

LA-UR-03-8102

Approved for public release;  
distribution is unlimited.

*Title:* Patch to update MCNP5 version MCNP5\_RSICC\_1.14 to  
version MCNP5\_RSICC\_1.20

*Author(s):* X-5 Monte Carlo Team  
Diagnostics Applications Group  
Los Alamos National Laboratory

October 23, 2003

*Submitted to:* Web page:  
<http://laws.lanl.gov/x5/MCNP/theresources.html>



Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the University of California for the U.S. Department of Energy under contract W-7405-ENG-36. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

## **Patch to update MCNP5 version MCNP5\_RSICC\_1.14 to version MCNP5\_RSICC\_1.20**

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

October 23, 2003

This patch (patch-MCNP5\_RSICC\_1.14\_to\_1.20) updates the Radiation Safety Information Computational Center (RSICC) release of MCNP5 from version MCNP5\_RSICC\_1.14 to version MCNP5\_RSICC\_1.20. The issues addressed by this patch file as well as instructions on applying this patch are given below. This patch can be downloaded from:

[http://laws.lanl.gov/x5/MCNP/mc/patch-MCNP5\\_RSICC\\_1.14\\_to\\_1.20](http://laws.lanl.gov/x5/MCNP/mc/patch-MCNP5_RSICC_1.14_to_1.20)

### **ISSUES**

Only one issue has been identified in MCNP5\_RSICC\_1.14 that will provide incorrect answers when running in sequential mode (see issue #1 below). Two other issues have been identified that will provide incorrect answers when using MPI/PVM and/or OpenMP (see issues #6 and #7 below). Other issues addressed by this patch are also listed below.

#### *Major issues:*

1. Fixed error in surface area calculation. Affects some surface area calculations and therefore affects surface tallies. (J. Bull)
2. Fixed access violation error when using FIR5 tally with a F4 tally. Problem reported by Jeffrey Gross. (J. Bull)

#### *Installation issues:*

3. Minor changes to Darwin, Linux, and OSF1 configuration files. (F. Brown, J. Sweezy, and S. Post)
4. Minor fix to install script to reset NMPI, NPVM, and NTRD to 1 when MPI, PVM, or OpenMP is not used. Causes failure during testing when the user changes from multi-processing to sequential, without resetting NMPI, NPVM, and NTRD manually. (J. Sweezy)
5. Minor fix for Lahey/Fujitsu Fortran 95 v6.2 for Linux and OpenMP. Causes compile failure. (J. Sweezy)

#### *OpenMP issues:*

6. Fix to problems with OpenMP that were causing multiple threads to have simultaneous read/write access to global variables. Causes possible wrong answers when using OpenMP. (J. Sweezy, J. Bull, and T. Goorley)

*MPI/PVM issues:*

7. Fix to explicitly open the scratch bank file. The scratch bank file was not being explicitly opened, causing a default scratch file "fort.60" to be created. For MPI/PVM processes sharing a common filesystem this caused all MPI/PVM processes to read and write the same "fort.60". Also thread task bank file unit numbers, 60+ktask, may interfere with the WWOUT and WWONE files. Moved WWOUT and WWONE to unit numbers less than 60. Causes possible wrong answers or fatal errors when a scratch bank file is used while running with MPI or PVM. (J. Bull)
8. Minor fix to support MPI/PVM when using the Intel compiler. Causes MCNP5 to crash. (J. Sweezy)

*Miscellaneous issues:*

9. Changed the format statements for writing of the number of particles used to normalize the tallies from F12.2 to F16.2. Affects output only. (J. Bull)
10. Increase mxdt (Maximum number of detectors) from 20 to 100. Increase ntlmx (Maximum number of tallies) from 100 to 1000. Increases the possible number of detectors and tallies. (F. Brown)
11. Fixed problem with allocation of memory when using mesh tallies in continue runs on the OSF1 platform. (J. Bull)
12. Fixed a bug in the plotter that resulted in the display of wrong cross section plots. (T. Goorley)

## **INSTALLATION INSTRUCTIONS**

For best results apply this patch to an unmodified copy of the June, 2003 RSICC release of MCNP5 (CCC-710) version MCNP5\_RSICC\_1.14. This patch may not work with a modified version of MCNP5\_RSICC\_1.14. 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 = 1.14
```

After verifying that you have an unmodified copy of MCNP5\_RSICC\_1.14 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 that the version may be different.

```
$ patch -v
patch 2.5.4
Copyright 1984-1988 Larry Wall
Copyright 1989-1999 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\_1.14\_to\_1.20” 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_1.14_to_1.20
```

- 5) Rebuild the MCNP5 executable.

```

# patch-MCNP5_RSICC_1.14_to_1.20 --- LA-UR-03-8102
#
# Patch to create MCNP5_RSICC_1.20 from MCNP5_RSICC_1.14
#
# USAGE
# -----
# To apply this patch to an unmodified copy of the June, 2003 RSICC release of
# MCNP5 (CCC-710), MCNP5_RSICC_1.14, 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 2.5.4
# Copyright 1984-1988 Larry Wall
# Copyright 1989-1999 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_1.14_to_1.20" 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_1.14_to_1.20
#
# 5) Recompile MCNP5.
#
# Note: This patch may fail if you have modified MCNP5.
# -----
#
Prereq: 1.14
diff -NaurwbDn MCNP5/Source/config/VC_info.gcf Modified_MCNP5/Source/config/VC_info.gcf
--- MCNP5/Source/config/VC_info.gcf      Fri Oct 17 15:51:43 2003
+++ Modified_MCNP5/Source/config/VC_info.gcf  Fri Oct 17 15:47:53 2003
@@ -2,4 +2,4 @@
# --- Thread Name
  THREAD = MCNP5_RSICC
# --- Thread Version Number
-THD_VERS = 1.14
+THD_VERS = 1.20
diff -NaurwbDn MCNP5/Source/config/Darwin.gcf Modified_MCNP5/Source/config/Darwin.gcf
--- MCNP5/Source/config/Darwin.gcf      Fri Oct 17 15:51:43 2003
+++ Modified_MCNP5/Source/config/Darwin.gcf  Fri Oct 17 15:47:53 2003
@@ -58,7 +58,7 @@
  else
    FDEBUG =
    CDEBUG =
-   FOPT  = -O
+   FOPT  = -O1 -Z1643 -Z1644
    COPT  =
  endif

diff -NaurwbDn MCNP5/Source/config/Linux.gcf Modified_MCNP5/Source/config/Linux.gcf
--- MCNP5/Source/config/Linux.gcf Fri Oct 17 15:51:43 2003

```

```

+++ Modified_MCNP5/Source/config/Linux.gcf      Fri Oct 17 15:47:53 2003
@@ -164,9 +164,22 @@

# --- Portland Workstation pgf90
ifeq (PORTLAND,$(findstring PORTLAND,$(FCOMPILER)))
+
+ # Syntax for -tp (target architecture)
+ # -tp {p5 | p6 | px | athlon}
+ #
+ # p5      - Pentium
+ # p6      - Pentium Pro/II/III or AMD Athlon systems
+ # px      - A blended p5/p6 style of code generation.
+ #          Executables will run on any x86 system
+ # athlon  - AMD Athlon-specific code generation
+
+ ifeq (debug,$(findstring debug,$(CONFIG)))
+     CDEBUG = -g
+     FDEBUG = -g
+     FOPT = -O0
+ else
+     FOPT = -O1 -tp px
+ endif

# --- Must do preprocessing
@@ -182,16 +195,6 @@
+     endif
+ endif

- # Syntax for -tp (target architecture)
- # -tp {p5 | p6 | px | athlon}
- #
- # p5      - Pentium
- # p6      - Pentium Pro/II/III or AMD Athlon systems
- # px      - A blended p5/p6 style of code generation.
- #          Executables will run on any x86 system
- # athlon  - AMD Athlon-specific code generation
-
- FOPT = -O2 -tp px
- FFLAGS = $(FDEBUG) $(FOPT) $(SMMP) -pc 64 $(I8R8)  $(MPIO)
- OBJF   = .o
- DEF_FCOMPILER= -DPGF90
diff -NaurwbDn MCNP5/Source/config/OSF1.gcf Modified_MCNP5/Source/config/OSF1.gcf
--- MCNP5/Source/config/OSF1.gcf Fri Oct 17 15:51:43 2003
+++ Modified_MCNP5/Source/config/OSF1.gcf      Fri Oct 17 15:47:53 2003
@@ -35,9 +35,9 @@
DOTCOMMROOT      = ../dotcomm
DOTCOMM_INTERNAL = mpi
DEF_DMMP         = -DMULTP -DMPI -DDMMP_NAME=$Q$(EXEC)$Q $ (MPICH)
- INC_DMMP_INTERNAL=
- INC_DMMP         = $(MOD_INC)$(DOTCOMMROOT)/src
- LIB_DMMP        = -L$(DOTCOMMROOT)/src -ldotcomm -lmpi
+ INC_DMMP_INTERNAL= ${MPI_COMPILE_FLAGS}
+ INC_DMMP         = $(MOD_INC)$(DOTCOMMROOT)/src ${MPI_COMPILE_FLAGS}
+ LIB_DMMP        = -L$(DOTCOMMROOT)/src -ldotcomm ${MPI_LD_FLAGS} -lmpi
LIBDOTCOMM      = $(DOTCOMMROOT)/src/libdotcomm.a
MOD_DMMP        = $(MOD_INC)../././src
MPIO            =

diff -NaurwbDn MCNP5/Source/install Modified_MCNP5/Source/install
--- MCNP5/Source/install Fri Oct 17 15:51:43 2003
+++ Modified_MCNP5/Source/install Fri Oct 17 15:47:53 2003
@@ -1038,6 +1038,8 @@
+     }
+     else

```

```

        menudistmem=seq
+       NPVM=1
+       NMPI=1
        fi
        ;;

@@ -1045,6 +1047,7 @@
        if [ $menusharedmem = 'omp' ]
        then {
+       menusharedmem=none
+       NTRD=1
        }
        else
            menusharedmem=omp
diff -NaurwbBdN MCNP5/Source/src/bankit.F90 Modified_MCNP5/Source/src/bankit.F90
--- MCNP5/Source/src/bankit.F90    Fri Oct 17 15:51:43 2003
+++ Modified_MCNP5/Source/src/bankit.F90    Fri Oct 17 15:47:55 2003
@@ -49,7 +49,7 @@
        ! *** note: sm_lon removed

        ! Create the bank backup file if necessary.
-       if( lbb(ktask+1)/=0 ) then
+       if( lbb(ktask+1)==0 ) then
            lbb(ktask+1) = 1
            write(iuo,30) npstc
            if( ntasks<=1 .and. ltasks<=1 ) then
diff -NaurwbBdN MCNP5/Source/src/calcva.F90 Modified_MCNP5/Source/src/calcva.F90
--- MCNP5/Source/src/calcva.F90    Fri Oct 17 15:51:43 2003
+++ Modified_MCNP5/Source/src/calcva.F90    Fri Oct 17 15:47:53 2003
@@ -28,8 +28,7 @@
        call axis(ie)
        if( ie == 0 ) then
            call putnq(ie)
-       else
-       return
+       if (ie /= 0 ) return
        endif
    endif

diff -NaurwbBdN MCNP5/Source/src/fmesh_mod.F90 Modified_MCNP5/Source/src/fmesh_mod.F90
--- MCNP5/Source/src/fmesh_mod.F90    Fri Oct 17 15:51:43 2003
+++ Modified_MCNP5/Source/src/fmesh_mod.F90    Fri Oct 17 15:47:54 2003
@@ -189,43 +189,16 @@
        read(iu) nmesh
        if( nmesh==0 ) return

-       if( allocated (fmtal) ) then
-       do i = 1,nmesh
-       is_assoc = associated( fmtal(i)%tally)
-       if( is_assoc ) deallocate(fmtal(i)%tally)
-       enddo
-       deallocate(fmtal)
-       endif
-       if( allocated (fm) ) then
-       do i = 1,nmesh
-       is_assoc = associated(fm(i)%ireact)
-       if( is_assoc ) deallocate(fm(i)%ireact)
-       is_assoc = associated(fm(i)%xrbn)
-       if( is_assoc ) deallocate(fm(i)%xrbn)
-       is_assoc = associated(fm(i)%yzbin)
-       if( is_assoc ) deallocate(fm(i)%yzbin)
-       is_assoc = associated(fm(i)%ztbin)
-       if( is_assoc ) deallocate(fm(i)%ztbin)

```

```

-       is_assoc = associated(fm(i)%enbin)
-       if( is_assoc ) deallocate(fm(i)%enbin)
-       is_assoc = associated(fm(i)%de)
-       if( is_assoc ) deallocate(fm(i)%de)
-       is_assoc = associated(fm(i)%df)
-       if( is_assoc ) deallocate(fm(i)%df)
-       is_assoc = associated(fm(i)%fmarry)
-       if( is_assoc ) deallocate(fm(i)%fmarry)
-       is_assoc = associated(fm(i)%fmerr)
-       if( is_assoc ) deallocate(fm(i)%fmerr)
-       enddo
-       deallocate(fm)
-     endif
-
+       ! 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"')
+       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"')
+       endif

+       ! Next read in the scalar and non-allocatable arrays  of derived type fm
+       do i = 1,nmesh
@@ -238,9 +211,15 @@
+       ! 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)
+           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
+         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)
@@ -249,6 +228,7 @@
+           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
@@ -259,12 +239,15 @@
+         ! If there is a dose response function, allocate and read in its values to RUNTPE
+         do i = 1,nmesh
+           if( fm(i)%intrapol>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"')
+               read(iu) fm(i)%de,fm(i)%df
+             endif

```



```

+     endif
+     enddo

@@ -274,12 +257,15 @@
    iy = fm(i)%nyzb-1
    iz = fm(i)%nztb-1
    ie = fm(i)%nenb-1
+   is_assoc = associated(fm(i)%fmarray)
+   if( .not. is_assoc ) then
    allocate (fm(i)%fmarray(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)%fmarray(:,:,:,1), &
        &        fm(i)%fmerr( :,:,:,1)
@@ -298,13 +284,15 @@
    enddo
    i_size_bins=i_size_bins*0.2/ntasks + 1

-   if( allocated (i_bins) ) deallocate (i_bins)
-   if( allocated (num_bins)) deallocate (num_bins)
-
+   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"')
+   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"')
+   endif
    num_bins=0
    i_bins=0

@@ -1754,7 +1742,7 @@
    write(iumt,'(1x,a80)') aid

        write(iumt,'(1x,"Number of histories used for normalizing tallies = ",      &
-           & f12.2)') sp_norm
+           & f16.2)') sp_norm

    do j = 1,nmesh

@@ -2087,7 +2075,7 @@
    enddo
    close (iumt)

-   write (iuo,"(/,' Mesh tallies written to file ',a8,'.')" ) meshtal
+   write (iuo,'(/," Mesh tallies written to file ",a8,".')" ) meshtal
    return

    end subroutine fmesh_print
@@ -2378,32 +2366,21 @@
!-----

-   subroutine fmesh_vtask(ntasks)
+   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) :: ntasks
- integer :: i,j,k1,k2,k3,k4
+ integer, intent(in) :: ktask
+ integer :: i,kt

+ kt = ktask+2
do i = 1,nmesh
- do j = 2,ntasks+1
- do k4=1,fm(i)%nenb-1
- do k3=1,fm(i)%nztb-1
- do k2=1,fm(i)%nyzb-1
- do k1=1,fm(i)%nxrb-1
- fm(i)%fmarray(k1,k2,k3,k4,1) = fm(i)%fmarray(k1,k2,k3,k4,1) &
- & +fm(i)%fmarray(k1,k2,k3,k4,j)
- fm(i)%fmerr(k1,k2,k3,k4,1) = fm(i)%fmerr(k1,k2,k3,k4,1) &
- & +fm(i)%fmerr(k1,k2,k3,k4,j)
- enddo
- enddo
- enddo
- enddo
- fm(i)%fmarray(:, :, :, :, 2:ntasks+1) = 0
- fm(i)%fmerr(:, :, :, :, 2:ntasks+1) = 0
+ fm(i)%fmarray(:, :, :, :, 1) = fm(i)%fmarray(:, :, :, :, 1)+fm(i)%fmarray(:, :, :, :, kt)
+ fm(i)%fmerr(:, :, :, :, 1) = fm(i)%fmerr(:, :, :, :, 1)+fm(i)%fmerr(:, :, :, :, kt)
+ fm(i)%fmarray(:, :, :, :, kt) = 0
+ fm(i)%fmerr(:, :, :, :, kt) = 0
enddo

end subroutine fmesh_vtask
diff -NaurwbBdN MCNP5/Source/src/mcnp_input.F90 Modified_MCNP5/Source/src/mcnp_input.F90
--- MCNP5/Source/src/mcnp_input.F90 Fri Oct 17 15:51:43 2003
+++ Modified_MCNP5/Source/src/mcnp_input.F90 Fri Oct 17 15:47:55 2003
@@ -9,7 +9,7 @@

! Module Parameters:
integer,parameter :: nkcd = 101 ! = Number of different types of input cards.
- integer,parameter :: ntalmx = 100 ! = Maximum number of tallies.
+ integer,parameter :: ntalmx = 1000 ! = Maximum number of tallies.
integer,parameter :: mopts = 7 ! = Number of M card options (gas, estep, etc.).

! Module Reals:
diff -NaurwbBdN MCNP5/Source/src/mcnp_params.F90 Modified_MCNP5/Source/src/mcnp_params.F90
--- MCNP5/Source/src/mcnp_params.F90 Fri Oct 17 15:51:43 2003
+++ Modified_MCNP5/Source/src/mcnp_params.F90 Fri Oct 17 15:47:55 2003
@@ -80,7 +80,7 @@
integer,parameter :: mstp = 4 ! = Coarsening factor for electron energy grids.
integer,parameter :: mtop = 89 ! = Number of bremsstrahlung energy groups + 1.
integer,parameter :: mwng = (mtop+1)/2 ! = Number of photon energy groups in ECH.
- integer,parameter :: mxdt = 20 ! = Maximum number of detectors.
+ integer,parameter :: mxdt = 100 ! = Maximum number of detectors.
integer,parameter :: mxdx = 10 ! = Maximum number of DXTRAN spheres.
integer,parameter :: mxlv = 10 ! = Maximum number of levels allowed for.
integer,parameter :: mxss = 6 ! = Spare dimension of surface source arrays.
@@ -105,11 +105,8 @@
integer,parameter :: iur = 33 ! = I/O unit for file of restart dumps.
integer,parameter :: iux = 34 ! = I/O unit for files of cross section tables.
integer,parameter :: iud = 35 ! = I/O unit for directory of cross section tables.

```

```

- integer,parameter :: iub = 60      != I/O unit for bank backup file.
integer,parameter :: iup = 37      != I/O unit for intermediate file of plots.
integer,parameter :: ius = 38      != I/O unit for KCODE source file.
- integer,parameter :: iuwe = 81     != I/O unit for output WWOUT file.
integer,parameter :: iuw1 = 82     != I/O unit for output WWONE file.
integer,parameter :: iu1 = 39      != I/O unit for a scratch file.
integer,parameter :: iu2 = 40      != I/O unit for another scratch file.
integer,parameter :: iusw = 41     != I/O unit for surface source output file.
@@ -126,6 +123,9 @@
integer,parameter :: iupx = 52     != Unit of file for writing plot print points.
integer,parameter :: iuw = 53      != I/O unit for input WWINP file.
integer,parameter :: iumt = 54     != I/O unit for the mesh tally output file
+ integer,parameter :: iuwe = 55   != I/O unit for output WWOUT file.
+ integer,parameter :: iuw1 = 56   != I/O unit for output WWONE file.
+ integer,parameter :: iub = 60    != I/O unit for bank backup file.

! General real constants:
real(dknd),parameter :: &
diff -NaurwbBdN MCNP5/Source/src/mcnp_random.F90 Modified_MCNP5/Source/src/mcnp_random.F90
--- MCNP5/Source/src/mcnp_random.F90      Fri Oct 17 15:51:43 2003
+++ Modified_MCNP5/Source/src/mcnp_random.F90  Fri Oct 17 15:47:54 2003
@@ -98,8 +98,8 @@
& RN_NPS                                ! current particle number

common /RN_THREAD/ RN_SEED, RN_COUNT, RN_NPS
- !$OMP THREADprivate ( /RN_THREAD/ )
save /RN_THREAD/
+ !$OMP THREADprivate ( /RN_THREAD/ )

!-----
! Shared data, to collect info on RN usage
diff -NaurwbBdN MCNP5/Source/src/mcplot_module.F90 Modified_MCNP5/Source/src/mcplot_module.F90
--- MCNP5/Source/src/mcplot_module.F90    Fri Oct 17 15:51:43 2003
+++ Modified_MCNP5/Source/src/mcplot_module.F90  Fri Oct 17 15:47:55 2003
@@ -2483,7 +2483,7 @@
ord(i) = 0.
do j = 1,kxsplt
y = 0.
- if( kxspen(j)>0 .and. kxspxs(j)>0 ) then
+ if( kxspen(j)>=0 .and. kxspxs(j)>0 ) then
kxsptp = nty(kxspie(j))
do
kj = kxspnx(mxel+j)
diff -NaurwbBdN MCNP5/Source/src/messages.F90 Modified_MCNP5/Source/src/messages.F90
--- MCNP5/Source/src/messages.F90 Fri Oct 17 15:51:43 2003
+++ Modified_MCNP5/Source/src/messages.F90  Fri Oct 17 15:47:54 2003
@@ -97,7 +97,7 @@
real(dknd), intent(inout) :: ptr(:, :)
integer, intent(in) :: nstart,n
integer :: rc
-#if defined(AIX) || defined(LAHEYLF95) || defined(ABSOFIT) || defined(PGF90)
+#if defined(AIX) || defined(LAHEYLF95) || defined(ABSOFIT) || defined(PGF90) || defined
(INTEL)
real(dknd), dimension(size(ptr)) :: tmp
if( n<size(ptr) ) then !preserve remainder of array
tmp = reshape(ptr,shape(tmp))
@@ -120,7 +120,7 @@
real(dknd), intent(inout) :: ptr(:, :, :)
integer, intent(in) :: nstart,n
integer :: rc
-#if defined(AIX) || defined(LAHEYLF95) || defined(ABSOFIT) || defined(PGF90)

```

```

+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90) || defined
(INTEL)
    real(dknd), dimension(size(ptr)) :: tmp
    if( n<size(ptr) ) then          !preserve remainder of array
        tmp = reshape(ptr,shape(tmp))
@@ -143,7 +143,7 @@
    real(dknd), intent(inout)      :: ptr(:, :, :, :)
    integer,    intent(in)         :: nstart,n
    integer                                          :: rc
-#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90)
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90) || defined
(INTEL)
    real(dknd), dimension(size(ptr)) :: tmp
    if( n<size(ptr) ) then          !preserve remainder of array
        tmp = reshape(ptr,shape(tmp))
@@ -166,7 +166,7 @@
    real(dknd), intent(inout)      :: ptr(:, :, :, :, :)
    integer,    intent(in)         :: nstart,n
    integer                                          :: rc
-#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90)
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90) || defined
(INTEL)
    real(dknd), dimension(size(ptr)) :: tmp
    if( n<size(ptr) ) then          !preserve remainder of array
        tmp = reshape(ptr,shape(tmp))
@@ -214,7 +214,7 @@
    integer,    intent(inout)      :: ptr(:, :)
    integer,    intent(in)         :: nstart,n
    integer                                          :: rc
-#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90)
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90) || defined
(INTEL)
    integer, dimension(size(ptr)) :: tmp
    if( n<size(ptr) ) then          !preserve remainder of array
        tmp = reshape(ptr,shape(tmp))
@@ -237,7 +237,7 @@
    integer,    intent(inout)      :: ptr(:, :, :)
    integer,    intent(in)         :: nstart,n
    integer                                          :: rc
-#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90)
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90) || defined
(INTEL)
    integer, dimension(size(ptr)) :: tmp
    if( n<size(ptr) ) then          !preserve remainder of array
        tmp = reshape(ptr,shape(tmp))
@@ -261,7 +261,7 @@
    integer,    intent(inout)      :: ptr(:, :, :, :)
    integer,    intent(in)         :: nstart,n
    integer                                          :: rc
-#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90)
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90) || defined
(INTEL)
    integer, dimension(size(ptr)) :: tmp
    if( n<size(ptr) ) then          !preserve remainder of array
        tmp = reshape(ptr,shape(tmp))
@@ -345,7 +345,7 @@
    real(dknd), intent(in)          :: ptr(:, :)
    integer,    intent(in)         :: nstart,n
    integer                                          :: rc
-#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90)
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90) || defined
(INTEL)
    real(dknd), dimension(size(ptr)) :: tmp

```

```

    tmp = reshape(ptr, shape(tmp))
    call dm_put(tmp(nstart:nstart+n-1), n, rc)
@@ -363,7 +363,7 @@
    real(dknd), intent(in)          :: ptr(:, :, :)
    integer,    intent(in)          :: nstart, n
    integer                                           :: rc
-#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90)
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90) || defined
(INTEL)
    real(dknd), dimension(size(ptr)) :: tmp
    tmp = reshape(ptr, shape(tmp))
    call dm_put(tmp(nstart:nstart+n-1), n, rc)
@@ -381,7 +381,7 @@
    real(dknd), intent(in)          :: ptr(:, :, :, :)
    integer,    intent(in)          :: nstart, n
    integer                                           :: rc
-#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90)
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90) || defined
(INTEL)
    real(dknd), dimension(size(ptr)) :: tmp
    tmp = reshape(ptr, shape(tmp))
    call dm_put(tmp(nstart:nstart+n-1), n, rc)
@@ -399,7 +399,7 @@
    real(dknd), intent(in)          :: ptr(:, :, :, :, :)
    integer,    intent(in)          :: nstart, n
    integer                                           :: rc
-#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90)
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90) || defined
(INTEL)
    real(dknd), dimension(size(ptr)) :: tmp
    tmp = reshape(ptr, shape(tmp))
    call dm_put(tmp(nstart:nstart+n-1), n, rc)
@@ -441,7 +441,7 @@
    integer,    intent(in)          :: ptr(:, :)
    integer,    intent(in)          :: nstart, n
    integer                                           :: rc
-#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90)
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90) || defined
(INTEL)
    integer, dimension(size(ptr)) :: tmp
    tmp = reshape(ptr, shape(tmp))
    call dm_put(tmp(nstart:nstart+n-1), n, rc)
@@ -459,7 +459,7 @@
    integer,    intent(in)          :: ptr(:, :, :)
    integer,    intent(in)          :: nstart, n
    integer                                           :: rc
-#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90)
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90) || defined
(INTEL)
    integer, dimension(size(ptr)) :: tmp
    tmp = reshape(ptr, shape(tmp))
    call dm_put(tmp(nstart:nstart+n-1), n, rc)
@@ -477,7 +477,7 @@
    integer,    intent(in)          :: ptr(:, :, :, :)
    integer,    intent(in)          :: nstart, n
    integer                                           :: rc
-#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90)
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90) || defined
(INTEL)
    integer, dimension(size(ptr)) :: tmp
    tmp = reshape(ptr, shape(tmp))
    call dm_put(tmp(nstart:nstart+n-1), n, rc)
@@ -547,7 +547,7 @@

```

```

        real(dknd), intent(inout)  :: ptr(:, :, :)
        integer,   intent(in)      :: n
        integer                                         :: rc
-#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90)
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90) || defined
(INTEL)
        real(dknd), dimension(size(ptr)) :: tmp
        if( n<size(ptr) ) then          !preserve remainder of array
            tmp = reshape(ptr, shape(tmp))
@@ -584,7 +584,7 @@
        integer, intent(inout)  :: ptr(:, :)
        integer,   intent(in)   :: n
        integer                                         :: rc
-#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90)
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90) || defined
(INTEL)
        integer, dimension(size(ptr)) :: tmp
        if( n<size(ptr) ) then          !preserve remainder of array
            tmp = reshape(ptr, shape(tmp))
@@ -620,7 +620,7 @@
        real(dknd), intent(in)      :: ptr(:, :, :)
        integer,   intent(in)       :: n
        integer                                         :: rc
-#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90)
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90) || defined
(INTEL)
        real(dknd), dimension(size(ptr)) :: tmp
        tmp = reshape(ptr, shape(tmp))
        call dm_put(tmp(:, ), n, rc)
@@ -651,7 +651,7 @@
        integer, intent(in)          :: ptr(:, :)
        integer,   intent(in)       :: n
        integer                                         :: rc
-#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90)
+#if defined(AIX) || defined(LAHEYLF95) || defined (ABSOFTE) || defined(PGF90) || defined
(INTEL)
        integer, dimension(size(ptr)) :: tmp
        tmp = reshape(ptr, shape(tmp))
        call dm_put(tmp(:, ), n, rc)
diff -NaurwBdN MCNP5/Source/src/msgtsk.F90 Modified MCNP5/Source/src/msgtsk.F90
--- MCNP5/Source/src/msgtsk.F90  Fri Oct 17 15:51:43 2003
+++ Modified MCNP5/Source/src/msgtsk.F90      Fri Oct 17 15:47:55 2003
@@ -328,7 +328,7 @@
        call dm_send(0,4,i)

        ! Send tallies in chunks.
- l1 = ktal + mxf + 1
+ l1 = mxf + 1
        l2 = l1 - 1 + (nmxf-1)*mxf
        do l = l1, l2, lchnk
            n = min( lchnk, l2-l+1 )
@@ -336,6 +336,7 @@
            call msg_put( tal,  l, n )
            call dm_send(0,4,i)
        enddo
+ tal(l1:l2)=zero

        ! Send task arrays & tally arrays - shsd, stt, nhsd.
        call dm_sndi
@@ -380,13 +381,16 @@
            call msg_put( nwws(:, i), 1, 2)      !one column is passed
        enddo
        do j = 1, ndnd

```

```

-   call msg_put( ddn(22,j)-ddn(kddn+22,j) )   !stride = 24
+   call msg_put( ddn(22,j)-ddn(22,ndnd+j) )   !stride = 24
enddo
+   ddn(21:24,ndnd+1:ndnd+ndnd)=zero
+
do k = 1,mxdx*min(1,nxn)
  do j = 1,mipt
-   call msg_put( dxd(j,22,k)-dxd(kxd+j,22,k) )   ! stride = 22
+   call msg_put( dxd(j,22,k)-dxd(j,22,mxdx+k) )   ! stride = 22
  enddo
enddo
+   dxd(1:mipt,21:24,mxdx+1:mxdx+mxdx)=zero

! Send variables depending on if the last microtask.
if( mynum/=nc(nm,2) ) then
diff -NaurwbDn MCNP5/Source/src/setdas.F90 Modified_MCNP5/Source/src/setdas.F90
--- MCNP5/Source/src/setdas.F90   Fri Oct 17 15:51:43 2003
+++ Modified_MCNP5/Source/src/setdas.F90   Fri Oct 17 15:47:55 2003
@@ -425,13 +425,13 @@
  yla = 0.0
  allocate( ddm( 1:2, 1:ntall*(npert1+1)*mt ) )
  ddm = 0.0
-  allocate( ddn( 1:24, 1:ndnd1*mt ) )
+  allocate( ddn( 1:24, 1:ndnd1*(mt+1) ) )
  ddn = 0.0
  allocate( dec( 1:3, 1:mxal*ndnd1*mt ) )
  dec = 0.0
  allocate( dxc( 1:3, 1:mxal*nxn1*mt ) )
  dxc = 0.0
-  allocate( dxd( 1:mipt, 1:24, 1:mxdx*mt ) )
+  allocate( dxd( 1:mipt, 1:24, 1:mxdx*(mt+1) ) )
  dxd = 0.0
  allocate( febl( 1:2, 1:max(16,igml)*mt ) )
  febl = 0.0
@@ -960,34 +960,34 @@
endif
  call msg_put( ddm, 1, 2*ntall*(npert+1))
  call msg_put( ddn, 1, 24*ndnd1)
-  call msg_put( dec, 1, 3*mxal*ndnd1)
-  call msg_put( dxc, 1, 3*mxal*nxn1)
+  ! call msg_put( dec, 1, 3*mxal*ndnd1)
+  ! call msg_put( dxc, 1, 3*mxal*nxn1)
  call msg_put( dxd, 1, mipt*24*mxdx)
-  call msg_put( febl, 1, 2*max(16,igml))
-  call msg_put( flx, 1, min(1,icwl)*mxal*igml)
+  ! call msg_put( febl, 1, 2*max(16,igml))
+  ! call msg_put( flx, 1, min(1,icwl)*mxal*igml)
  call msg_put( fso, 1, size(fso))
-  call msg_put( wwfa, 1, (mgww1+mipt)*nwwma1)
-  call msg_put( swwfa, 1, (mgww1+mipt)*nwwma1)
+  ! call msg_put( wwfa, 1, (mgww1+mipt)*nwwma1)
+  ! call msg_put( swwfa, 1, (mgww1+mipt)*nwwma1)

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

-  if( nsr==71 ) then

```

```

-   call msg_put( sump, 1, npert1)
-   endif
-   call msg_put( wns, 1, 2*(mxxs1/4))
+ ! if( nsr==71 ) then
+ !   call msg_put( sump, 1, npert1)
+ ! endif
+ ! call msg_put( wns, 1, 2*(mxxs1/4))
  call msg_put( fme, 1, size(fme))
  call msg_put( rho, 1, size(rho))
-   call msg_put( isef, 1, 2*mxal)
+ ! call msg_put( isef, 1, 2*mxal)
  call msg_put( jfq, 1, size(jfq))          ! jfq(1:,0) msg_put
  call msg_put( laj, 1, size(laj))
  call msg_put( lcaj, 1, size(lcaj))
  call msg_put( lse, 1, mxal)
-   call msg_put( maze, 1, 3*nmaz1*(kpt1(1)+kpt1(2)+kpt1(3)))
-   call msg_put( ndpf, 1, 6*ndnd1)
-   call msg_put( ndr, 1, mxel)
+ ! call msg_put( maze, 1, 3*nmaz1*(kpt1(1)+kpt1(2)+kpt1(3)))
+ ! call msg_put( ndpf, 1, 6*ndnd1)
+ ! call msg_put( ndr, 1, mxel)
  call msg_put( npsw, 1, size(npsw))
  call msg_put( nsl, 1, size(nsl))
  call msg_put( ntbb, 1, size(ntbb))
@@ -1150,33 +1150,33 @@
  endif
  call msg_get( ddm, 1, 2*ntal1*(npert+1))
  call msg_get( ddn, 1, 24*ndnd1)
-   call msg_get( dec, 1, 3*mxal*ndnd1)
-   call msg_get( dxc, 1, 3*mxal*nxxn1)
+ ! call msg_get( dec, 1, 3*mxal*ndnd1)
+ ! call msg_get( dxc, 1, 3*mxal*nxxn1)
  call msg_get( dxd, 1, mipt*24*mxdx)
-   call msg_get( febl, 1, 2*max(16,igml))
-   call msg_get( flx, 1, min(1,icwl)*mxal*igml)
+ ! call msg_get( febl, 1, 2*max(16,igml))
+ ! call msg_get( flx, 1, min(1,icwl)*mxal*igml)
  call msg_get( fso, 1, size(fso))
-   call msg_get( wwfa, 1, (mgww1+mipt)*nwwma1)
-   call msg_get( swwfa, 1, (mgww1+mipt)*nwwma1)
+ ! call msg_get( wwfa, 1, (mgww1+mipt)*nwwma1)
+ ! call msg_get( swwfa, 1, (mgww1+mipt)*nwwma1)
  call msg_get( pac, lpac+1, mipt*10*mxal)
  call msg_get( pan, lpan+1, 3*8*npn1)
-   call msg_get( pcc, 1, 3*mxal*kpt1(2))
+ ! call msg_get( pcc, 1, 3*mxal*kpt1(2))
  call msg_get( pwb, lpwb+1, mipt*22*mxal)

-   if( nsr==71 ) then
-     call msg_get( sump, 1, npert1)
-   endif
-   call msg_get( wns, 1, 2*(mxxs1/4))
+ ! if( nsr==71 ) then
+ !   call msg_get( sump, 1, npert1)
+ ! endif
+ ! call msg_get( wns, 1, 2*(mxxs1/4))
  call msg_get( fme, 1, size(fme))
  call msg_get( rho, 1, size(rho))
-   call msg_get( isef, 1, 2*mxal)
+ ! call msg_get( isef, 1, 2*mxal)
  call msg_get( jfq, 1, size(jfq))          ! jfq(1:,0) msg_get
  call msg_get( laj, 1, size(laj))
  call msg_get( lcaj, 1, size(lcaj))

```



```

    call msg_get( lse, 1, mxal)
- call msg_get( maze, 1, 3*nmaz1*(kpt1(1)+kpt1(2)+kpt1(3)))
- call msg_get( ndpf, 1, 6*ndnd1)
- call msg_get( ndr, 1, mxel)
+ ! call msg_get( maze, 1, 3*nmaz1*(kpt1(1)+kpt1(2)+kpt1(3)))
+ ! call msg_get( ndpf, 1, 6*ndnd1)
+ ! call msg_get( ndr, 1, mxel)
    call msg_get( npsw, 1, size(npsw))
    call msg_get( nsl, 1, size(nsl))
    call msg_get( ntbb, 1, size(ntbb))
@@ -1191,36 +1191,76 @@
    use mcnp_global
    use mcnp_debug
    implicit none
- integer :: i

- call msg_put( ddm, kddm+1, 2*ntall*(npert1+1))
- call msg_put( ddn, kddn+1, 24*ndnd1)
- call msg_put( dec, kdec+1, 3*mxal*ndnd1)
- call msg_put( dxc, kdx+1, 3*mxal*nxn1)
- call msg_put( dxd, kdxd+1, mipt*24*mxdx)
- call msg_put( febl, kfeb+1, 2*max(16, igm1))
- call msg_put( flx, kflx+1, min(1, icw1)*mxal*igm1)
- call msg_put( wwfa, kwfa+1, (mgww1+mipt)*nwmmal)
- call msg_put( swwfa, ksww+1, (mgww1+mipt)*nwmmal)
- call msg_put( pac, kpac+1, mipt*10*mxal)
- call msg_put( pan, kpan+1, 3*8*npn1)
- call msg_put( pcc, kpcc+1, 3*mxal*kpt1(2))
- call msg_put( pwb, kpwb+1, mipt*22*mxal)
+ call msg_put( ddm, 1, 2*ntall*(npert1+1))
+
+ call msg_put( ddn, 24*ndnd1+1, 24*ndnd1)
+ ddn(1:20, ndnd1+1:ndnd1+ndnd1)=zero
+
+ call msg_put( dec, 1, 3*mxal*ndnd1)
+ dec(1:3, 1:mxal*ndnd1)=zero
+
+ call msg_put( dxc, 1, 3*mxal*nxn1)
+ dxc(1:3, 1:mxal*nxn1)=zero
+
+ call msg_put( dxd, mipt*24*mxdx+1, mipt*24*mxdx)
+ dxd(1:mipt, 1:20, mxdx+1:mxdx+mxdx)=zero
+
+ call msg_put( febl, 1, 2*max(16, igm1))
+ febl(1:2, 1:max(16, igm1))=zero
+
+ call msg_put( flx, 1, min(1, icw1)*mxal*igm1)
+ flx(1:min(1, icw1)*mxal*igm1)=zero
+
+ call msg_put( wwfa, 1, (mgww1+mipt)*nwmmal)
+ wwfa(1:(mgww1+mipt)*nwmmal)=zero
+
+ call msg_put( swwfa, 1, (mgww1+mipt)*nwmmal)
+ swwfa(1:(mgww1+mipt)*nwmmal)=zero
+
+ call msg_put( pac, mipt*10*mxal+1, mipt*10*mxal)
+ pac(1:mipt, 1:10, mxal+1:mxal+mxal)=zero
+
+ call msg_put( pan, 3*8*npn1+1, 3*8*npn1)
+ pan(1:3, 1:8, npn1+1:npn1+npn1)=zero
+
+ call msg_put( pcc, 1, 3*mxal*kpt1(2))
+ pcc(1:3, 1:mxal*kpt1(2))=zero

```

```

+
+ call msg_put( pwb, mipt*22*mxal+1, mipt*22*mxal)
+ pwb(1:mipt,1:22,mxal+1:mxal+mxal)=zero
+
+   if( nsr==71 ) then
-     call msg_put( sump, ksum+1, npert1)
+     call msg_put( sump, 1, npert1)
+     sump(1:npert1)=zero
+   endif
-   call msg_put( wns, kwns+1, 2*(mxxs1/4))

-   call msg_put( isef, kise+1, 2*mxal)
-   call msg_put( maze, kmaz+1, 3*nmaz1*(kpt1(1)+kpt1(2)+kpt1(3)))
-   call msg_put( ndpf, kndp+1, 6*ndnd1)
-   call msg_put( ndr, kndr+1, mxel)
+   call msg_put( wns, 1, 2*(mxxs1/4))
+   wns(1:2,1:(mxxs1/4))=zero
+
+   call msg_put( isef, 1, 2*mxal)
+   isef(1:2,1:mxal)=0
+
+   call msg_put( maze, 1, 3*nmaz1*sum(kpt))
+   maze(1:3*nmaz1*sum(kpt))=0
+
+   call msg_put( ndpf, 1, 6*ndnd1)
+   ndpf(1:6,1:ndnd1)=0
+
+   call msg_put( ndr, 1, mxel)
+   ndr(1:mxel)=0

+   if( ntal>0 ) then
-     call msg_put( shsd, kshs+1, nspt*ntal*(npert+1))
-     call msg_put( stt, kstt+1, ntp*ntal*(npert+1))
-     call msg_put( nhsd, knhs+1, nsp12*ntal*(npert+1))
+     call msg_put( shsd, 1, nspt*ntal*(npert+1))
+     shsd(1:nspt,1:ntal*(npert+1))=zero
+
+     call msg_put( stt, 1, ntp*ntal*(npert+1))
+     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
+   endif
+
+   return
end subroutine task_arrays_msgput

diff -NaurwbBdN MCNP5/Source/src/tallyq.F90 Modified_MCNP5/Source/src/tallyq.F90
--- MCNP5/Source/src/tallyq.F90 Fri Oct 17 15:51:43 2003
+++ Modified_MCNP5/Source/src/tallyq.F90 Fri Oct 17 15:47:54 2003
@@ -21,7 +21,7 @@
! print the tally heading and modification notices.
call tallyh
if( nsr==71 .or. nsr==6 ) write(iuo,10) 1./fpi
-10 format( " number of histories used for normalizing tallies =",f12.2)
+10 format( " number of histories used for normalizing tallies =",f16.2)

! image detector grid prints.
jt = jptal(8,ital)
diff -NaurwbBdN MCNP5/Source/src/utask.F90 Modified_MCNP5/Source/src/utask.F90
--- MCNP5/Source/src/utask.F90 Fri Oct 17 15:51:43 2003
+++ Modified_MCNP5/Source/src/utask.F90 Fri Oct 17 15:47:55 2003
@@ -31,10 +31,10 @@

```

```

! Define the das offsets for this task.
kddm = (ktask+1)*2*ntal*(npert+1)
- kddn = (ktask+1)*24*ndnd
+ kddn = (ktask+2)*24*ndnd
kdec = (ktask+1)*3*mx*ndnd
kdx = (ktask+1)*3*mx*nxx
- kdx = (ktask+1)*mipt*24*mdx
+ kdx = (ktask+2)*mipt*24*mdx
kfeb = (ktask+1)*2*max(16,igm)
kflx = (ktask+1)*min(1,icw)*mx*igm
kwfa = (ktask+1)*(mgww(mipt+1)+mipt)*nwwma
diff -NaurwbBdN MCNP5/Source/src/volume.F90 Modified_MCNP5/Source/src/volume.F90
--- MCNP5/Source/src/volume.F90 Fri Oct 17 15:51:43 2003
+++ Modified_MCNP5/Source/src/volume.F90 Fri Oct 17 15:47:54 2003
@@ -80,6 +80,7 @@
do ital_tmp=1,ntal
ital = ital_tmp
if( lsat(ital)==0 .or. jptal(2,ital)<4 ) cycle
+ if ( jptal(8,ital) > 2 ) cycle
do il=1,iptal(1,3,ital)
ip = itds(iptal(1,1,ital)+il)
do i=1,itds(ip)
@@ -116,6 +117,7 @@
do ital_tmp=1,ntal
ital = ital_tmp
if( lsat(ital)==0 .or. jptal(2,ital)>=4 ) cycle
+ if ( jptal(8,ital) > 2 ) cycle
do il=1,iptal(1,3,ital)
ip = itds(iptal(1,1,ital)+il)
do i=1,itds(ip)
diff -NaurwbBdN MCNP5/Source/src/vtask.F90 Modified_MCNP5/Source/src/vtask.F90
--- MCNP5/Source/src/vtask.F90 Fri Oct 17 15:51:43 2003
+++ Modified_MCNP5/Source/src/vtask.F90 Fri Oct 17 15:47:55 2003
@@ -16,12 +16,17 @@

! Multiprocessing - use mynum and itask to determine action.

+ ! define a multiplier which is 1 if mynum>0, 0 otherwise
lfix = 0
+ if( mynum>0 ) then
+ lfix = 1
+ endif
+
if( mcnp_opt_multp ) then
if( mynum>0 .and. itask==-2 ) go to 109
endif

- if( mynum==0 .or. ntasks>1 ) then
+ ! if( mynum==0 .or. ntasks>1 ) then
! Sort task tallies with lock off for improved multitasking.
!$ call sm_loff(jlock,1)

@@ -33,13 +38,7 @@
enddo

!$ call sm_lon(jlock,1)
- endif
-
- ! to make it easy to choose either l-offsets or master-thread-k-offsets,
- ! define a multiplier which is 1 if mynum>0 && ntasks>1, 0 otherwise
- if( mynum>0 .and. ntasks>1 ) then
- lfix = 1

```

```

- endif
+ ! endif

! Sweep data from task common to variable common.
if( kc8/=2 ) then
@@ -67,17 +66,17 @@
sumk(1:3) = sumk(1:3) + sumktc(1:3)

if( mcnp_opt_parallel ) then
- ! Return if master thread of subtask.
- if( mynum>0 .and. ktask==0 ) return
+
+
! 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

- ln = lfix * nsp12*ntal*(npert+1)
- ls = lfix * nspt *ntal*(npert+1)
- nhsd(ln+1:ln+nsp+5,it) = nhsd(ln+1:ln+nsp+5,it)+nhsd(knhs+1:kshs+nsp+5,it)
+ 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
@@ -106,7 +105,7 @@
109 continue

if( mcnp_opt_parallel ) then
- l = lfix * 2*ntal*(npert+1)
+ l = 0
n = ntal*(npert+1)
where( ddm(kddm+1,1:n)>ddm(l+1,1:n) )
ddm(l+1,1:n) = ddm(kddm+1,1:n)
@@ -122,12 +121,12 @@
ddn(l+23,1:n) = ddn(kddn+23,1:n)
endwhere

- l = lfix * 3*mx*ndnd
+ l = 0
n = mx*ndnd
dec(l +1:1 +3,1:n) = dec(l+1:1+3,1:n)+dec(kdec+1:kdec+3,1:n)
dec(kdec+1:kdec+3,1:n) = 0.

- l = lfix * 3*mx*nxnx
+ l = 0
n = mx*nxnx
dxc(l +1:1 +3,1:n) = dxc(l+1:1+3,1:n)+dxc(kdxc+1:kdxc+3,1:n)
dxc(kdxc+1:kdxc+3,1:n) = 0.
@@ -144,27 +143,28 @@
enddo
enddo

- l = lfix * 2*max(16,igm)
+ l = 0
n = max(16,igm)
febl(l +1:1 +2,1:n) = febl(l+1:1+2,1:n)+febl(kfeb+1:kfeb+2,1:n)
febl(kfeb+1:kfeb+2,1:n) = 0.

- l = lfix * min(1,icw)*mx*igm

```

```

+   l = 0
+   n = min(1,icw)*mxa*igm
+   flx(l +1:l +n) = flx(l+1:l+n)+flx(kflx+1:kflx+n)
+   flx(kflx+1:kflx+n) = 0.

-   l = lfix * (mgww(mipt+1)+mipt)*nwwma
+   l = 0
+   n = (mgww(mipt+1)+mipt)*nwwma
+   wwfa( l +1:l +n) = wwfa(l+1:l+n)+ wwfa(kwfa+1:kwfa+n)
+   wwfa( kwfa+1:kwfa+n) = 0.
+   swwfa(l +1:l +n) = swwfa(l+1:l+n)+swwfa(ksww+1:ksww+n)
+   swwfa(ksww+1:ksww+n) = 0.

+
-   if( kc8/=2 ) then
+   l = lfix * 2*mipt*10*mx
+   l = lfix*mipt*10*mx
+   else
-   l = (lfix+1) * mipt*10*mx
+   l = mipt*10*mx
+   endif
+   do k = 1,mipt
+   pac(l +k,1:10,1:mx) = pac(l+k,1:10,1:mx)+pac(kpac+k,1:10,1:mx)
@@ -172,24 +172,24 @@
+   enddo

-   if( kc8/=2 ) then
+   l = lfix * 2*3*8*npn
+   l = lfix*3*8*npn
+   else
-   l = (lfix+1) * 3*8*npn
+   l = 3*8*npn
+   endif
+   do k = 1,3
+   pan(l +k,1:8,1:n) = pan(l+k,1:8,1:n)+pan(kpan+k,1:8,1:n)
+   pan(kpan+k,1:8,1:n) = 0.
+   enddo

-   l = lfix * 3*mx*kpt(2)
+   l = 0
+   n = mx*kpt(2)
+   pcc(l +1:l +3,1:n) = pcc(l+1:l+3,1:n)+pcc(kpcc+1:kpcc+3,1:n)
+   pcc(kpcc+1:kpcc+3,1:n) = 0.

-   if( kc8/=2 ) then
+   l = lfix * 2*mipt*22*mx
+   l = lfix*mipt*22*mx
+   else
-   l = (lfix+1) * mipt*22*mx
+   l = mipt*22*mx
+   endif
+   do k = 1,mipt
+   pwb(l +k,1:22,1:mx) = pwb(l+k,1:22,1:mx)+pwb(kpwb+k,1:22,1:mx)
@@ -197,34 +197,34 @@
+   enddo

-   if( nsr==71 ) then
+   l = lfix * 3*npert
+   l = 0
+   sump(l +1:l +npert) = sump(l+1:l+npert)+sump(ksum+1:ksum+npert)
+   sump(ksum+1:ksum+npert) = 0.
+   endif

```

```

-   l = lfix * (mxxs/2)
+   l = 0
    n = mxxs/4
    wns(l+1:l+2,1:n) = wns(l+1:l+2,1:n)+wns(kwns+1:kwns+2,1:n)
    wns(kwns+1:kwns+2,1:n) = 0.

-   l = lfix * 2*mxs
+   l = 0
    isef(l+1:l+2,1:mxs) = isef(l+1:l+2,1:mxs)+isef(kise+1:kise+2,1:mxs)
    isef(kise+1:kise+2,1:mxs) = 0

-   l = lfix * 3*nmaz*(kpt(1)+kpt(2)+kpt(3))
+   l = 0
    n = 3*nmaz*(kpt(1)+kpt(2)+kpt(3))
    maze(l+1:l+n) = maze(l+1:l+n)+maze(kmaz+1:kmaz+n)
    maze(kmaz+1:kmaz+n) = 0

-   l = lfix * 6*ndnd
+   l = 0
    ndpf(l+1:l+6,1:ndnd) = ndpf(l+1:l+6,1:ndnd)+ndpf(kndp+1:kndp+6,1:ndnd)
    ndpf(kndp+1:kndp+6,1:ndnd) = 0

-   l = lfix * mxel
+   l = 0
    ndr(l+1:l+mxe) = ndr(l+1:l+mxe)+ndr(kndr+1:kndr+mxe)
    ndr(kndr+1:kndr+mxe) = 0

-   l = lfix * (nmxf*mxs+ktls)
+   l = 0
    do i = 1, (nmxf-1)*mxs
        tal(l+mxs+i) = tal(l+mxs+i)+tal(ktal+mxs+i)
        tal(ktal+mxs+i) = 0.
@@ -234,7 +234,7 @@
! Merge the mesh tally values into the 1st array bin.
! This needs to be done even if run as a sequential code
if( nmesh>0 ) then
-   call fmesh_vtask(ntasks)
+   call fmesh_vtask(ktask)
endif

return

```