particleHP Format Guide

Release 1.0

Geant4 Collaboration

Nov 30, 2022

CONTENTS

CHAPTER

ONE

INTRODUCTION

1.1 About this document

This guide describes the format of the Geant4 particleHP database files, version G4NDL4.6 of the neutron database and G4TENDL1.4 of the charged particle database. This format has not changed for at least ten year, and it is not expected to change in the near future. In case it does, this guide will be updated.

This guide is maintained at the following address

http://geant4.org/...

We will use through this document many terms common to the Monte Carlo simulation terminology and specifically to the Geant4 terminology. If you are new to it, please read before, for example, the Geant4 documentation. We have tried though to make this document self-consistent,

1.2 Information provided in this document

The Geant4 particleHP database files are distributed together with the Geant4 code. Both databases G4NDL and G4TENDL are written with the same format and therefore the descriptions that can be found in this guide will be valid for both.

For each type of file being explained an example file from the G4NDL or G4TENDL directory have been chosen. The file format is explained in this guide walking through the selected file line by line. The explanation of the meaning of each line starts with the line number (*LINE nnnn:*), followed by the name of the variable of the Geant4 code (very often from the variable name it can be deduced the type of data), plus by a brief explanation when needed. Last, the explanation is always followed by a copy of the lines from the example file. Where an explanation of a variable name is needed a few lines have been added in the line explanation. Also the C++ file and the method which reads the explained data is written between parenthesis, but only when a C++ class/method, different than the one of the previous line, is used.

The files in the two database are compressed to facilitate the distribution. If you want to follow the explanation you will have to uncomprees the zipped files. You may use for example zlib-flate -uncompress < $G4NEUTRONHPDATA/Inelastic/F06/8_16_Oxygen.z > $G4NEUTRONHPDATA/Inelastic/F06/8_16_Oxygen

Remember that Geant4 always reads the compressed file. Then, if you want to change (at your own risk) the database file you would have to delete the zipped file and optionally you may compress your modified file.

1.3 File directories in the particleHP database

Each incident particle data is written in a separate directory, G4NDL for neutron and G4TENDL/Proton G4TENDL/Deuteron G4TENDL/Triton G4TENDL/He3 and G4TENDL/Alpha for each of the charged particles. To use this databases in your Geant4 application, you must set the corresponding environment variables to point to these directories: G4NEUTRONHPDATA, G4PROTONHPDATA, G4DEUTERONHPDATA, G4TRITONHPDATA, G4H3HPDATA and G4ALPHAHPDATA.

The data for each incident particle is written in a separate directory, G4NDL for neutron and G4TENDL/Proton G4TENDL/Deuteron G4TENDL/Triton G4TENDL/He3 and G4TENDL/Alpha for each of the charged particles. To use the database in your Geant4 application you must set the corresponding environment variables to point to these directories: G4NEUTRONHPDATA, G4PROTONHPDATA, G4DEUTERONHPDATA, G4TRITONHPDATA, G4H3HPDATA and G4ALPHAHPDATA.

- *Inelastic*

while for neutron database the following directories can be found, each one correspoding to a physics process:

- *Inelastic*
- *Elastic*
- *Capture*
- *Fission*

plus the directories

- *ThermalScattering*: it has three subdirectories corresponding to the three different scattering processes: - *Coherent*: cross sections derived from Bragg edges and structure factors - *Incoherent*: cross sections derived from the bound cross section and Debye-Waller integral - *Inelastic*: uses the S(alpha,beta) formalism and the short-collision-time approximation
- *JENDL_HE*: Cross-section data set for a high precision (based on JENDL_HE evaluated data
- *IsotopeProduction*: isotope production cross sections

Each process directory contains two subdirectories:

- *CrossSection*: total cross sections
- *FS*: double differential cross sections

For the Inelastic process the information for each channel is written in a separate subdirectory with name Fnn, where *nn* are the Geant4 channel ID. For neutrons the channel IDs and the corresponding MT numbers of the ENDF format are:

- *F01*: (n,n') 50-90,91,4
- *F02*: (anything) 5
- *F03*: (n,2nd) 11
- *F04*: (n,2n) 16
- *F05*: (n,3n) 17
- *F06*: (n,na) 22
- *F07*: (n,n3a) 23
- *F08*: (n,2na) 24
- *F09*: (n,3na) 25
- *F10*: (n,np) 28
- *F11*: (n,n2a) 29
- *F12*: (n,2n2a) 30
- *F13*: (n,nd) 32
- *F14*: (n,nt) 33
- *F15*: (n,n3He) 34
- *F16*: (n,nd2a) 35
- *F17*: (n,nt2a) 36
- *F18*: (n,4n) 37
- *F19*: (n,2np) 41
- *F20*: (n,3np) 42
- *F21*: (n,n2p) 44
- *F22*: (n,npa) 45
- *F23*: (n,p) 600-648,649,103
- *F24*: (n,d) 650-698,699,104
- *F25*: (n,t) 700-748,749,105

- *F26*: (n,3He) 750-798,799,106
- *F27*: (n,a) 800-848,849,107
- *F28*: (n,2a) 108
- *F29*: (n,3a) 109
- *F30*: (n,2p) 111
- *F31*: (n,pa) 112
- *F32*: (n,d2a) 113
- *F33*: (n,t2a) 114
- *F34*: (n,pd) 115
- *F35*: (n,pt) 116
- *F36*: (n,da) 117

For a charged particle just change the first *n* by the corresponding particle name. These are the only channels that can be simulated with Geant4; if you have data from a different channel and want that it is read by the ParticleHP package, please contact the Geant4 authors for instructions.

We do not describe in this guide the functionality of the particleHP code, that is, how these variables are used. If you are interested in this, you should start by looking at the name of variables, as usually they are explicative of their functionality. You can also navigate through the Geant4 C++ class that is indicated at the corresponding line of this guide to read the C++ files and check where and how the variables are used.

CHAPTER

TWO

TOTAL CROSS SECTION FILES

Files in a *CrossSection* directory contain the total cross section of the interaction of the corresponding incident particle with an isotope. All these files have a very similar format. We will use the file at *G4NDL4.6/Inelastic/CrossSection/8_16_Oxygen* as example.

2.1 File format

// G4ParticleHPIsoData starts reading database file

LINES 1-2: >> dummy >> dummy;
(*G4ParticleHPIsoData::Init()*)

```
0
0
```

LINE 3: >> nData;

```
1354
```

Loops *nData* pairs energy / XS, and fills container for this data (theChannelData = new G4ParticleHPVector;). The interpolation scheme does not appear in this file and therefore it will be LINLIN, see Appendix section on interpolation schemes

LINES 4-455: >> x >> y: Energy / XS pairs
(*G4ParticleHPVector::Init()*)

```
2.355319e+06 0.000000e+00 2.400000e+06 1.827800e-37 2.500000e+06
7.093300e-19
```

... sectnum:: :start:

CHAPTER

THREE

INELASTIC CHANNEL FILES

A *Fnn* directory contains the total cross section of the corresponding reaction channel and the double differential cross sections (cross section vs energy and angle of outcoming particles). All the files in these directories have a very similar format.

There are two main types of files, those that correspond to channels with only one particle product, namely *F01*, *F23*, *F24*, *F25*, *F26* and *F27*, and those with several particle products.

3.1 Multiple product particles file

We use the file *G4NDL4.6/Inelastic/F06/8_16_Oxygen* (n,na) as example, and other files to illustrate the different options of a parameter that can be found in other *Fnn* files. Where there is a parameter with different options but only one is explained, we have added the word *(OPT)*. Those options are described in the section *Files with different options*.

LINE 1: `>> infoType`: not used
(*G4ParticleHPInelasticBaseFS::Init()*)

```
1
```

LINE 2: `>> dataType` (MF for ENDF format): =3: channel cross sections. (OPT)
(The data in the total cross section file is repeated here)

```
3
```

Other posible formats and the C++ class that manages each one are:

=4: angular distributions (*G4ParticleHPAngular*)
=5: energy distributions (*G4ParticleHPEnergyDistribution*)
=6: double-differential XSs=energy-angle (*G4ParticleHPEnAngCorrelation*)
=12: Mean final state photons distributions (*G4ParticleHPPhotonDist*)
=13: Partial final state photons distributions (*G4ParticleHPPhotonDist*)
=14: Angular final state photons distributions (*G4ParticleHPPhotonDist*)

=15: Energies final state photons distributions (*G4ParticleHPPhotonDist*)

LINE 3: `>> Qvalue >> dummy`. Qvalue is only used to check if(Qvalue<1.*CLHEP::keV && Qvalue>-1.*CLHEP::keV)
LINE 3: `>> total`; Number of energy-XS pairs

```
-7161950 0 77
```

LINES 4-29: energy-XS pairs. LINLIN interpolation

```
7.613591e+06 0.000000e+00 1.025000e+07 2.740298e-06
1.045000e+07 4.767337e-04
...
```

LINE 30: `>> infoType`: not used

```
1
```

LINE 31: `>> dataType`: MF=6: double-differential XSs=energy-angle

```
6
```

LINE 32: `>>targetMass>>frameFlag>>nProducts;`
(*G4ParticleHPEnAngCorrelation::Init()*)

```
15.8575 2 2
```

if(frameFlag==1) // target rest (LAB)
else if(frameFlag==2) // CMS
else if(frameFlag==3) // if A<=4 CMS, else LAB
nProducts = number of outgoing particles

LOOP *nProducts*:

LINES 33-34: `>> theMassCode>>theMass>>theIsomerFlag>>theDistLaw >> theGroundStateQValue>>theActualStateQValue;`
(*G4ParticleHPProduct::Init()*)

```
1.000000e+00 1.000000e+00 0 1
-7.161950e+06 -7.161950e+06
```

theMassCode = 1000*A+Z (1.000000e+00 = neutron)
theMass = in units of neutron mass

theIsomerFlag: not used
theGroundStateQValue: not used
theDistLaw =0: distribution not known, use E-independent, isotropic (OPT)
(*G4ParticleHPIsotropic*)

Other posible formats and the C++ class that manages each one are:

=1: Continuum energy-angular distribution (*G4ParticleHPContEnergyAngular*)
=2: Discrete 2-body scattering (*G4ParticleHPDiscreteTwoBody*)
=3: Isotropic emission (*G4ParticleHPIsotropic*)
=4: Discrete 2-body recoil modification // not used for now. (*G4ParticleHPDiscreteTwoBody*)
=5: charged particles only // to be used in a later stage.
=6: N-Body phase space (*G4ParticleHPNBodyPhaseSpace*)
=7: Laboratory angular energy parametrisation (*G4ParticleHPLabAngularEnergy*)

LINES 35-43: production XS for this product
LINE 35: number of XS points
`>> total;`
(*G4ParticleHPVector::Init()*, from *G4ParticleHPProduct::Init()*)

```
3
```

LINE 36: interpolation scheme
(*G4InterpolationManager::Init()*)

```
>> nRanges; // number of interpolation ranges
   1
```

LINE 37: interpolation scheme to be used by interpolation manager
scheme = new G4InterpolationScheme[nRanges];

LOOP *nRanges*:

```
>>range[i]; // late point for which scheme i is used
>>it; // type of interpolation scheme (2=LINLIN)
3 2
```

LINE 38: list of energy-XS pairs

```
7.613591e+06 1.000000e+00 3.000000e+07 1.000000e+00
1.500000e+08 1.000000e+00
```

LINES 39-: double differential cross sections for all products

LINE 39: `>> theTargetCode >> theAngularRep >> theInterpolation >> nEnergy;`

(*G4ParticleHPContEnergyAngular::Init()*)

```
8016 2 1 31
```

theAngularRep: Interpolation scheme number =2: use Kallbach-Mann

theInterpolation : interpolation scheme

nEnergy : number of particle energies for which double differential XS is given

LINES 40-41: interpolation manager (see above)

```
1
31 2
```

LOOP INCIDENT PARTICLE ENERGIES:

LINE 42: `>> theEnergy >> nEnergies >> nDiscreteEnergies >> nAngularParameters`: starts energy data

```
7.61359e+06 2 0 2
```

theEnergy: incident particle energy

nEnergies: number of product energies

nDiscreteEnergies: number of discrete product energies

nAngularParameters: number of parameters used to describe angular distribution (2 for Continuum energy-angular distribution)

LINE 43: double differential cross section: incident particle energy / 2 parameters of Kallmann-Bach angular distribution

```
0.000000e+00 1.417416e-04 0.000000e+00 7.055094e+03
0.000000e+00 0.000000e+00
```

LINES 44-62: next energy

```
1.025e+07 35 0 2
```

```
0.000000e+00 1.665152e-06 8.374084e-01 7.055094e+03
3.030595e-06 7.955380e-01
```

LINES 1127-: next particle

```
2.004000e+03 3.968220e+00 0 1
```

LINE 2088: `>> infoType`: not used

```
1
```

LINE 2089: dataType MF=13: Partial final state photon distributions (*G4ParticleHPPhotonDist*)

```
13
```

LINE 2090: photon energy
(*G4ParticleHPPhotonDist::InitPartials()*) : Partial final state photons distributions

```
>> nDiscrete >> targetMass;
>>theGammas[i]>>theShells[i]>>isPrimary[i]>>disType[i]; (LOOP
nDiscrete)
1 15.8575 4.438000e+06 4.438000e+06 0 2 143
```

theGammas[i]: product gamma energy
disType[i]; distribution type : =1: continuum, =2: discrete

LINES 2090-2140: partial cross sections
LINE 2090: `>> total`; (=143, from line 2090)
LINE 2091-92: interpolation scheme
LINES 2093-2140: energy-XS pairs

LINE 2141: infoType: not used

```
1
```

LINE 2142: dataType MF=14: Angular final state photons distributions

```
14
```

LINE 2143: `>> isoFlag;`
(*G4ParticleHPPhotonDist::InitAngular()*) : Angular final state photons distributions

```
    0
```

isoFlag : should be != 1, else it may give a warning

LINE 2144: `>> tabulationType >> nDiscrete2 >> nIso;`

```
    1 1 0
```

tabulationType : type of data to be read. 1 = *G4ParticleHPLegendreTable*, 2 = *G4ParticleHPAngularP* (OPT)

LOOP *nIso* (=0): LOOP *nDiscrete2-nIso* if(tabulationType==1):

LINE 2145: `>> theGammas[i] >> theShells[i] >> nNeu[i-nIso];`

```
    4.438000e+06 4.438000e+06 92
```

nNeu : number of neutral particles

LOOP *nNeu[i-nIso]*:

LINES 2146-2331: angular data (1 = *G4ParticleHPLegendreTable*)=
LINES 2146-2147: interpolation scheme
LINE 2148: `>> eNeu >> nPoly;`
(*G4ParticleHPLegendreTable::Init()*)

```
    7.613591e+06 4
```

nPoly: number of coefficients

LINE 2149: Legendre coefficients

```
    0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
```

LINES 2150-2331: angular data for the other energies
See above for MF=6 format

3.2 Single particle product files

We use the file *G4NDL4.6/Inelastic/F01/8_16_Oxygen* (n,n') as example, and other files to illustrate the differet options of a parameter that can be found in other *Fnn* files. Where there is a

parameter with different options but only one is explained, we have added the word *(OPT)*. Those options are described in the section *Files with different options*.

File starts with the total cross section

LINE 1: `>> infoType`: not used
(*G4ParticleHPInelasticCompFS::Init()*)

```
    1
```

LINE 2: `>> dataType`: MF=3: channel cross sections.
(The data in the total cross section file is repeated here)

```
    3
```

LINE 3: `>> sfType >> dummy;`

```
    4 0
```

For this file there are several sets of total cross sections, one for each of the subchannels (different MF values, each one identified by the corresponding sfType). For each subchannel there is a different angular distribution. The code will select one of the subchannels proportionally to their XS for the incident particle energy. sfType=4 will be used for the final state photons (MF=13)

LINES 4-5: `>> dqi >> ilr;`
`>> total;`

```
    -6049400 0 281
```

dqi: Reaction Q value. Used for calculating excitation energy of residual (if MF=4 and MF=5 data is found)
ilr: Only used if the reaction is with Carbon, using NRESP model (NRESP is a Monte Carlo simulation code developed at the Physikalisch-Technische Bundesanstalt (PTB), Germany, to study the response of organic scintillation detectors to fast neutrons between 0.02 and 20 MeV)

LINES 5-98: energy-XS pairs. LINLIN interpolation

```
    6.430885e+06 0.000000e+00 6.475000e+06 4.648382e-07
    6.500000e+06 2.000000e-04
```

LINES 99-560: repeated above structure for sfType = 51,52,53,54,55,56,57,91

LINE 561: `>> infoType`: not used

```
1
```

LINE 562: `>> dataType`: MF=4: angular distributions (*G4ParticleHPAngular*)channel cross sections.

```
4
```

LINE 563: `>> sfType >> dummy;`

```
51 0
```

LINE 564: `>> theAngularDistributionType >> targetMass;`

`>> frameFlag;`

(*G4ParticleHPAngular::Init()*)

```
1 15.8575 2
```

theAngularDistributionType = 0: isotropic (OPT)

= 1: use *G4ParticleHPLegendreStore*

= 2: use *G4ParticleHPPartial*

frameFlag : see above

LINE 565: `>> nEnergy;`

LINES 566-567: interpolation scheme

```
1
2 2
```

LOOP *nEnergy*. For each one create a *G4ParicleHPLegendreTable*:

LINES 568-647: `>> temp >> energy >> tempdep >> nLegendre;`

```
0.000000e+00 6.430885e+06 0 2
```

energy : incident particle energy

nLegendre: numbre of Legendre coefficients

LOOP *nLegendre* LegendreTable coefficients:

LINE 569: `>> coeff;`

```
0.000000e+00 0.000000e+00
```

LINES 570-931: repeat above for sfType 52,53,54,55,56,57

LINE 932: `>> infoType`: not used

```
1
```

LINE 933: `>> dataType`: MF=6: double-differential XSs=energy-angle `6`

LINE 934: `>> sfType >> dummy;`

```
91 0
```

LINES 935-3028: See above for MF=6 format (it repeats for neutron and later for gamma)

LINE 3029: `>> infoType`: not used

```
1
```

LINE 3030: `>> dataType`: MF=13: Partial final state photons distributions (*G4ParticleHPPhotonDist*)

LINE 3031: `>> sfType >> dummy;`

```
4 0
```

LINE 3032: `>> nDiscrete >> targetMass`

```
    >>theGammas[i]
15 15.8575 282
```

LINES 3033-3202: as nDiscrete != 1 read total XS (with 282 points)

LOOP *nDiscrete*:

LINES 3203-4297: nDiscrete gamma info `>>theGammas[i]>>theShells[i]>>isPrimary[i]>>disType[i];` (LOOP *nDiscrete*)

```
        7.116850e+06 7.116850e+06 0 2 255
```

See above for MF=13 format

LINE 4298: `>> infoType`: not used

```
1
```

LINE 4299: `>> dataType`: MF=6: double-differential XSs=energy-angle

```
14
```

LINE 4300: `>> sfType >> dummy;`

```
4 0
```

LINE 4301: `>> isoFlag;`

```
0
```

LINE 4302: `>> tabulationType >> nDiscrete2 >> nIso;`

```
1 15 10
```

LOOP *nIso*:

LINES 4303-4312: `>> theGammas[i] >> theShells[i];`

LOOP *nDiscrete2-nIso*:

LINES 4313:5239

See above for MF=14 format

3.3 Files with different options

3.3.1 MF:

3.3.1.1 MF=5: energy distributions (*G4ParticleHPEnergyDistribution*)

We use the file *G4NDL4.6/Inelastic/F06/12_24_Magnesium* (n,na)

LINE 22: `>> infoType`: not used
(*G4ParticleHPInelasticBaseFS::Init()*)

```
1
```

LINE 23: `>> dataType`

```
5
```

LINE 24: `>> dummy >> theNumberOfPartials;`
(*G4ParticleHPEnergyDistribution::Init()*)

```
23.779 1
```

LOOP *theNumberOfPartials*:

LINE 25: `>> theRepresentationType;`

```
1 2
```

Other possible formats and the C++ class that manages each one are:
theRepresentationType=

- 0: *G4ParticleHPArbitaryTab*
- 5: *G4ParticleHPEvapSpectrum*
- 7: *G4ParticleHPFissionSpectrum*
- 9: *G4ParticleHPSimpleEvapSpectrum*
- 11: *G4ParticleHPWattSpectrum*
- 12: *G4ParticleHPMadlandNixSpectrum*

LINE 25: `>> nDistFunc;`

(*G4ParticleHPArbitaryTab::Init()*)

LINES 26-27: interpolation scheme

```
1
2 2
```

LOOP *total*:

LINE 28: `>> x >> y`

```
9.706250e+06 1.000000e+00 2.000000e+07 1.000000e+00
```

LINE 29: `>> nDistFunc`: number of incoming n energy points
(*G4ParticleHPArbitaryTab::Init()*)

```
10
```

LINES 30-31: interpolation scheme

```
1
10 22
```

LOOP *nDistFunc* (=10):

LINE 32: `>> currentEnergy;`

```
1
34 1
```

LINES 32-47: first theDistFunc data (*G4ParticleHPVector*)

```
0.000000e+00 0.000000e+00 1.000000e-08 0.000000e+00
1.778280e-08 2.766140e-07
```

LINES 47-202: other 9 theDistFunc data

3.3.1.2 MF=12: Mean final state photons distributions (*G4ParticleHPPhotonDist*)

We use the file *G4NDL4.6/Inelastic/F01/8_17_Oxygen* (n,na).

LINE 3672: `>> infoType`: not used
(*G4ParticleHPInelasticCompFS::Init()*)

```
1
```

LINE 3673: `>> dataType`: MF=12: Mean final state photons distributions
(*G4ParticleHPPhotonDist*)

```
12
```

LINE 3674 `>> sfType >> dummy;`

```
51 0
```

LINE 3675: `>> repFlag`

```
>> targetMass
```

(*G4ParticleHPPhotonDist::InitMean()*) : mean yields

```
2 16.8531
```

if(repFlag==2) | LINE 3676: `>> theInternalConversionFlag;` | `1`

LINE 3677: `>> theBaseEnergy;`

```
>> theInternalConversionFlag;
>> nGammaEnergies;
870730 1 1
```

LOOP *nGammaEnergies* (=1):

if(theInternalConversionFlag == 1)
LINE 3678: `>> theLevelEnergies[ii] >>`
`theTransitionProbabilities[ii];`

```
0 1
```

if(theInternalConversionFlag == 2) `>> theLevelEnergies[ii] >> theTransitionProbabilities[ii] >> thePhotonTransitionFraction[ii]`

LINES 3679-3881: repeat MF=12 for fsType=51-80

3.3.1.3 MF=15: Energies final state photons distributions (*G4ParticleHPPhotonDist*)

We use the file *G4NDL4.6/Inelastic/F06/12_24_Magnesium* (n,na)

LINE 217: `>> infoType`: not used
(*G4ParticleHPInelasticBaseFS::Init()*)

```
1
```

LINE 218: `>> dataType`

```
5
```

LINE 219: `>> nPartials;`
(*G4ParticleHPPhotonDist::InitEnergies()*) : energy distributions

```
1
```

LOOP *nPartials*:

LINE 220: `>> dummy;`

```
1 2
```

LINE 220: `>> total`
(*G4ParticleHPVector::Init()*)

LINES 221-222: interpolation scheme

```
1
2 2
```

LINE 223: x-y pairs

```
9.706250e+06 1.000000e+00 2.000000e+07 1.000000e+00
```

LINE 224: `>> nen;` Number of partial XS

```
9
```

LINES 225-226: interpolation scheme

```
1
9 2
```

LOOP *nen*:

LINE 227: `>> e`: energy

```
9.706250e+06 8
```

LINE 227: `>> neg`
(*G4ParticleHPPartial::InitData()*)
neg: number of points energy-probability

LINES 228-229: interpolation scheme

```
1
8 1
```

LOOP *neg*:

LINE 230: 233: `>> eg >> pg`: energy-probability

```
0.000000e+00 0.000000e+00 2.500000e+05 0.000000e+00
5.000000e+05 0.000000e+00
```

LINES 234-312: repeat for the other 8 partial XS

```
1
```

3.3.2 theDistLaw

3.3.2.1 theDistLaw=2: Discrete 2-body scattering (*G4ParticleHPDiscreteTwoBody*)

We use the file *G4TENDL1.4/Proton/Inelastic/F01/1_3_Hydrogen* (p,p')

LINES 64-65: `>> theMassCode>>theMass>>theIsomerFlag>>theDistLaw >> theGroundStateQValue>>theActualStateQValue;`
(*G4ParticleHPInelasticCompFS::Init()*)

```
1.000000e+00 1.000000e+00 0 2
-7.638700e+05 -7.638700e+05
```

LINES 66-69: read *G4ParticleHPVector* (see case theDistLaw = 1)

```
2
1
2 2
```

LINE 70: `>> nEnergy;`
(*G4ParticleHPDiscreteTwoBody::Init()*)

```
67
```

LINES 71-72: interpolation scheme

```
1
67 2
```

LINE 73: `>> energy >> aRep >> nCoeff;`

```
1.0191e+06 0 6
```

aRep : *G4ParticleHPLegendreTable* coefficients

LOOP *nCoeff*:

LINE 74: `>> y`; *G4ParticleHPLegendreTable* coefficients

```
0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
0.000000e+00 0.000000e+00
```

LINES 75-206: LOOP *nEnergy* (67 energies)

3.3.2.2 theDistLaw=3: Isotropic emission (*G4ParticleHPIsotropic*)

We use the file *G4NDL4.6/Inelastic/F01/28_58_Nickel* (n,n')

LINES 3448-3449: `>> theMassCode>>theMass>>theIsomerFlag>>theDistLaw >> theGroundStateQValue>>theActualStateQValue;`
(*G4ParticleHPInelasticCompFS::Init()*)

```
1.000000e+00 1.000000e+00 0 3
0.000000e+00 -2.903000e+06
```

LINES 3450-3453: read *G4ParticleHPVector* (see case theDistLaw = 1)

```
2
1
2 2
2.953540e+06 1.000000e+00 2.000000e+07 1.000000e+00
```

LINE 209: *G4ParticleHPIsotropic*

3.3.2.3 theDistLaw=4: Discrete 2-body recoil modification // not used for now

We use the file *G4TENDL1.4/Proton/Inelastic/F01/1_3_Hydrogen* (p,p')

LINES 207-208: `>> theMassCode>>theMass>>theIsomerFlag>>theDistLaw >> theGroundStateQValue>>theActualStateQValue;`
(*G4ParticleHPInelasticCompFS::Init()*)

```
2.003000e+03 2.989000e+00 0 4
-7.638700e+05 -7.638700e+05
```

LINES 209-210: read *G4ParticleHPVector* (see case theDistLaw = 1)

```
2
1
2 2
```

Distribution is set to 0, so no angular distribution is used

3.3.2.4 theDistLaw=5: charged particles only - to be used in a later stage

No data file has theDistLaw=5

3.3.2.5 theDistLaw=6: N-Body phase space (*G4ParticleHPNBodyPhaseSpace*)

We use the file *G4TENDL1.4/Proton/Inelastic/F10/1_2_Hydrogen* (p,np)

LINES 207-208: `>> theMassCode>>theMass>>theIsomerFlag>>theDistLaw >> theGroundStateQValue>>theActualStateQValue;`
(*G4ParticleHPInelasticBaseFS::Init()*)

```
1.000000e+00 1.000000e+00 0 6 -2.224585e+06 -2.224585e+06
```

LINES 209-210: read *G4ParticleHPVector* (see case theDistLaw = 1)

```
2
1
2 2
3.337123e+06 1.000000e+00 1.500000e+08 1.000000e+00
```

LINE 211: `>> theTotalMass >> theTotalCount;`
(*G4ParticleHPNBodyPhaseSpace::Init()*)

```
2.99862 3 1.001000e+03 9.986200e-01 0 6
```

(the 4th last values are for next product : `>> theMassCode>>theMass>>theIsomerFlag>>theDistLa`
)

3.3.2.6 theDistLaw=7: Laboratory angular energy paraetrisation (*G4ParticleHPLabAngularEnergy*)

We use the file *G4NDL4.6/Inelastic/F04/1_2_Hydrogen* (n,2n)

LINES 20-21: `>> theMassCode>>theMass>>theIsomerFlag>>theDistLaw >> theGroundStateQValue>>theActualStateQValue;`
(*G4ParticleHPInelasticBaseFS::Init()*)

```
1.000000e+00 1.000000e+00 0 7
-2.224566e+06 -2.224566e+06
```

LINES 22-25: read *G4ParticleHPVector* (see case theDistLaw = 1)

```
2
1
```

```
    2 1
    3.338632e+06 2.000000e+00 3.000000e+07 2.000000e+00
```

LINE 26: `>> nEnergies;`
(*G4ParticleHPLabAngularEnergy::Init()*)

```
    54
```

LINES 27-29: interpolation scheme

```
    1
    54 2
```

LOOP *nEnergies*:

LINE 30: `>> theEnergies[i];`
`>> nCosTh[i];`

```
    3.338632e+06 37
```

LINES 31-32: interpolation scheme

```
    1
    37 2
```

LINE 33: `>> label;`

```
    -1.000000e+00 8
```

LINES 33-37: data for the theData[0][0] (*G4ParticleHPVector*'s)

LINES 38-256: repeat 33-37 for the other 36 nCosTh[i];

LINES 257-29261: repeat 30-256 for the other 53 energies

3.3.3 theAngularDistributionType

3.3.3.1 theAngularDistributionType=0: isotropic

We use the file *G4NDL4.6/Inelastic/F01/24_50_Chromium* (n,n')

LINE 609: `>> theAngularDistributionType >> targetMass;`
`>> frameFlag;`
(*G4ParticleHPInelasticCompFS::Init()*)

```
0 49.517 2
```

No more data is read, it passes to the next MF

3.3.3.2 theAngularDistributionType=2: use G4ParticleHPPartial

We use the file *G4NDL4.6/Inelastic/F04/12_24_Magnesium* (n,2n)

LINE 8: `>> theAngularDistributionType >> targetMass;`
`>> frameFlag;`
(*G4ParticleHPAngular::Init()*)

```
2 23.779 1
```

LINE 9: `>> nEnergy;`

```
2
```

LINES 10-11: interpolation scheme

```
1
2 2
```

LINE 12: `>> temp >> energy >> tempdep;`

```
0.000000e+00 1.722580e+07 0 2
```

LINE 12: `>> neg`; read data for this energy

(*G4ParticleHPPartial::InitData()*) | `0.000000e+00 1.722580e+07 0 2`

LINES 13-14: interpolation scheme

```
1
2 2
```

LOOP *neg*:

LINE 15: `>> eg >> pg;`

```
-1.000000e+00 5.000000e-01 1.000000e+00 5.000000e-01
```

LINES 16-19: read data for other energy

```
0.000000e+00 2.000000e+07 0 2
```

3.3.4 G4ParticleHPPhotonDist::repFlag

3.3.4.1 repFlag = 1

We use the file *G4NDL4.6/Inelastic/F04/12_24_Magnesium* (n,2n)

LINE 93: `>> repFlag`

```
>> targetMass
```

(*G4ParticleHPPhotonDist::InitMean()*) : Read mean yields

```
1 23.779
```

LINE 94: `>> nDiscrete;`

```
1
```

LINE 95: `>> disType[i]>>energy[i];`

```
1 0 5
```

LINES 95-99: *G4ParticleVector* data

CHAPTER

FOUR

ELASTIC CHANNEL FILES

Files in a *Elastic/FS* directory contain the double differential cross sections of elastic interactions. All the files in these directories have a very similar format. We use the file *G4NDL4.6/Elastic/FS/8_16_Oxygen*

4.1 File format

LINE 1: `>> repFlag >> targetMass >> frameFlag;`
(*G4ParticleHPElasticFS::Init()*)

```
    3 15.8575 2
```

LINE 2: `>> nEnergy_Legendre;`

```
    1161
```

LINES 3-4: interpolation scheme

LOOP *nEnergy_Legendre* (create a *G4ParticleHPLegendreStore*):

LINE 5: `>> temp >> energy >> tempdep >> nLegendre;`

```
0.000000e+00 1.000000e-05 0 8
```

LOOP *nLegendre*:

LINES 6-7: `>> coeff;`

```
0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00
0.000000e+00
0.000000e+00 0.000000e+00
```

Set coefficients of first G4ParticleHPLegendreTable of first *G4ParticleHPLegendreStore*

LINES 8-3487: coefficients of the other 1160 *G4ParticleHPLegendreStore*

LINE 3488: `>> nEnergy_Prob;`

```
14
```

LINES 3489-3490: interpolation scheme

```
1
14 2
```

LOOP *nEnergy_Prob*:

LINE 3491: `>> temp >> energy >> tempdep >> nPoints;`

```
0.000000e+00 3.000000e+07 0 73
```

create nEnergy_Prob *G4ParticleHPPartial*

LINES 3492-3493: interpolation scheme of first *G4ParticleHPPartial*

```
1
73 4
```

LOOP *nPoints*:

LINES 3494-3518: `>> costh >> prob;`

LINES 3519-3882: data of the other 13 *G4ParticleHPPartial*

CHAPTER

FIVE

CAPTURE FILES

The *Capture/FS* directory contains the files with the distributions of the photons created in a capture process. All the files in this directory have a very similar format.

A file in the *FS* directory describes the cross section of gamma production as well as the outgoing gamma properties. It can divided in three blocks: - Mean yields - Angular final state photons distributions - Energy distributions

There is a second set of files, in the *FSMF6* directory, which contain double-differential cross sections.

We use the file *G4NDL4.6/Capture/FS/7_14_Nitrogen*

5.1 *FS files*

5.1.1 Mean yields

LINE 1: `>> repFlag`
`>> targetMass`
(*G4ParticleHPPhotonDist::InitMean()*)
`1 13.8828`
All Capture files have repFlag = 1

LINE 2: `>> nDiscrete;`
`59`

LOOP *nDiscrete*:

LINE 3: `>> disType[i]>>energy[i];`

```
    2 1.08291e+07 38
```

LINES 3-18: *G4ParticleVector* data containing yield XS

LINES 19-262: the other 58 *G4ParticleVector*'s

5.1.2 Angular final state photons distributions

LINE 263: `>> isoFlag;`
(*G4ParticleHPPhotonDist::InitAngular()*)

```
    0
```

LINE 264: `>> tabulationType >> nDiscrete2 >> nIso;`

```
    1 59 58
```

LOOP *nIso* (tabulationType==1):

LINES 265-322: `>> theGammas[i] >> theShells[i];`

```
    1.06978e+07 0
```

LOOP *nDiscrete2-nIso*:

LINE 323: `>> theGammas[i] >> theShells[i] >> nNeu[i-nIso];`

```
    1.082910e+07 0.000000e+00 39
```

Create a *G4ParticleHPLegendreTable*

LINES 324-325: interpolation scheme

```
    1
    39 2
```

LINE 326: `>> eNeu >> nPoly;`

(*G4ParticleHPLegendreTable::Init()*)

```
    1.000000e-05 2
```

nPoly: number of coefficients

LINE 327: Legendre coefficients

```
    0.000000e+00 0.000000e+00
```

LINES-328-403: other 38 Legendre coefficients

```
    1.200000e+06 2
```

....

5.1.3 Energy distributions

LINE 404 : `>> nPartials;`

(*G4ParticleHPPhotonDist::InitEnergies()*)

```
    1
```

nPartials: number of probabilities and partial cross sections

LOOP *nPartials*:

LINE 405: `>> dummy;`

```
    1 2
```

Read probabilities (*G4ParticleHPVector*)

LINES 405-408: interpolation manager and data of first probability

```
    1
    2 2
    1.000000e-05 1.000000e+00 1.500000e+08 1.000000e+00
```

LINES 409-411: `>> nen;`

```
    14
```

nen: number of subpartials (partials of the first partial)

LINES 410-411: interpolation scheme and data of first subpartial of first partial

```
1
14 2
```

LOOP *nen*:

LINE 412: `>> e`: energy
(*G4ParticleHPPartial::Init()*)

```
1.000000e-05 11
```

LINE 412: `>> neg`

```
1.000000e-05 11
```

neg: number of points energy-probability

LINES 413-414: interpolation scheme of subpartial

```
1
11 1
```

LINES 415-418: energy-probability of first subpartial

```
0.000000e+00 1.277503e-09 7.500001e+05 3.147728e-08
1.250000e+06 1.582778e-08
```

LINES 419-629: data of the other 13 subpartials of first partial

LINE —-: data of the other 0 partials (only 1 partial in this file)

5.2 *FSMF6 files*

The information in these files is read by the C++ class *G4ParticleHPContEnergyAngular*, starting at (*G4ParticleHPContEnergyAngular::Init()*). You can find the format description in the corresponding section of the Inelastic chapter.

CHAPTER

SIX

APPENDIX

6.1 Database units

Energies are given in eV and cross sections in barns

6.1.1 XS interpolation schemes

The numbers in a file that sets the type of interpolation scheme corresponds to one of the following types:

> *START, HISTO, LINLIN, LINLOG, LOGLIN, LOGLOG, RANDOM, CSTART_, CHISTO, CLINLIN, CLINLOG, CLOGLIN, CLOGLOG, CRANDOM, USTART, UHISTO, ULINLIN, ULINLOG, ULOGLIN, ULOGLOG, URANDOM*

Therefore a value =0 : START, =1 : HISTO, . . .

6.2 C++ flux to open a database file

The line numbers correspond to Geant4 version 11.00.p02.

6.2.1 *Inelastic CrossSection*

- User physics list instantiates G4ParticleHPInelasticData

G4NeutronPHPBuilder.cc:93
if(theHPInelasticData==0) theHPInelasticData = new
G4ParticleHPInelasticData(G4Neutron::Neutron());

- G4ParticleHPInelasticData instantiates G4ParticleHPData

G4ParticleHPInelasticData.cc:106
theHPData = new G4ParticleHPData(theProjectile);

- G4ParticleHPData has one G4ParticleHPElementData (=theData) for each G4Element; initialises it for the corresponding G4Element

G4ParticleHPData.cc:58
(*theData[i]).Init((*(G4Element::GetElementTable()))[i], projectile, theDataDirVariable);

- G4ParticleHPElementData has one G4ParticleHPIsoData for each G4Isotope. For each one gets A,Z,M, RelativeAnbundanceVector)=frac) and calls

G4ParticleHPElementData.cc:76
UpdateData(A, Z, M, count++, frac, projectile, dataDirVariable);

- Initialize isotope data

G4ParticleHPElementData.cc:106
theIsotopeWiseData[index].Init(A, Z, M, abundance,projectile, dataDirVariable);

6.2.2 *Inelastic Fx*

- G4HadronicProcess::BuildPhysicsTable()

G4HadronicProcess.cc:182
theEnergyRangeManager.BuildPhysicsTable(p);

- G4EnergyRangeManager::BuildPhysicsTable

G4EnergyRangeManager.cc:181
hadi->BuildPhysicsTable(aParticleType);

-G4ParticleHPInelastic::BuildPhysicsTable (| G4ParticleHPInelastic.cc:548 | ((*theInelastic)[i])->Register(new G4ParticleHPNInelasticFS , “F06”); // calls all the channels by alphabetical order

- G4ParticleHPChannelList::Register

G4ParticleHPChannelList.cc:219
theChannels[theInitCount]->Register(theFS);

- G4ParticleHPChannel::Register

G4ParticleHPChannel.cc:128
UpdateData(A, Z, M, count++, frac, theProjectile);

- G4ParticleHPChannel::UpdateData

G4ParticleHPChannel.cc:163
theFinalStates[index]->Init(A, Z, M, theDir, theFSType, projectile);

- G4ParticleHPInelasticFS::Init

G4ParticleHPNAInelasticFS.cc:44 : "F06" is (n,na), therefore C++ class has a *NA*

- For multiple particle product the file reading starts in G4ParticleHPInelasticBaseFS::Init(A, Z, M, dirName, aFSType, projectile);

G4ParticleHPInelasticBaseFS.cc:77

- For single particle product the file reading starts in G4ParticleHPInelasticCompFS::Init(A, Z, M, dirName, aFSType, projectile);

G4ParticleHPInelasticCompFS.cc:86

6.2.3 *Capture FS*

- G4HadronicProcess::BuildPhysicsTable

G4HadronicProcess.cc:182
theEnergyRangeManager.BuildPhysicsTable(p);

- G4EnergyRangeManager::BuildPhysicsTable

G4EnergyRangeManager.cc:181
hadi->BuildPhysicsTable(aParticleType);

- G4ParticleHPCapture::BuildPhysicsTable

G4ParticleHPCapture.cc:221

- G4ParticleHPChannel::Register

G4ParticleHPChannel.cc:128

- G4ParticleHPChannel::UpdateData

G4ParticleHPChannel.cc:163

- G4ParticleHPCaptureFS::Init

G4ParticleHPCaptureFS.cc:369