

LA-UR-11-01654

Approved for public release;
distribution is unlimited.

<i>Title:</i>	LNK3DNT Geometry Support: User Guidance for Creating and Embedding
<i>Author(s):</i>	Lawrence J. Cox, Ph.D.
<i>Intended for:</i>	Distribution to MCNP6 Users as part of the RSICC package



Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

LNK3DNT Geometry Support: User Guidance for Creating and Embedding

Lawrence J. Cox

ABSTRACT

The ability to create and import structured geometries has been added to MCNP6. This note provides instruction on creating and using LNK3DNT geometries as embedded objects. The user interface (input file formats) have been made common between structured and unstructured geometry types.

1 Overview

An *embedded geometry* [EG]¹ is a geometry description of a mass/material distribution consisting of a collection of *geometry elements* each of which may have zero or more materials occupying it. When used in MCNP, an EG resides in an MCNP universe. Typical MCNP universes are essentially infinite—albeit often cropped by the cell in which they are placed—and *embedded geometry universes* [EGU] also have this property. However, embedded geometries are generally finite in extent and a typical MCNP cell filled with an EGU may not be fully filled by the assigned EG alone. Therefore, the EGU also includes the (infinite) surroundings of the EG.

Another property of a typical MCNP universe is that it, in effect, does not know wherein it resides (*i.e.*, the structure of the levels above it) and often is used more than once. An EGU does, thus, also not know its holder(s).

An EGU is “reusable,” *i.e.*, can be placed in more than one location in the ways MCNP allows placement of other universes. When used more than once, an MCNP universe always has the same configuration with respect to geometry, materials and densities.

2 Using MCNP to Create a LNK3DNT-format Geometry

MCNP can create LNK3DNT-format files from a standard MCNP input deck. Two new cards are used for this: **MESH** and **DAWWG**.

The **MESH** card is used to specify the layout and orientation of the geometry to be generated with respect to the input deck coordinate system. See the *MCNP5 Manual Volume II*, §3.IV.C.12 for details on its parameters. The **XYZ** (3D Cartesian) and **CYL** (2D[RZ] and 3D[RZθ] cylindrical) and **SPH** (1D spherical) options are the only coordinate systems supported at this time.

¹ A table of *concepts, terms and descriptions* is given in Section 6.

The **DAWWG** card is used to specify the number of points to sample in each gElement while calculating the material presence and densities. The **points** entry specifies how many steps to randomly sample in each coordinate direction when estimating the composition of each gElement. A value of 10 means that $10^3=1000$ points will be sampled in each gElement. **NOTE:** In the current implementation, the **DAWWG** card must occur *after* the **MESH** card in the input file's data section. (*To be resolved*). Other PartiSn options may also be placed on the **DAWWG** card. See ??? for full specification of this card.

For example, **MESH** and **DAWWG** cards specifying a cylindrical geometry with diameter and length of 20cm having two radial, two axial and four azimuthal divisions (eight total elements) may look like this:

```
mesh geom cyl
  origin  0.0  0.0 -10.0  $ bottom center of cylinder
  imesh   10                    $ outside radius
  iints   2                      $ # radial divisions
  jmesh   20                    $ axial (z) length
  jints   2                      $ # axial divisions
  kmesh   1                      $ 1 rotation (0-2π)
  kints   4                      $ # azimuthal divisions {0, π/2, π, 3π/2, 2π}
dawwg points=10 [xsec=ndilib]$2
```

The optional **ORG**, **AXS** and **VEC** keywords of the **MESH** card can be used to arbitrarily align the EG mesh with the MCNP geometry from which it will be generated.

NOTE: *The original orientation is not stored in the EG output.* The generated EG is defined to be geometrically centered at **(0,0,0)** and aligned with the standard MCNP Cartesian coordinate system. For cylindrical geometries, the defaults are that the cylinder axis is aligned with the positive Z axis and the azimuthal plane ($\theta=0$) is aligned with the positive X axis.

To generate the mesh file, MCNP is executed with a new command-line option, **m**.

```
> mcnp6 m i=myinp linkout=mylnk [...]
```

When so invoked, MCNP exits after processing the input and generating the **LINKOUT** file, in this case named **mylnk**.

3 Embedding: Specifying an Embedded Geometry

One new card, **EMBED**, is required for embedding a **LNK3DNT** geometry into MCNP input. There are other cards related to using EGs, including **EMBEE**, **EMBEM**, **EMBEB**, **EMBTM** and **EMBTB**. However, these other cards are not currently used with embedded *structured* (**LNK3DNT**) geometries.

For each unique EG used in an MCNP input deck, there is an **EMBED** card with some required and some optional entries. The **EMBED** card and its keywords are listed below in a form suitable for inclusion into the MCNP6 manual.

EMBED: Embedded Geometry Specification

Form: **EMBEDn key=value(s) [key=value(s)] [...]**

n = EGU number (int)

² The **xsec** entry declares which cross section library would be passed to PARTISN for weight-window generation. It is not explicitly used in generation of the mesh. As currently coded, this keyword is required.

Required keys:

MESHGEO = EG format – ‘**LNK3DNT**’
MGEOIN = EG input filename
BACKGROUND = **c1** bCell number matching an iCell card (int)
MATCELL = **m1 c1 m2 c2 m3 c3...**
Integer pairs, one for each material in the EG
m<n> values are EG material numbers
c<n> values are the gCell numbers matching an iCell

Optional keys:

CALC_VOLS = ‘**yes**’, if present, calculate the gCell volumes and masses
DEBUG = ‘**echomesh**’, if present, write EG parameters to output

The **matcell** entry must have one **m/c** pair for each unique material in the EG. This also means there must be an iCell for each material present. A unique iCell be used for each unique material. If there are void elements in the EG, there must also be a **matcell** entry for material **0** and an associated iCell. The assigned **background** cell must be a unique iCell. A warning is issued if there are iCells that are not listed on the **background** or **matcell** entries. It is a fatal error if a material appears in the **LNK3DNT** file that is not mapped to an iCell.



NOTE: While iCells associated with non-void EG elements must have a specified density (gram or number), MCNP uses the element-specific densities stored in the **LNK3DNT** file for transport (*e.g.*, cross section lookup) and plotting. The density on the iCell card should be considered a reference density.

4 Embedding: Placing an Embedded Geometry into the MCNP Cell-based World

An EG is used in MCNP by defining an *embedded geometry universe* [EGU] that is tied to an **EMBED** card and placing that universe somewhere in the MCNP geometry hierarchy.

In MCNP, space is defined by a collection of cells. Cell geometries are typically defined by surfaces, lattices, universes, fills, etc. Fully functional use of an EG in MCNP, therefore, requires defining the cells that represent the geometry and its composition. This collection of cells comprises the EGU.

4.1 Cell Types Used with an Embedded Geometry

There are three special MCNP cell categories for specifying the use of an EG: *geometry cells*, *background cells* and *fill cells*. Geometry cells and background cells are instances of a special cell-type category: *inferred cells*.

An EGU consists of one or more *geometry cells* and a single *background cell*.

4.1.1 Inferred Cells

The cells used to define an EGU are called *inferred-geometry cells* or simply *inferred cells* [iCell]³ and are defined by a **NULL** surface. The cell-card format for iCells has the following properties that differ from those of typical MCNP cells:

- iCells have a single null-surface entry in the surface description block (*i.e.*, **0**);
- iCells are assigned to an EGU (*e.g.*, **u=e10** or **u=10**);

³ An *inferred cell* is what in the past has been called a *pseudo-cell*.

- The iCell's EGU number matches the **N** specified on the **embedN** card with which is associated;
- iCells cannot be filled by another universe or lattice (*i.e.*, no **fill=** or **lat=** entry).

As with other MCNP cells, iCells may have material, density, importance and other cell properties defined on their cell cards. All cell-card fields that are typically required are also required for iCells.

As mentioned above, there are two categories of (uses for) iCells:

Geometry Cells: For each material defined in an EG input file, an *inferred geometry cell* [gCell] must be defined. If there are void elements, an additional, unique gCell is required to represent them. Pure elements (one material or void) will belong to only one gCell. Multi-material elements will belong to each gCell that is associated with a material contained in the element. The gCells are connected to an EG through the **MATCELL** entries on the associated **EMBED** card. See Section 3 for a full description of the **EMBED** card.

Background cells: To make an EGU infinite in extent, it is required to have an *inferred background cell* [bCell] that consists of all space outside the associated EG. The background cell is connected to the EGU through the **BACKGROUND** entry on the **EMBED** card. Because of its infinite extent, a bCell may not be *specified-as-not-truncated*.⁴

4.1.2 Fill cells

Placement of an EGU within an MCNP input is done via a regular MCNP *fill cell*. MCNP fill cells may be filled with a lattice or a universe. It is now also possible to fill a cell with an EGU. The cell card for a cell filled with an EGU has the following properties that differ from those of other MCNP fill cells:

- An EGU-filled cell cannot be an inferred cell.
- They are filled with an EGU. That is, they have a fill entry of the form **fill=eN** or **fill=N**, where **N** is the MCNP universe number of the EGU (the **N** specified on the **embedN** card). The **fill** entry may have a transformation (**TR**) to permit realignment of the EG within the fill cell. Recall that **LNK3DNT** geometries are always defined relative to the MCNP origin.
- EGU filled cells cannot have a lattice entry.

5 Plotting with a **LNK3DNT** geometry

Color (shaded) plotting of the materials, density or number-density of a **LNK3DNT** mesh is supported in the MCNP plotter. For all these cases, the values from the **LNK3DNT** file are used, which (may) vary element to element.

For material plots, multi-material zones may appear striped as the color to plot is chosen randomly based on the material *mass fraction*. This means that redrawing a *color-by-mat* plot may give a slightly different striping. For example, in a two material element with a 50/50 mass mix, there will be approximately 50/50 striping (horizontally). If *mat* is selected from the right-side interactive menu, clicking on a spot containing multiple materials will randomly select which material to report. Repeated clicking on such a spot may show different materials

⁴ See Chapter 3 §IV.B.3.1 of the MCNP5 Manual, Volume II.

on different clicks. Void elements in the mesh are not shaded (i.e., shown as white) on material plots.

For density (*color-by-den*) and number density plots (*color-by-rho*), each element will be shown in one solid color. What is plotted for these cases is the *element-net* value, i.e., the net density or net number density of the element. The color distribution is set by the minima and maxima. It is shown on the color bar superimposed on such plots. These net values are also the values reported for plot queries (clicks on the plot) when *den* or *rho* is selected from the right-side interactive menu.

MCNP geometry plotting now also allows arbitrary zooming in color modes, with or without embedded geometries. In prior versions, if a zoom level would not plot any cell or surface *lines*, the plot would not be drawn or the plot would be reset to the default extent and origin. The user can now zoom arbitrarily if a color-fill mode is on. This is very useful for examining the details of an embedded **LNK3DNT** mesh.

6 **EMBED** Example

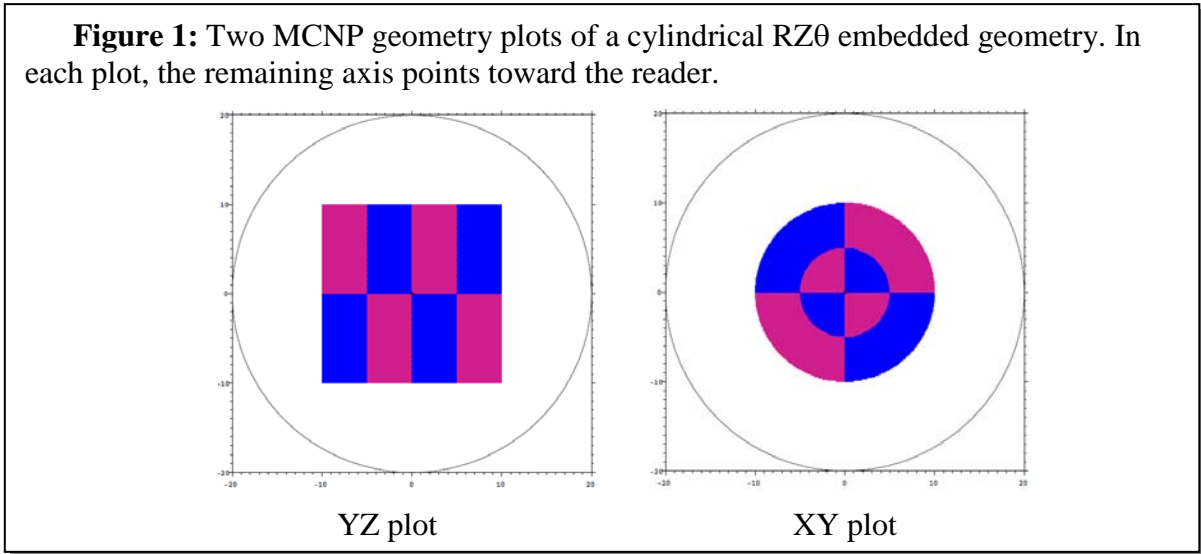
There are a variety of **EMBED** test problems in the **MCNP6/Testing/FEATURES/DAWWG/DAWWG** directory. The relevant parts of an MCNP input that uses the geometry file created by the **MESH** given in Section 2 (and named **cyl.linkout**) might look like this:

```
c Cell cards
11  3 -18.7      0    u=e10 imp:n=1 $ gCell
12  4 -0.001    0    u=e10 imp:n=1 $ gCell
13  0           0    u=e10 imp:n=1 $ bCell
20  0           -1 fill=e10 imp:n=1 $ fill cell
...

c Surface cards
1 ...

c Data Cards
...
embed10 meshgeo = lnk3dnt
      mgeoin   = cyl.linkout
      matcell  = 1 11 2 12
      background = 13
...
```

In this example gCell(11) maps to mesh-material(1), gCell(12) maps to mesh-material(2) and gCell(13) is the cell that fills out EGU-10. Note that the ‘e’ is optional for the **u** and **fill** keywords. **Figure 1** shows two views of the resulting geometry with the embedded geometry shaded by material, which in this case alternates between gElements.

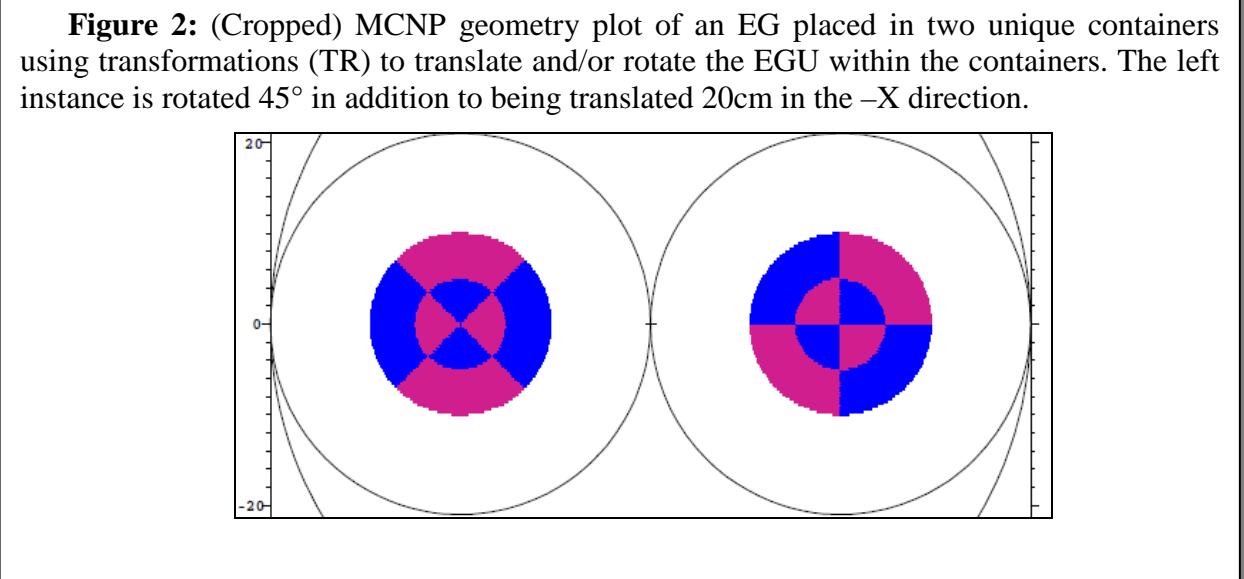


A single EG can be used more than once. One way is to place the EGU into more than one unique fill cell. Another is to place an EGU-filled cell into its own universe and use that universe to fill other MCNP cells. These two options make geometry embedding compatible with the MCNP geometry hierarchy options available when using universes.

As an example of the first option, adding another container filled with the same EGU with appropriate transformations gives an image like the one shown in **Figure 2**. For this example, surfaces 1 and 2 are off-center spheres and the transformations shift the EGU to be aligned with their geometric centers (± 20 cm).

```

c Cell cards
11  3  -18.7      0    u=e10 imp:n=1          $ gCell
12  4  -0.001     0    u=e10 imp:n=1          $ gCell
13  0              0    u=e10 imp:n=1          $ bCell
20  0  -1 fill=e10 ( 20 0 0 ) imp:n=1        $ fill cell 1
21  0  -2 fill=e10 (-20 0 0 -1 1 0 1 1 0) imp:n=1 $ fill cell 2
  
```



7 Concepts and Terms

The following table defines some of the concepts used in this memo and the abbreviated terms used for them.

Term	Concepts	Description
EG	embedded geometry	A finite geometrical mesh description having material and density information for each mesh element.
element	geometry element	A single element of the embedded geometry mesh.
iCell	inferred cell	An MCNP cell defined by a null surface. The geometrical layout of the cell is inferred from the object with which the cell is associated.
gCell	inferred geometry cell	An <i>iCell</i> mapped to one or more <i>gElements</i> of a specified <i>EG</i> . The mapping is defined by associating the <i>gCell</i> 's material with one of the <i>EG</i> materials. Note that a given <i>gElement</i> will belong to more than one <i>gCell</i> if it contains more than one material.
bCell	inferred background cell	An infinite (space filling) <i>iCell</i> that excludes the volume of an <i>EG</i> .
EGU	embedded geometry universe	A universe consisting of the <i>gCells</i> and one <i>bCell</i> associated with an <i>EG</i> , thereby filling all space.

8 New Card Specifications

8.1 EMBED Card

EMBED: Embedded Geometry Specification

Form: `EMBEDn key=value(s) [key=value(s)] [...]`

`n` = EGU number (int)

Required keys:

`MESHGEO` = EG format – '`LNK3DNT`'

`MGEOIN` = EG input filename

`BACKGROUND` = `c1` bCell number matching an iCell card (int)

`MATCELL` = `m1 c1 m2 c2 m3 c3...`

Integer pairs, one for each material in the EG

`m<n>` values are EG material numbers

`c<n>` values are the gCell numbers matching an iCell

Optional keys:

`CALC_VOLS` = '`yes`', if present, calculate the gCell volumes and masses

`DEBUG` = '`echomesh`', if present, write EG parameters to output

The `matcell` entry must have one `m/c` pair for each unique material in the EG. This also means there must be an iCell for each material present. A unique iCell be used for each unique material. If there are void elements in the EG, there must also be a `matcell` entry for material `0` and an associated iCell. The assigned `background` cell must be a unique iCell. A warning is issued if there are iCells that are not listed on the `background` or `matcell` entries. It is a fatal error if a material appears in the `LNK3DNT` file that is not mapped to an iCell.

8.2 DAWWG Card

DAWWG: Deterministic Automated Weight Window Generation

Form: DAWWG key=value(s) [key=value(s)] [...]

Required keys:

POINTS = I Number of points to sample in each direction
XSEC = <name>

Optional keys:

TALLY = I
BLOCK = {1,3,5,6} key=value [key=value,...]

The **points** entry specifies how many steps to randomly sample in each coordinate direction when estimating the composition of each gElement. A value of 10 means that $10^3=1000$ points will be sampled in each gElement.

The **block=<n>** entries are all optional and provide a pass-through of key/value pairs to the **PartiSn** input file that MCNP6 writes. It is incumbent on the user to ensure the keys and associated values are valid and that they are provided in the correct **PartiSn** block. For each **block**, all key/value pairs will be assigned to it until the next DAWWG keyword is found. Blocks and their contents can appear in any order. There is no requirement (or support) for **block-2** (?) or **block-4** (materials) entries. **Block-4** can be (and is) fully generated from other data in the MCNP input. The blocks will be written to the **PartiSn** input file in order, but the key/value pairs will be written in the order of appearance on the **DAWWG** card.

Block1 has some keys that may not be included on the **DAWWG** card as their values are determined from other data. These keys are: **igeom**, **nzone**, **im**, **it**, **jm**, **jt**, **km**, and **kt**. These are the geometry type, the number of materials, and the counts of coarse and total zones for each axis (**i j k**), respectively. Error checking for valid keys and values will be added later.

Table 1: PartiSn Blocks – Key/value types and defaults suitable for the **DAWWG** card.

Block	Key	Type	Default	Description
Block1 Dimension and Controls	ngroup	Integer	30	Number of energy groups
	isn	Integer	8	Sn order
	niso	Integer	0	Number of isotopes
	mt	Integer	1	Number of materials
	iquad	Integer	6	Quadrature: {1-9}
	fmmix	Integer	1	1=read composition from LNK3DNT file
	nosolv	Integer	0	1 = suppress solver module
	noedit	Integer	0	1 = suppress edit module
	nogeod	Integer	0	1 = suppress writing GEODST file
	nomix	Integer	0	1 = suppress writing mixing files
	noasg	Integer	0	1 = suppress writing ASGMAT file
	nomacr	Integer	0	1 = suppress writing MACRXS file
	noslnp	Integer	0	1 = suppress writing SOLINP file
Block3 Nuclear Data Type and Options	noedtt	Integer	0	1 = suppress writing EDITIT file
	noadjm	Integer	0	1 = suppress writing ADJMAC file
	lib	Text	ndilib	Name/Form of the XS data file
	libname	Text	mendf5	XS filename
	fissneut	Integer	0	Fission neutron flag
	lng	Integer	0	Number of the last neutron group
	balxs	Integer	0	XS balance control (-1,0,1)
	ntichi	Integer	0	MEDNF fission fraction to use

Block	Key	Type	Default	Description
Block5 Solver Input	ievt	Integer	1	Calculation type (0-4)
	isct	Integer	3	Legendre order
	ith	Integer	0	Direct (0) or adjoint (1) calculation
	trcor	Text	diag	
	ibl	Integer	0	Left boundary condition
	ibr	Integer	0	Right boundary condition
	ibt	Integer	0	Top boundary condition
	ibb	Integer	0	Bottom boundary condition
	ibfrnt	Integer	0	Front boundary condition
	ibback	Integer	0	Back boundary condition
	epsi	Real	0.0001	Convergence precision
	oitm	Integer	20	Maximum outer iteration count
	nosigf	Integer	0	1=inhibit fission multiplication
	srcacc	Text	dsa	Transport acceleration (DSA,TSA,NO)
	diffsol	Text	mg	Diffusion operator solver
	tsasn	Integer	0	Sn order for low order TSA sweeps
	tsepsi	Real	0.0	Convergence criteria for TSA sweeps
	tsaits	Integer	0	Maximum TSA iteration count
	tsabeta	Real	0.0	Scattering XS reduction for TSA
	ptconv	Integer	0	1 = Special criticality convergence scheme
	norm	Real	1.0	
	xsectp	Integer	0	XS print flag (0,1,2)
	fissrp	Integer	1	1 = Print fission source rate print
	sourcp	Integer	0	Source print flag (0,1,2,3)
	angp	Integer	0	1 = Print angular flux
	balp	Integer	0	1 = Print coarse-mesh balance tables
	raflux	Integer	0	1 = Prepare angular flux file
	rmflux	Integer	0	1 = Prepare flux moments file
	avatar	Integer	0	1 = Prepare special XMFLUXA file
	asleft	Integer	0	i = Right-going flux at plane i
	asrite	Integer	0	i = Left-going flux at plane i
	asbott	Integer	0	j = Top-going flux at plane j
astop	Integer	0	j = Bottom-going flux at plane j	
asfrnt	Integer	0	k = Back-going flux at plane k	
asback	Integer	0	k = Front-going flux at plane k	
npeg	Integer	0	0/1/2: Downscatter source iteration method and energy decomposition	
Block6 Edit Controls	massed	Integer	1	1 = Mass edits
	pted	Integer	0	1 = Edits by fine mesh
	zned	Integer	0	1 = Edits by (edit) zone
	rzflux	Integer	0	1 = Write a-flux file
	rzmlfx	Integer	0	1 = Write b-flux file
	edoutf	Integer	3	ASCII output files control [-3:3]
	byvolp	Integer	0	1 = Printed point reaction rates scaled by mesh volume
	ajed	Integer	0	Regular (0) or Adjoint (1) edit
fluxone	Integer	0	1 = Flux override	