1) )  lgc(j1-j2+1) = 1-lgc(j1-j2+1)
         enddo
```

```
-        dti(n1) = huge
+        dti(n1) = huge_float
         if( lgeval(lgc,j3-j2+1)==0 ) then
           jp(ll) = iti(n1)
           exit
@@ -370,6 +368,6 @@
   jap = jp(levp)
   !
   ! return with error indicator if no intersection was found.
-  if( dls==huge )  kdb = 2
+  if( dls==huge_float )  kdb = 2
   return
 end subroutine track
diff  -Naurd MCNP5/Source/src/trfmat.F90 MCNP5_new/Source/src/trfmat.F90
--- MCNP5/Source/src/trfmat.F90      2003-04-30 20:13:00.000000000 -0600
+++ MCNP5_new/Source/src/trfmat.F90  2004-07-22 15:14:43.000000000 -0600
@@ -7,6 +7,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

@@ -18,7 +19,7 @@
   tr = 0.0
   do i=1,14
     tr(i) = trf(i,jt)
-    if( tr(i)/=huge .and. i>=5 .and. i<=13 ) nb=nb+1
+    if( tr(i)/=huge_float .and. i>=5 .and. i<=13 ) nb=nb+1
   end do
   if( tr(14)==0. )  tr(14)=1.

@@ -37,7 +38,7 @@
     !  >>>>>  nb=3 -- one vector.
     do k=1,2
       do j=1,3
-        if( tr(3*j+2)/=huge .and. tr(3*j+3)/=huge .and. tr(3*j+4)/=huge ) then
+        if( tr(3*j+2)/=huge_float .and. tr(3*j+3)/=huge_float .and. 
tr(3*j+4)/=huge_float ) then
           r = sqrt(tr(3*j+2)**2+tr(3*j+3)**2+tr(3*j+4)**2)
           if( r==0. )  go to 320
           do i=2,4
@@ -67,7 +68,7 @@
     !  >>>>>  nb=6 -- two vectors in the same system.
     do k=1,2
       do j=1,3
-        if( tr(3*j+2)==huge .and. tr(3*j+3)==huge .and. tr(3*j+4)==huge ) then
+        if( tr(3*j+2)==huge_float .and. tr(3*j+3)==huge_float .and. 
tr(3*j+4)==huge_float ) then
           do l=1,2
             m = mod(j+l-1,3)+1
             r = sqrt(tr(3*m+2)**2+tr(3*m+3)**2+tr(3*m+4)**2)
@@ -99,17 +100,17 @@
   case( 5 )
     !  >>>>>  nb=5 -- one vector in each system.
     do i1=1,3
-      if( tr(3*i1+2)/=huge .and. tr(3*i1+3)/=huge .and. tr(3*i1+4)/=huge ) then
+      if( tr(3*i1+2)/=huge_float .and. tr(3*i1+3)/=huge_float .and. 
tr(3*i1+4)/=huge_float ) then
        do j1=1,3
-          if( tr(j1+4)/=huge .and. tr(j1+7)/=huge .and. tr(j1+10)/=huge ) then
+          if( tr(j1+4)/=huge_float .and. tr(j1+7)/=huge_float .and. 
tr(j1+10)/=huge_float ) then
             r = tr(3*i1+j1+1)**2
             do i=5,13
-              if( tr(i)/=huge )  r=r+tr(i)**2
+              if( tr(i)/=huge_float )  r=r+tr(i)**2
             end do
             r = sqrt(.5*r)
             if( r==0. )  go to 320
             do i=5,13
-              if( tr(i)/=huge )  tr(i)=tr(i)/r
+              if( tr(i)/=huge_float )  tr(i)=tr(i)/r
             end do
             if( tr(3*i1+j1+1)**2==1. )  go to 320
             i2 = mod(i1,3)+1
diff  -Naurd MCNP5/Source/src/trfsrf.F90 MCNP5_new/Source/src/trfsrf.F90
--- MCNP5/Source/src/trfsrf.F90      2003-04-30 20:13:00.000000000 -0600
+++ MCNP5_new/Source/src/trfsrf.F90  2004-07-22 15:14:43.000000000 -0600
@@ -6,6 +6,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/trnspt.F90 MCNP5_new/Source/src/trnspt.F90
--- MCNP5/Source/src/trnspt.F90      2003-04-30 20:13:00.000000000 -0600
+++ MCNP5_new/Source/src/trnspt.F90  2004-07-22 15:14:43.000000000 -0600
@@ -13,6 +13,7 @@

   implicit real(dknd) (a-h,o-z)

+
   call utask

   call secnd(t1)
@@ -26,7 +27,7 @@
     do while( .not. time_to_stop() )

       ! Run the next history.
-      nps   = nps+1
+      nps   = nps+1_i8knd
       npstc = nps

       if( ltasks<=1 )  nbal(ktask+1) = nbal(ktask+1)+1
@@ -64,15 +65,15 @@
     ! see if it's time to stop running histories &
     ! return to do output, a dump, plotting, handle an
     ! interrupt, or terminate the problem...
-
     logical :: time_to_stop
+    integer(i8knd) :: n,ne

     time_to_stop = .false.

     ! Quit if enough particles have been started from the source.
-    if( nsr==6    .and. nrrs>=nrss            )  nst = nst+1024
-    if( ltasks<=1 .and. nps>=npp .and. npp/=0 )  nst = nst+4
-    if( ltasks>1  .and. nps>=npp              )  nst = nst+4   ! npp may be 0
+    if( nsr==6    .and. nrrs>=nrss                  )  nst = nst+1024
+    if( ltasks<=1 .and. nps>=npp .and. npp/=0_i8knd )  nst = nst+4
+    if( ltasks>1  .and. nps>=npp                    )  nst = nst+4   ! npp may be 0

     ! Quit if too many particles are lost.
     if( nerr>=lost(1) )  nst = nst+8
@@ -113,21 +114,21 @@
         endif

         ! Set output flags at specified histories.
-        if( prn>0. .and. mod(n,max(1,int(prn)))==0 )  mpc = 1
-        if( dmp>0. .and. mod(n,max(1,int(dmp)))==0 )  mdc = 1
+        if( prn>0. .and. mod(n,int(max(1_i8knd,int(prn,i8knd)),i8knd))==0 )  mpc = 1
+        if( dmp>0. .and. mod(n,int(max(1_i8knd,int(dmp,i8knd)),i8knd))==0 )  mdc = 1
         if( kmplot/=0 .or. &
-          & (krtm/=0 .and. freq>0. .and. mod(n,max(1,int(freq)))==0) )  mrm = 1
+          & (krtm/=0 .and. freq>0. .and. mod(n,int(max(1,int(freq)),i8knd))==0) )  mrm =
1
 30  continue
```

```
     ! Check the time once a second and for output.
     call secnd(t2)
     if( nps>=ntc1 .or. nst+jtfc+mpc+mdc+mrm/=0 .or. &
       & (nps==npsmg .and. npsmg/=0 .and. npsout/=npsmg) .or. &
-      & (nsr==71 .and. nsa==0) .or. (nps==200 .and. npsout/=200) ) then
+      & (nsr==71 .and. nsa==0) .or. (nps==200_i8knd .and. npsout/=200_i8knd) ) then
       if( t2<cp2(ktask+1)-1. ) then
         write(jtty,40) kod,cp2(ktask+1)/60.,t2/60.
         write( iuo,40) kod,cp2(ktask+1)/60.,t2/60.
-40      format(  " restarted ",a8,8x,  "old time =",f7.2,4x,"new time =",f7.2)
+40      format(  " restarted ",a8,8x,  "old time =",f9.2,4x,"new time =",f9.2)
         nrc = nrc+1
       else
         cp1 = cp1+(t2-cp2(ktask+1))/ntasks
@@ -155,8 +156,8 @@
       ! set flag to stop running histories if output is required.
       if(  (nsr==71 .and. nsa==0)        .or. &
         &  jtfc+nst/=0                   .or. &
-        &  (nps==npsmg .and. npsmg/=0 .and. npsout/=npsmg) .or. &
-        &  (nps==200  .and. npsout/=200) .or. &
+        &  (nps==npsmg .and. npsmg/=0_i8knd .and. npsout/=npsmg) .or. &
+        &  (nps==200_i8knd  .and. npsout/=200_i8knd) .or. &
         &  (ltasks<=1 .and. mpc+mdc+mrm/=0)  )  time_to_stop = .true.
     endif
     return
diff  -Naurd MCNP5/Source/src/tskcom.F90 MCNP5_new/Source/src/tskcom.F90
--- MCNP5/Source/src/tskcom.F90      2003-04-30 20:13:02.000000000 -0600
+++ MCNP5_new/Source/src/tskcom.F90  2004-07-22 15:14:43.000000000 -0600
@@ -10,11 +10,37 @@
   use mcnp_params
   use mcnp_debug
   use fixcom, only : ltskpt
-  use varcom, only : nvarsw, lvarsw
+  use varcom, only : nvarsw, l8varsw, lvarsw

   implicit none

-  ! Swept tskcom reals - order is important, see varcom.
+
+  !--------------------------------------------------------------------------------------
--
+  !  (1) parameters for the length of the 3 portions of /tskcm/
+  ! Swept-tskcom,itskpt lengths and equivalences.
+
+  integer,parameter :: ntskcm = & != Size of floating-point part of /TSKCM/.
+    & nvarsw + 25*(1) + 3*(3) + 1*(5) + 1*(7) + 1*(64)&
+    & + 1*(3*11) + 2*(mipt) + 1*(mlgc) + 1*(3*maxv) + 4*(10*mxlv+10)
+
+  integer,parameter :: l8tskcm =    & != Size of INTEGER*8 part of /TSKCM.
+    & l8varsw + 2*(1)
+
+  integer,parameter :: ltskcm = & != Size of integer part of /TSKCM/.
+    & lvarsw + 43*(1) + 1*(5*(1+mxlv)) + 1*(maxv) &
+    & + 1*(mipt) + 1*(mlgc) + 1*(mlgc+1) + 1*(nlocks)
+
+  !--------------------------------------------------------------------------------------
--
+  !  (2) declarations for 3 arrays equivalenced to /tskcm/
+
+  real(dknd)     :: gtskcm(ntskcm)    != Equivalence to real      part of /TSKCM/.
+  integer(i8knd) :: i8tskcm(l8tskcm)  != Equivalence to integer*8 part of /TSKCM/.
+  integer        :: jtskcm(ltskcm)    != Equivalence to integer   part of /TSKCM/.
+
+  !--------------------------------------------------------------------------------------
--
+  !  (3) declarations for /tskcm/ REALS
+
+  ! Swept tskcom reals - order is important, see varcom - length = nvarsw
   real(dknd) :: colltc(mipt)            != Task copy of COLL.
   real(dknd) :: eacctc(4)               != Task copy of EACC.
   real(dknd) :: paxtc(6,21,mipt)        != Task copy of PAX.
```

```
@@ -25,8 +51,6 @@
   real(dknd) :: tmavtc(mipt,3)          != Task copy of TMAV.
   real(dknd) :: wgtstc(2)               != Task copy of WGTS.

-  common /tskcm/ colltc, eacctc, paxtc, rlttc, rnrtc, smultc, sumktc, tmavtc, wgtstc
-
   ! Non-swept tskcom reals.
   real(dknd) :: amfp                    != Mean free paths to detector or DXTRAN sphere.
   real(dknd) :: ang(3)                  != Surface normal and cosine of track direction.
@@ -69,21 +93,24 @@
   real(dknd) :: ycn                     != Temperature-normalized neutron velocity.
   real(dknd) :: ztskcm                  != Marker after floating-point part of /TSKCM/.

-  common /tskcm/&
-    & amfp,  ang,   cbwf,  cmult, colout,ddet,  deb,   deltas,deltat,         &
-    & dti,   eg0,   ergace,pfp,   ple,   pmf,   psc,   ptbtc, qpl,   rnb,     &
-    & rr0,   sff,   siga,  ssr,   stp,   totgp1,totm,  totpn, tpd,   tpp,     &
-    & ttn,   udt,   udtsav,uold,  vtr,   wcs1tc,wcs2tc,ycn,   ztskcm
+  !--------------------------------------------------------------------------------------
--
+  !  (4) declarations for /tskcm/ INTEGER*8
+
+  ! Swept integer*8 - order is important, see varcom - length = l8varsw
+  integer(i8knd) :: nrnhtc(3)    != Task copy of NRNH.

+  ! Non-swept Integer*8
+  integer(i8knd) :: npsrtc       != Task copy of nspr
+  integer(i8knd) :: npstc        != Task copy of nps.
+
+  !--------------------------------------------------------------------------------------
--
+  !  (5) declarations for /tskcm/ INTEGERS
+  ! Swept tskcom integers - order is important, see varcom.
   ! Swept tskcom integers - order is important, see varcom.
   integer :: nbhwtc              != Task copy of NBHWM.
   integer :: nbttc(mipt)         != Task copy of NBT.
-  integer :: nrnhtc(3)           != Task copy of NRNH.
   integer :: nziytc(8,mxdx,mipt) != Task copy of NZIY.

-  common /tskcm/&
-    & nbhwtc,nbttc, nrnhtc,nziytc
-  !
   ! Non-swept tskcom integers.
   integer :: iax                 != Flag for presence of AXS vector.
   integer :: ibc                 != Index of the tally cosine bin.
@@ -131,34 +158,47 @@
   integer :: nlt                 != Number of entries in DTI.
   integer :: nmc                 != Counter for weight window generator tracking.
   integer :: npb                 != Number of saved particles in GPB9CM.
-  integer :: npsrtc              != Task copy of nspr
-  integer :: npstc               != Task copy of nps.
   integer :: nter                != Type of termination of the track.
   integer :: ntii                != Indicator of multiple time interrupts.
   integer :: ntx                 != Number of calls of TALLYX in user bins loop.
   integer :: ntyn                != Type of reaction in current collision.
   integer :: mtskcm              != Marker after integer part of /TSKCM/.

-  common /tskcm/&
+
+  !--------------------------------------------------------------------------------------
--
+  !  (6) declaration for /tskcm/   (real, interger*8, integer)
+
+  ! REAL
+  common /tskcm/ &   ! swept - order important
+    & colltc, eacctc, paxtc, rlttc, rnrtc, smultc, sumktc, tmavtc, wgtstc
+  common /tskcm/&    ! non-swept
+    & amfp,  ang,   cbwf,  cmult, colout,ddet,  deb,   deltas,deltat,         &
+    & dti,   eg0,   ergace,pfp,   ple,   pmf,   psc,   ptbtc, qpl,   rnb,     &
+    & rr0,   sff,   siga,  ssr,   stp,   totgp1,totm,  totpn, tpd,   tpp,     &
+    & ttn,   udt,   udtsav,uold,  vtr,   wcs1tc,wcs2tc,ycn,   ztskcm
```

```
+
+  ! INTEGER*8
+  common /tskcm/ &    ! swept
+    & nrnhtc
+  common /tskcm/ &    ! non-swept
+    & npsrtc,npstc
+
+  ! INTEGER
+  common /tskcm/&     ! swept
+    & nbhwtc,nbttc, nziytc
+  common /tskcm/&     ! non-swept
     & iax,   ibc,   ibe,   ibs,   ibt,   ibu,   ic0,   iclp,  id0,   idet,   &
     & iet,   imd,   iper,  ipsc,  irt,   isic,  ital,  iti,   iw0,   ixcos,  &
     & ixre,  jap,   jbd,   jbnk,  jev,   jlock, jtls,  kdb,   kqss,  ktask,  &
     & levp,  lgc,   lsb,   mbb,   mkc,   mpan,  nbnk,  nch,   ngp,   nlaj,   &
-    & nlse,  nlt,   nmc,   npb,   npsrtc,npstc, nter,  ntii,  ntx,   ntyn,   &
-    & mtskcm
-
-  ! Swept-tskcom,itskpt lengths and equivalences.
-  integer,parameter :: ntskcm = & != Size of floating-point part of /TSKCM/.
-    & nvarsw + 25*(1) + 3*(3) + 1*(5) + 1*(7) + 1*(64)&
-    & + 1*(3*11) + 2*(mipt) + 1*(mlgc) + 1*(3*maxv) + 4*(10*mxlv+10)
-  integer,parameter :: ltskcm = & != Size of integer part of /TSKCM/.
-    & lvarsw + 45*(1) + 1*(5*(1+mxlv)) + 1*(maxv) &
-    & + 1*(mipt) + 1*(mlgc) + 1*(mlgc+1) + 1*(nlocks)
+    & nlse,  nlt,   nmc,   npb,   nter,  ntii,  ntx,   ntyn,  mtskcm

-  real(dknd) :: gtskcm(ntskcm) != Equivalence to real part of /TSKCM/.
-  integer    :: jtskcm(ltskcm) != Equivalence to integer part of /TSKCM/.
+  EQUIVALENCE     (colltc,  gtskcm)
+  EQUIVALENCE     (nrnhtc, i8tskcm)
+  EQUIVALENCE     (nbhwtc,  jtskcm)

-  equivalence (colltc,gtskcm), (nbhwtc,jtskcm)
+  !------------------------------------------------------------------------------------
--
+  !  (7) declaration for /itskpt/  - k-pointers for parallel calcs

   ! Itskpt common -- parallel offset variables.
   integer :: kbnk                != Task offset for IBNK array.
diff  -Naurd MCNP5/Source/src/ttbr.F90 MCNP5_new/Source/src/ttbr.F90
--- MCNP5/Source/src/ttbr.F90 2003-04-30 20:13:02.000000000 -0600
+++ MCNP5_new/Source/src/ttbr.F90    2004-07-22 15:14:43.000000000 -0600
@@ -7,7 +7,9 @@
   ! d = length of path segment.

   use mcnp_global
+  use dxtran_mod
   use mcnp_debug
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

@@ -64,11 +66,7 @@
   npa = 1
   ncp = 0
   jsu = 0
-  idx = 0
-  do i = 1,ndx(2)
-    if( (xxx-dxx(2,1,i))**2+(yyy-dxx(2,2,i))**2+&
-      & (zzz-dxx(2,3,i))**2<dxx(2,5,i) )  idx = i
-  enddo
+  idx = inside_dxtran_sphere()
   ipsc = 13
   deb  = sqrt(erg*(erg+2.*gpt(3)))/(erg+gpt(3))

diff  -Naurd MCNP5/Source/src/ttyint.F90 MCNP5_new/Source/src/ttyint.F90
--- MCNP5/Source/src/ttyint.F90      2003-04-30 20:13:02.000000000 -0600
+++ MCNP5_new/Source/src/ttyint.F90  2004-07-22 15:14:43.000000000 -0600
@@ -6,11 +6,6 @@
   ! Set up the tty interrupt.
   use mcnp_debug
-#ifdef PCDOS
-  use mcnp_global
-  implicit none
-  integer :: k
-#endif

   external pttyin

@@ -40,9 +35,5 @@

 #endif /*def.unix.or.dec*/

-#ifdef PCDOS
-  call break(lockl)
-#endif /*def.pcdos*/
-
   return
 end subroutine ttyint
diff  -Naurd MCNP5/Source/src/ufiles.F90 MCNP5_new/Source/src/ufiles.F90
--- MCNP5/Source/src/ufiles.F90      2003-04-30 20:13:04.000000000 -0600
+++ MCNP5_new/Source/src/ufiles.F90  2004-07-22 15:14:43.000000000 -0600
@@ -6,6 +6,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/unimaz.F90 MCNP5_new/Source/src/unimaz.F90
--- MCNP5/Source/src/unimaz.F90      2003-04-30 20:13:04.000000000 -0600
+++ MCNP5_new/Source/src/unimaz.F90  2004-07-22 15:14:43.000000000 -0600
@@ -9,6 +9,8 @@
   use mcnp_global
   use mcnp_debug
   use dynamic_arrays
+  use erprnt_mod
+
   implicit real(dknd) (a-h,o-z)


diff  -Naurd MCNP5/Source/src/utask.F90 MCNP5_new/Source/src/utask.F90
--- MCNP5/Source/src/utask.F90       2003-11-05 17:23:12.000000000 -0700
+++ MCNP5_new/Source/src/utask.F90   2004-07-22 15:14:43.000000000 -0600
@@ -14,14 +14,17 @@

   ! Initialize /tskcom/ and /pblcom/.
   gtskcm(1:ntskcm) = 0.
+  i8tskcm(1:l8tskcm) = 0_i8knd
   jtskcm(1:ltskcm) = 0
+
   gpblcm(1:npblcm+1) = 0.
   gpb9cm(1:mpb,1:npblcm+1) = 0.
   jpblcm(1:lpblcm+1) = 0
   jpb9cm(1:mpb,1:lpblcm+1) = 0
+
   wcs1tc(1:mipt) = wcs1(1:mipt)
   wcs2tc(1:mipt) = wcs2(1:mipt)
-  wgtstc(1) = huge
+  wgtstc(1) = huge_float

   if( mcnp_opt_parallel ) then
     if( ntasks>1 )  jlock = -1
@@ -81,7 +84,7 @@

     ! Clear and reset history score and tally blocks for this task.
     if( itask>=0 ) then
-      nhsd( knhs+1:knhs+nsp12, 1:ntal*(npert+1) ) = 0
+      nhsd( knhs+1:knhs+nsp12, 1:ntal*(npert+1) ) = 0_i8knd
```

```
       shsd( kshs+1:kshs+nspt,  1:ntal*(npert+1) ) = 0.
       ddm( kddm+1, 1:ntal*(npert+1) ) = ddm( 1, 1:ntal*(npert+1) )
       shsd(kshs+nspt-1, 1:ntal*(npert+1)) = shsd(nspt-1, 1:ntal*(npert+1))
diff  -Naurd MCNP5/Source/src/varcom.F90 MCNP5_new/Source/src/varcom.F90
--- MCNP5/Source/src/varcom.F90      2003-04-30 20:13:08.000000000 -0600
+++ MCNP5_new/Source/src/varcom.F90  2004-07-22 15:14:43.000000000 -0600
@@ -22,23 +22,41 @@
   ! varcom tag:
   character(len=64), parameter, private :: fx_tag = "varcom 11-01-2002"

-  ! Varcom length parameters:
-  integer, parameter :: nvarsw =& != Number of swept variable common
+  !-----------------------------------------------------------------------------------------
--
+  !  (1) parameters for the length of the 3 portions of /varcm/
+
+  integer, parameter :: nvarsw =&    != Number of swept REALS
     &   1*(1) + 1*(2) + 1*(3)    + 1*(4) + 1*(7) + 1*(4*2)&
     & + 1*(6*21*mipt) + 1*(mipt) + 1*(mipt*3)

-  integer, parameter :: nvarcm =& != Size of floating-point part of
+  integer, parameter :: nvarcm =&    != Size of REAL part of /varcm/
     & nvarsw + 12*(1) + 2*(3)  + 1*(4)   + 1*(10) &
     &        + 1*(30) + 1*(50) + 1*(2*3) + 2*(3*3) + 2*(mipt)

-  integer, parameter :: lvarsw =& != Number of swept variable common
+  integer, parameter :: l8varsw = &  != Number of swept INTEGER*8
+    &       1*(3)
+
+  integer, parameter :: l8varcm =&   != Size of INTEGER*8 part of /VARCM/.
+    & l8varsw + 13*(1) + 1*(20)
+
+  integer, parameter :: lvarsw =& != Number of swept INTEGERS
     & 1*(1) + 1*(8*mxdx*mipt) + 1*(mipt) + 1*(3)

   integer, parameter :: lvarcm =& != Size of integer part of /VARCM/.
-    & lvarsw + 47*(1) + 2*(2)  + 1*(3)    + 1*(10)&
-    &        + 1*(20) + 1*(50) + 1*(2*99) + 1*(100)
+    & lvarsw + 34*(1) + 2*(2)  + 1*(3)    + 1*(10)&
+    &        + 1*(50) + 1*(2*99) + 1*(100)

-  ! Real variables swept from tskcom:
+  !-----------------------------------------------------------------------------------------
--
+  !  (2) declarations for 3 arrays equivalenced to /varcm/
+
+  real(dknd)     ::   gvarcm(nvarcm)
+  integer(i8knd) ::  i8varcm(l8varcm)
+  integer        ::   jvarcm(lvarcm)
+
+  !-----------------------------------------------------------------------------------------
--
+  !  (3) declarations for /varcm/ REALS
+
+  ! Real variables swept from tskcom - length = nvarsw
   real(dknd)    :: coll(mipt)           != Number of collisions in problem.
   real(dknd)    :: eacc(4)              != Weight and energy of electrons above EMAX.
   real(dknd)    :: pax(6,21,mipt)       != Ledger of creation and loss. See App. E.
@@ -75,10 +93,35 @@
   real(dknd)    :: wt0                  != Weight of each KCODE source point.
   real(dknd)    :: zvarcm               != Marker after floating-point part of /VARCM/.

+  !-----------------------------------------------------------------------------------------
--
+  !  (4) declarations for /varcm/ INTEGER*8
+
+  ! swept Integer*8 variables
+  integer(i8knd) :: nrnh(3)      != Information about number of random numbers used.
+
+  ! non-swept INTEGER*8
+  integer(i8knd) :: nps          != Count of source particles started.
+  integer(i8knd) :: npsmg        != Number of source particles that contribute
+                                 !=  to image grid (2nd nps card entry).
+  integer(i8knd) :: npsout       != NPS when output was last done.
+  integer(i8knd) :: npc(20)      != NPS for tally fluctuation charts. See App. E.
+  integer(i8knd) :: npd          != NPS step in tally fluctuation chart.
+  integer(i8knd) :: npp          != Number of histories to run, from NPS card.
+  integer(i8knd) :: npsr         != History number last read from surface source.
+  integer(i8knd) :: nqss         != Number of histories read from surface source.
+  integer(i8knd) :: nqsw         != Number of histories written to surface source.
+  integer(i8knd) :: nrrs         != Number of tracks read from surface source.
+  integer(i8knd) :: nrsw         != Number of tracks written to surface source.
+  integer(i8knd) :: ntc          != Control variable for time check.
+  integer(i8knd) :: ntc1         != Second control variable for time check.
+  integer(i8knd) :: ntss         != Number of surface source tracks accepted.
+
+  !------------------------------------------------------------------------------
--
+  !  (5) declarations for /varcm/ INTEGERS
+
   ! Integer variables swept from tskcom:
   integer :: nbhwm               != Largest number of particles ever in the bank.
   integer :: nbt(mipt)           != Total numbers particles banked.
-  integer :: nrnh(3)             != Information about number of random numbers used.
   integer :: nziy(8,mxdx,mipt)   != DXTRANs lost to zero importance.

   ! Non-swept integer variables:
@@ -111,21 +154,9 @@
   integer :: notrn               != Flag to only calculate direct source
                                  !=  to point detector tallies
                                  !=  (no particles are transported).
-  integer :: npc(20)             != NPS for tally fluctuation charts. See App. E.
-  integer :: npd                 != NPS step in tally fluctuation chart.
   integer :: npnm                != Count of times neutron-reaction MT not found.
-  integer :: npp                 != Number of histories to run, from NPS card.
   integer :: nppm                != Count of times photon-production MT not found.
-  integer :: nps                 != Count of source particles started.
-  integer :: npsmg               != Number of source particles that contribute
-                                 !=  to image grid (2nd nps card entry).
-  integer :: npsout              != NPS when output was last done.
-  integer :: npsr                != History number last read from surface source.
   integer :: npum                != Flag for Photonuclear production failure.
-  integer :: nqss                != Number of histories read from surface source.
-  integer :: nqsw                != Number of histories written to surface source.
-  integer :: nrrs                != Number of tracks read from surface source.
-  integer :: nrsw                != Number of tracks written to surface source.
   integer :: nsa                 != Source particles yet to be done in this cycle.
   integer :: nsa0                != Saved NSA value to rerun lost history.
   integer :: nskk                != Number of histories in first IKZ KCODE cycles.
@@ -135,12 +166,9 @@
   integer :: nss                 != Count of source points stored for the next cycle.
   integer :: nss0                != Saved NSS value to rerun lost history.
   integer :: nssi(10)            != Numbers of rejected surface source tracks.
-  integer :: ntc                 != Control variable for time check.
-  integer :: ntc1                != Second control variable for time check.
   integer :: ntprt(100)          != List of tally numbers on talnp card that
                                  !=  will not have bin values printed
                                  !=  to the outp file.
-  integer :: ntss                != Number of surface source tracks accepted.
   integer :: nwer                != Count of warning messages printed.
   integer :: nwsb                != Count of source weights below cutoff.
   integer :: nwse                != Count of source energies below cutoff.
@@ -149,10 +177,9 @@
   integer :: nwws(2,99)          != Like NWSG and NWSL but binned.
   integer :: mvarcm              != Marker variable at end of /VARCM/.

-  real(dknd)     ::       gvarcm(nvarcm)
-  EQUIVALENCE       (coll,gvarcm)
-  integer ::        jvarcm(lvarcm)
-  EQUIVALENCE       (nbhwm,jvarcm)
+
+  !------------------------------------------------------------------------------------
--
+  !  (6) declaration for /varcm/   (real, interger*8, integer)

   ! real variables swept from tskcom - order important, see tskcom.
   common /varcm/ &
@@ -164,151 +191,30 @@
     & rrmc,  rkk,   rssp,  rsum,  rsum2, snit,  swtm,  twac,  twss,  &
     & wcs1,  wcs2,  wssi,  wt0,   zvarcm

+ ! INTEGER*8
+  common /varcm/ &        ! swept - order important, see tskcom.
+    & nrnh
+  common /varcm/ &        ! non-swept
+     & nps,   npsmg, npsout,npc,   npd,   npp,   npsr,  nqss,  nqsw,  &
+     & nrrs,  nrsw,  ntc,   ntc1,  ntss
+
   ! int variables swept from tskcom - order important, see tskcom.
   common /varcm/ &
-    & nbhwm, nbt,   nrnh,  nziy
+    & nbhwm, nbt,  nziy

   ! non-swept integer variables.
   common /varcm/ &
     & irmc,  ist,   ist0,  ixak,  ixak0, jrad,  kc8,   kcz,   kcsf,  &
-    & kct,   kcy,   knod,  ksdef, lost,  monod, nbov,  ndmp,  &
-    & nerr,  nesm,  netb,  nfer,  notal, notrn, npc,   npd,   npnm,  &
-    & npp,   nppm,  nps,   npsmg, npsout,npsr,  npum,  nqss,  nqsw,  &
-    & nrrs,  nrsw,  nsa,   nsa0,  nskk,  nsom,  nss,   nss0,  nssi,  &
-    & ntc,   ntc1,  ntprt, ntss,  nwer,  nwsb,  nwse,  nwsg,  nwst,  &
-    & nwws,  mvarcm
-
-
-contains
-
-  !--------------------------------------------------------------------------
-  subroutine vr_write (iu,ierr)
-    ! description:
-    ! write varcom data to pre-positioned file.
-
-    ! argument declarations:
-    integer, intent(in)  :: iu      !  file unit number.
-    integer, intent(out) :: ierr    !  status.
-
-    ! local declarations:
-    logical           :: lopen
-    character(len=11) :: hformat
-
-    ! file must be opened and unformatted.
-    inquire(UNIT = iu, OPENED = lopen, FORM = hformat)
-
-    if( lopen .and. hformat == "UNFORMATTED") then
-      ierr = 0
-
-      ! write tag.
-      write(iu) fx_tag
-
-      ! write swept reals.
-      write(iu)&
-        & avarcm,coll,eacc,pax,rlt,rnr,smul ,sumk,tmav,wgts
-
-      ! write non-swept reals.
-      write(iu)&
-        & bcw,cpk,cts,dbcn,dmp,osum,osum2,pptme,prn,&
-        & rrmc,rkk,rssp,rsum,rsum2,snit,swtm,twac,twss,&
-        & wcs1,wcs2,wssi,wt0,zvarcm
-
-      ! write swept integers.
-      write(iu)&
-        & nbhwm,nbt,nrnh,nziy
-
```

```
-      ! write non-swept integers.
-      write(iu)&
-        & irmc,ist,ist0,ixak,ixak0,jrad,&
-        & kc8,kcz,kcsf,kct,kcy,knod,ksdef,lost,monod,&
-        & nbov,ndmp,nerr,nesm,netb,nfer,notal,notrn,npc,npd,&
-        & npnm,npp,nppm,nps,npsmg,npsout,npsr,npum,nqss,nqsw,nrrs,nrsw,&
-        & nsa,nsa0,nskk,nsom,nss,nss0,nssi,ntc,ntc1,ntprt,ntss,nwer,&
-        & nwsb,nwse,nwsg,nwst,nwws,mvarcm
-
-    else                               ! either not open or not unformatted.
-      ierr = -1                        ! return an error.
-
-    endif
-    return
-  end subroutine vr_write
-
-  !---------------------------------------------------------------------------
-
-  subroutine vr_read(iu,ierr)
-    ! description:
-    ! read varcom data from pre-positioned file. Position verified
-    ! by module tag.
-
-    ! argument declarations:
-    integer, intent(in)  :: iu      !  file unit number.
-    integer, intent(out) :: ierr    !  status.
-
-    ! local declarations:
-    logical           :: lopen
-    character(len=11) :: hformat
-    character(len=64) :: tag
-
-    ! file must be opened and unformatted.
-    inquire(UNIT = iu, OPENED = lopen, FORM = hformat)
-
-    if ( .not. lopen .or. hformat/="UNFORMATTED" )  then
-      ierr = -1 ! return error
-
-    else
-      ! read tag and verify it against module tag parameter.
-      read(iu) tag
-      if( tag/=fx_tag ) then
-        ierr = -2 ! return error
-
-      else
-        ierr = 0
-
-        ! read swept reals.
-        read(iu)&
-          & avarcm,coll,eacc,pax,rlt,rnr,smul ,sumk,tmav,wgts
-
-        ! read non-swept reals.
-        read(iu)&
-          & bcw,cpk,cts,dbcn,dmp,osum,osum2,pptme,prn,&
-          & rrmc,rkk,rssp,rsum,rsum2,snit,swtm,twac,twss,&
-          & wcs1,wcs2,wssi,wt0,zvarcm
-
-        ! read swept integers.
-        read(iu)&
-          & nbhwm,nbt,nrnh,nziy
-
-        ! read non-swept integers.
-        read(iu)&
-          & irmc,ist,ist0,ixak,ixak0,jrad, &
-          & kc8,kcz,kcsf,kct,kcy,knod,ksdef,lost,monod,&
-          & nbov,ndmp,nerr,nesm,netb,nfer,notal,notrn,npc,npd,&
-          & npnm,npp,nppm,nps,npsmg,npsout,npsr,npum,nqss,nqsw,nrrs,nrsw,&
-          & nsa,nsa0,nskk,nsom,nss,nss0,nssi,ntc,ntc1,ntprt,ntss,nwer,&
-          & nwsb,nwse,nwsg,nwst,nwws,mvarcm
-
-      endif
```

```
-
-    endif
-
-    return
-  end subroutine vr_read
-
-  !-----------------------------------------------------------------------------
-
-  subroutine vr_cast(mh,mx,ie)
-    ! Description:
-    ! DMMP bcast of varcom data.
-    ! Arguments:
-    integer,intent(in)    :: mh ! action flag, 0 -> sender.
-    integer,intent(in)    :: mx ! message chunk size (max).
-    integer,intent(inout) :: ie ! return status.
+    & kct,   kcy,   knod,  ksdef, lost,  monod, nbov,  ndmp,  nerr,  &
+    & nesm,  netb,  nfer,  notal, notrn, npnm,  nppm,  npum,  nsa,   &
+    & nsa0,  nskk,  nsom,  nss,   nss0,  nssi,  ntprt, nwer,  nwsb,  &
+    & nwse,  nwsg,  nwst,  nwws,  mvarcm

-    call dm_bcast(mh,gvarcm,nvarcm,mx,ie)
-    call dm_bcast(mh,jvarcm,lvarcm,mx,ie)
+  EQUIVALENCE       (coll,   gvarcm)
+  EQUIVALENCE       (nrnh,  i8varcm)
+  EQUIVALENCE       (nbhwm,  jvarcm)

-    return
-  end subroutine vr_cast
+  !--------------------------------------------------------------------------------------
--

 end module varcom
 !-
diff  -Naurd MCNP5/Source/src/viewz.F90 MCNP5_new/Source/src/viewz.F90
--- MCNP5/Source/src/viewz.F90       2003-04-30 20:13:08.000000000 -0600
+++ MCNP5_new/Source/src/viewz.F90   2004-07-22 15:14:43.000000000 -0600
@@ -148,8 +148,8 @@
     nn = nint(2.*a+.5)
     coe(1,1,jsu) = coe(1,1,jsu)-ii*d1
     coe(4,1,jsu) = coe(4,1,jsu)-ii*d2
-    udt(2*ku-1,lev) = huge
-    udt(2*ku,lev) = -huge
+    udt(2*ku-1,lev) = huge_float
+    udt(2*ku,lev) = -huge_float
     do j=1,nn
       coe(1,1,jsu) = coe(1,1,jsu)+d1
       coe(4,1,jsu) = coe(4,1,jsu)+d2
@@ -198,8 +198,8 @@
   a = (e1*d1+e2*d2)/(e1**2+e2**2)
   g1 = d1-e1*a
   g2 = d2-e2*a
-  u = huge
-  v = -huge
+  u = huge_float
+  v = -huge_float
   do i=-1,1,2
     do j=-1,1,2
       a = (i-st(1,1))*g1+(j-st(1,2))*g2
@@ -210,8 +210,8 @@
   a = 1./(g1**2+g2**2)
   mm = (v-u)*a+3.
   ii = -u*a+2.
-  u = huge
-  v = -huge
+  u = huge_float
+  v = -huge_float
   do i=-1,1,2
     do j=-1,1,2
       a = (i-st(1,1))*f1+(j-st(1,2))*f2
@@ -226,10 +226,10 @@
     coe(1,1,jn(i)) = coe(1,1,jn(i))-ii*d1-jj*e1
     coe(4,1,jn(i)) = coe(4,1,jn(i))-ii*d2-jj*e2
   end do
-  udt(1,lev) = huge
-  udt(2,lev) = -huge
-  udt(3,lev) = huge
-  udt(4,lev) = -huge
+  udt(1,lev) = huge_float
+  udt(2,lev) = -huge_float
+  udt(3,lev) = huge_float
+  udt(4,lev) = -huge_float
   lx = abs(lca(lc+1))-lca(lc)-6
   do j=1,nn
     do k=1,6
diff  -Naurd MCNP5/Source/src/voidcd.F90 MCNP5_new/Source/src/voidcd.F90
--- MCNP5/Source/src/voidcd.F90      2003-04-30 20:13:08.000000000 -0600
+++ MCNP5_new/Source/src/voidcd.F90  2004-07-22 15:14:43.000000000 -0600
@@ -7,6 +7,7 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/volume.F90 MCNP5_new/Source/src/volume.F90
--- MCNP5/Source/src/volume.F90      2003-11-05 17:23:12.000000000 -0700
+++ MCNP5_new/Source/src/volume.F90  2004-07-22 15:14:43.000000000 -0600
@@ -7,6 +7,8 @@
   use mcnp_global
   use mcnp_debug
   use mcnp_input
+  use qttyin_mod, only: qttyin
+  use erprnt_mod, only: erprnt

   implicit real(dknd) (a-h,o-z)
   character(len=23) :: hh
@@ -32,9 +34,6 @@
 50 format( "1",5x,  "details of volume and area calculations")
   DO_250: do icl_tmp=1,mxa
     icl = icl_tmp
-#ifdef PCDOS
-    if(lockl)call pttyin
-#endif /*def.pcdos*/
     if( irup/=0 )  call qttyin(ncl(icl),'  "calculating volume in cell",i5')
     if( dbcn(7)/=0. ) write(iuo,60)ncl(icl)
 60  format(2/," cell",i6,4x,104( "*"))
diff  -Naurd MCNP5/Source/src/vtask.F90 MCNP5_new/Source/src/vtask.F90
--- MCNP5/Source/src/vtask.F90       2003-11-05 17:23:12.000000000 -0700
+++ MCNP5_new/Source/src/vtask.F90   2004-07-22 15:14:43.000000000 -0600
@@ -10,6 +10,7 @@
   use mcnp_global
   use mcnp_debug
   use fmesh_mod, only:  nmesh, fmesh_vtask
+  use hpsort_mod, only: hpsort

   implicit real(dknd) (a-h,o-z)
   real(dknd) :: ar(ntp)
@@ -17,7 +18,7 @@
   ! Multiprocessing - use mynum and itask to determine action.

   ! define a multiplier which is  1 if mynum>0, 0 otherwise
-  lfix = 0
+  lfix=0
   if( mynum>0 ) then
     lfix = 1
   endif
@@ -31,7 +32,7 @@
     !$ call sm_loff(jlock,1)

     do it = 1,ntal*(npert+1)
-      if( nhsd(knhs+nsp+6,it)>1 ) then
```

```
+      if( nhsd(knhs+nsp+6,it)>1_i8knd ) then
        call hpsort( it, knhs+nsp+5, kshs+nsp+5,&
          &         nhsd(knhs+nsp+6,it), nhsd(knhs+nsp+7,it) )
      endif
@@ -72,19 +73,19 @@
    ! Combine global and task largest tally points into global array.
    if( ntal>0 ) then
      do it = 1,ntal*(npert+1)
-        if( nhsd(knhs+nsp+2,it)==0 )  cycle
+        if( nhsd(knhs+nsp+2,it)==0_i8knd )  cycle

        ln = 0
        ls = 0
        nhsd(ln+1:ln+nsp+5,it) = nhsd(ln+1:ln+nsp+5,it)+nhsd(knhs+1:knhs+nsp+5,it)
        shsd(ls+1:ls+nsp+5,it) = shsd(ls+1:ls+nsp+5,it)+shsd(kshs+1:kshs+nsp+5,it)

-        if( nhsd(knhs+nsp+6,it)==0 )  cycle
+        if( nhsd(knhs+nsp+6,it)==0_i8knd )  cycle

        ! Do extreme tally merge only if this task has large points.
        nk = nhsd(knhs+nsp+6,it)+kshs+nsp+5
        nl = nhsd(ln  +nsp+6,it)+ls  +nsp+5
-        np = min( ntp, nhsd(knhs+nsp+6,it)+nhsd(ln+nsp+6,it) )
+        np = min( int(ntp,i8knd), nhsd(knhs+nsp+6,it)+nhsd(ln+nsp+6,it) )

        ! Merge global and task extreme tallies and keep sorted.
        do k = np,1,-1
@@ -209,7 +210,7 @@

    l = 0
    isef(l   +1:l   +2,1:mxa) = isef(l+1:l+2,1:mxa)+isef(kise+1:kise+2,1:mxa)
-    isef(kise+1:kise+2,1:mxa) = 0
+    isef(kise+1:kise+2,1:mxa) = 0_i8knd

    l = 0
    n = 3*nmaz*(kpt(1)+kpt(2)+kpt(3))
diff  -Naurd MCNP5/Source/src/wgtul.F90 MCNP5_new/Source/src/wgtul.F90
--- MCNP5/Source/src/wgtul.F90       2003-04-30 20:13:10.000000000 -0600
+++ MCNP5_new/Source/src/wgtul.F90   2004-07-22 15:14:43.000000000 -0600
@@ -39,15 +39,15 @@
        a = 0.
        b = 0.
        if( sqq(2,n)<sqq(5,n) ) then
-          a = huge
-          b = huge
+          a = huge_float
+          b = huge_float
        endif
        if( sqq(6,n)/=0. )  a = abs(sqq(6,n))**(sqq(2,n)-sqq(5,n))
        if( sqq(7,n)/=0. )  b = abs(sqq(7,n))**(sqq(2,n)-sqq(5,n))
        wl = wl*sqq(3,n)*min(a,b)
        w  = max(a,b)
-        if( w==huge  ) then
-          wu = huge
+        if( w==huge_float  ) then
+          wu = huge_float
        else
          wu = wu*sqq(3,n)*w
        endif
@@ -94,11 +94,11 @@
  if( spf(ksd(13,n)+3,m)==0. ) go to 80
  if( ksd(19,n)*ksd(18,n)/=0 ) w=w*spf(ksd(13,n)+2,m)/spf(ksd(13,n)+3,m)
  w1 = w1*w
-  if( w2/=huge )  w2=w2*w
+  if( w2/=huge_float )  w2=w2*w
 70 continue
  wl = min(wl,w1)
-  if( w2==huge )  wu=huge
-  if( wu/=huge )  wu=max(wu,w2)
+  if( w2==huge_float )  wu=huge_float
+  if( wu/=huge_float )  wu=max(wu,w2)
```

```
 80 continue
   m = m+1
   if( m<=ksd(4,n) )  go to 40
diff  -Naurd MCNP5/Source/src/wrwssa.F90 MCNP5_new/Source/src/wrwssa.F90
--- MCNP5/Source/src/wrwssa.F90      2003-04-30 20:13:12.000000000 -0600
+++ MCNP5_new/Source/src/wrwssa.F90  2004-07-22 15:14:43.000000000 -0600
@@ -10,6 +10,7 @@
 
   real(dknd) :: rs(6), rg(11)
   character hq*80, hf*9
+  integer(i8knd) :: np
 
   ! create wssa and write general information and table lengths.
   call unique(wssa,jtty)
@@ -59,9 +60,9 @@
   endif
   rewind iusw
   write(jtty,70) wssa,nrsw
-70 format(  " surface-source file ",a8,  " written with",i8, " tracks.")
+70 format(  " surface-source file ",a8,  " written with",i12, " tracks.")
   write(iuo,80) wssa,np,nrsw
-80 format(  " surface-source file ",a8,  " with nps =",i7, " and",i8,&
+80 format(  " surface-source file ",a8,  " with nps =",i12, " and",i12,&
     &   " tracks was written for")
 
   ! write surfaces without facets.
diff  -Naurd MCNP5/Source/src/wtcalc.F90 MCNP5_new/Source/src/wtcalc.F90
--- MCNP5/Source/src/wtcalc.F90      2003-04-30 20:13:12.000000000 -0600
+++ MCNP5_new/Source/src/wtcalc.F90  2004-07-22 15:14:43.000000000 -0600
@@ -28,8 +28,8 @@
     ! Case of no biased dependent variables.
     call wgtul(wl,wu,ivdis(iv),jf)
     w1 = w1*wl
-    if( wu==huge )  w2=huge
-    if( w2/=huge )  w2=w2*wu
+    if( wu==huge_float )  w2=huge_float
+    if( w2/=huge_float )  w2=w2*wu
     cycle DO_140
 
     ! Case of biased dependent variables.
@@ -64,15 +64,15 @@
         if( ksd(6,jd)==0 )  cycle
         call wgtul(wl,wu,int(spf(ksd(13,jd)+1,i)),jf)
         wp = wp*wl
-        if( wu==huge )  wq=huge
-        if( wq/=huge )  wq=wq*wu
+        if( wu==huge_float )  wq=huge_float
+        if( wq/=huge_float )  wq=wq*wu
       end do
       wa = min(wa,wp)
       wb = max(wb,wq)
     end do
     w1 = w1*wa
-    if( wb==huge )  w2=huge
-    if( w2/=huge )  w2=w2*wb
+    if( wb==huge_float )  w2=huge_float
+    if( w2/=huge_float )  w2=w2*wb
     cycle DO_140
 
     ! Case of dependent variables with q option.
@@ -102,15 +102,15 @@
          if( wu==wl .or. ksd(8,jd)==0 )  cycle
          call wgtul(wl,wu,int(spf(ksd(13,jd)+1,i)),jf)
          wp = wp*wl
-         if( wu==huge )  wq=huge
-         if( wq/=huge )  wq=wq*wu
+         if( wu==huge_float )  wq=huge_float
+         if( wq/=huge_float )  wq=wq*wu
        end do
        wa = min(wa,wp)
        wb = max(wb,wq)
       end do
       w1 = w1*wa
-      if( wb==huge )  w2=huge
-      if( w2/=huge )  w2=w2*wb
+      if( wb==huge_float )  w2=huge_float
+      if( w2/=huge_float )  w2=w2*wb
       cycle DO_140
     end if
 130 continue
@@ -124,8 +124,8 @@
   if( swtm==-1. )  swtm=0.

   ! Print the minimum and maximum source weights.
-  w = huge
-  if( w2<.01*huge )  w=w2*srv(1,17)
+  w = huge_float
+  if( w2<.01*huge_float )  w=w2*srv(1,17)
   write(iuo,150) w1*srv(1,17),w
 150 format(/4x,  "minimum source weight =",1pe11.4,&
     & 4x,  "maximum source weight =",1pe11.4)
diff  -Naurd MCNP5/Source/src/wwfile.F90 MCNP5_new/Source/src/wwfile.F90
--- MCNP5/Source/src/wwfile.F90      2003-04-30 20:13:14.000000000 -0600
+++ MCNP5_new/Source/src/wwfile.F90  2004-07-22 15:14:43.000000000 -0600
@@ -6,6 +6,7 @@
   use mcnp_global
   use mcnp_debug
   use rmc_mod
+  use erprnt_mod

   implicit real(dknd) (a-h,o-z)

diff  -Naurd MCNP5/Source/src/xsgen.F90 MCNP5_new/Source/src/xsgen.F90
--- MCNP5/Source/src/xsgen.F90       2003-04-30 20:13:16.000000000 -0600
+++ MCNP5_new/Source/src/xsgen.F90   2004-07-22 15:14:43.000000000 -0600
@@ -7,6 +7,8 @@

   use mcnp_global
   use mcnp_debug
+  use qttyin_mod, only : qttyin
+  use erprnt_mod, only : erprnt

   implicit real(dknd) (a-h,o-z)
   real(dknd) ::  p(243,4)
@@ -208,14 +210,8 @@
     end do

     ! calculate range, straggling, and scattering tables.
-#ifdef PCDOS
-    if(lockl) call pttyin
-#endif
     if( irup/=0 )  call qttyin(mkc,'  "get electron range, mat",i5')
     call ronge(az,aa,.00001d0*q/aa,nd)
-#ifdef PCDOS
-    if(lockl)call pttyin
-#endif
     if( irup/=0 )  call qttyin(mkc,'  "get electron scat. xsec, mat",i5')
     call scatt(az)

@@ -256,14 +252,8 @@
     if( xnum>=0. )  xnm(mkc)=xnum

     ! calculate bremsstrahlung cross sections.
-#ifdef PCDOS
-    if(lockl)  call pttyin
-#endif
     if( irup/=0 )  call qttyin(mkc,'  "get bremsstrahlung xsec, mat",i5')
     call brem
-#ifdef PCDOS
-    if(lockl)  call pttyin
-#endif /*def.pcdos*/
     if( irup/=0 )  call qttyin(mkc,'  "get bremsstrahlung scat, mat",i5')
     call brang

diff  -Naurd MCNP5/Source/src/ypbssp.F90 MCNP5_new/Source/src/ypbssp.F90
--- MCNP5/Source/src/ypbssp.F90      2003-04-30 20:13:16.000000000 -0600
+++ MCNP5_new/Source/src/ypbssp.F90  2004-07-22 15:14:43.000000000 -0600
@@ -4,6 +4,7 @@
 subroutine ypbssp
   ! start a banked surface-source particle.
   use mcnp_global
+  use dxtran_mod
   use mcnp_debug

   implicit real(dknd) (a-h,o-z)
diff  -Naurd MCNP5/Testing/config/test_options.mk
MCNP5_new/Testing/config/test_options.mk
--- MCNP5/Testing/config/test_options.mk     2003-04-30 20:13:40.000000000 -0600
+++ MCNP5_new/Testing/config/test_options.mk 2004-07-22 15:06:14.000000000 -0600
@@ -169,6 +169,17 @@
   ifeq (OSF1,$(OS))
     PRUN    :=   prun -n  $(NMPI)  -c $(NTRD)
     PRUNSEQ :=   prun -n  1
+    # -----special syntax for lampi.  This procedure for mixed mode
+    # only works for 1 MPI task per note (=4 threads on Q machine)
+    ifneq (,$(findstring lampi, $(CONFIG)))
+      ifeq (1,$(NTRD))
+        PRUN := mpirun -n $(NMPI)
+      else
+        PRUN := mpirun -n $(NMPI) -N $(NMPI)
+      endif
+      PRUNSEQ := mpirun -n 1
+    endif
+
   else
     PRUN    := mpirun -np $(NMPI)
     PRUNSEQ := mpirun -np 1
```