

ENDF2ACE User Guide

February 9, 2017

Authors :

- L. Perrot, IPN Orsay
- O. Méplan, LPSC Grenoble

1st version: September 2004,
Last Revision: January 2009

Contents

1	Preliminary Comments	3
1.1	Processing ENDF/B VII data	3
2	Introduction	4
2.1	Why a new code ?	4
2.1.1	Data : the 0 level	4
2.1.2	Data Evaluation	4
2.1.3	The Evaluated Nuclear Data File (ENDF) format	4
2.1.4	What append after the ENDF format ?	5
2.1.5	Why a new code !?!?!?	5
2.2	How convert nuclear data into the appropriate form for my application ?	6
2.2.1	NJOY	6
2.2.1.1	Introduction :	6
2.2.1.2	NJOY Modules :	7
2.2.1.3	Data Flow in NJOY	7
2.2.2	MAKXSF	11
3	The ENDF2ACE utility	12
3.1	Data Base in ACE format	12
3.1.1	BuildScript method	13
3.1.2	Converting ASCII ACE in binary	14
3.2	The Thermal Neutron Scattering data	15
4	Using the endf2ace	17
4.1	Input	17
4.2	The package	19
4.2.1	Installation	19
4.2.2	The command line	19
4.2.2.1	Beginners	19
4.2.2.2	Examples	20
4.3	Standard message	22
4.4	Ouput	25
4.5	Error messages	25

Chapter 1

Preliminary Comments

- The ENDF2ACE code installation and utilization is described in chapter 4.
- Using MAKXSf to convert ASCII ACE file in ACE binary file
 - check lines 1543 and 1544 in `endf2ace.cxx`, in the method `ENDF2ACE::BuildSpecs()` ; the number must be in agreement with your system installation (see MCNP user manual).
 - if you plane to use MURE with binary files generate by MAKXSf:
 - * if it has been compile with g77 (of GCC) there is nothing to do
 - * if you use a true FORTRAN compiler, you must be sure that fortran records are 1 byte long when reading/writing in unformatted files. For example, the default for Intel FORTRAN compiler (*ifort*) is 4 bytes ; then you must use the `-assume byterecl` flag of *ifort*. This remark is also true for the MCNP compilation using this compiler.

1.1 Processing ENDF/B VII data

- You must use the last NJOY version (UP259)
- The material number for ⁹⁹Tc find in the `ZAI_Mat.dat` file (line 145) has to be change: replace the **4331** value by **4325**.

Uncomment the line 1037 of `endf2ace.cxx` :

```
//input_file<<" \n"; // line to (un)comment if "***error in ptleg2***too many angles" or for ENDF/B 7
```

Chapter 2

Introduction

2.1 Why a new code ?

2.1.1 Data : the 0 level

Nuclear data are fundamental to the development and application of all nuclear sciences and technologies. Basic nuclear data, whether measured or calculated, follow a complex process of evaluation, correction and analysis before becoming “directly” available in applications.

2.1.2 Data Evaluation

As there is no predictive theories for neutron-induced reactions in the resonance energy range, the basic nuclear data have to be obtained through measurements at dedicated experimental facilities, such as linear particle accelerators for example. The measured raw data are then corrected for the experimental conditions, such as sample impurities, background effects, room temperature, sample geometry, detector efficiencies ...

However, the experimental data obtained are not directly suitable for application calculations. A thorough analysis of the data is necessary to produce a coherent set of usable data. Data evaluation, is a very difficult and long step to produce the coherent set of data for each nuclide. An evaluation is the process of analyzing experimentally measured cross-section data, combining them with the predictions of nuclear model calculations, and attempting to extract the “true” value of a cross-section (at 0 Kelvin). Parametrization and reduction of the data to tabular form produce an evaluated data set. If a written description of the preparation of a unique data set from the data sources is available, the data set is referred to as a documented evaluation.

2.1.3 The Evaluated Nuclear Data File (ENDF) format

Few countries, few experts carry out this painful work. But, since the end of the cold war (the winner is the free and democratic world), a cooperative effort of national laboratories, industry and universities leads to choose the ENDF formats and libraries for nuclear data. The ENDF formats are decided by the Cross Section Evaluation Working Group (CSEWG) and are maintained by the

National Nuclear Data Center (NNDC) in US. In the different countries which are doing data evaluation, everybody have accepted the ENDF format as the only one format.

What is the ENDF format ?

The ENDF format provides representations for neutron cross-sections and distributions, photon production from neutron reactions, a limited amount of charged-particle production from neutron reactions, photo-atomic interaction data, thermal neutron scattering data and radionuclide production and decay data (including fission products). In the last past decade, a new version of the format had delivered : the version 6 (ENDF-6) which allows higher incident energies, adds more complete descriptions of the distributions of emitted particles, and provides for incident charged particles and photo-nuclear data by partitioning the ENDF library into sub-libraries. Decay data, fission product yield data, thermal scattering data, and photo-atomic data have also been formally placed in sub-librairies.

The ENDF system was developped for the storage and retrieval of evaluated nuclear data to be used for applications of nuclear technology. These applications control many features of the system including the choice of materials to be included, the data used, the formats used, and the testing required before a library is released. An important consequence of this is that each evaluation must be complete for its intended application. If required data are not available for particular reactions, the evaluators should supply them by using systematics or nuclear models.

The ENDF system is logically divided into formats and procedures. Formats describe how the data are arranged in the libraries and give the formulas needed to reconstruct physical quantities such as cross-sections and angular distributions from the parameters in the library. Procedures are the more restrictive rules that specify what data types must be included, which format can be used in particular circumstances, and so on. Procedures are, generally, imposed by a particular organization, and library sanctioned by the CSEWG is referred to as ENDF/B. Other organizations may use somewhat different procedures, if necessary, but they face the risk that their libraries will not work with processing codes sanctioned by CSEWG.

2.1.4 What append after the ENDF format ?

Once the evaluated data sets have been prepared in ENDF format, they can be converted to forms appropriate for testing and actual applications using processing codes. Processing codes that generate group-averaged cross-sections for use in neutronics calculations from the ENDF library have been written. These codes include such functions as resonance reconstruction, Doppler broadening, multi-groups averaging, and/or rearrangement into specified interfaces. It is typically the case for the PREPRO code or NJOY code.

2.1.5 Why a new code ?!?!?!?

The present guide focuses only on how to process data in ENDF format in the adapted data format for the MCNP code.

MCNP code needs data in an appropriated format call ACE format (A Compact ENDF). The ACE format has evolved to include all the details of the ENDF representations for neutron and photon data. However, for sake of efficiency, the representation of data in ACE is quite different from that in ENDF. The fundamental difference is the use of random access with pointers to the various parts of the data. Other key differences include the use of union energy grids, equal-probability bins, and cumulative probability distributions (more details in Appendix F of the MCNP doc and NJOY doc p457-491 of PDF file). MCNP requires that all the cross-sections are given on a single union energy grid suitable for linear interpolation. The only solution to produce data in the ACE format is to use specific modules of NJOY code.

MCNP code needs also data process at a specific temperature. These data are produce with the NJOY code only from the adopted ENDF data format.

But NJOY is a certainly a too bigger code if one only needs data in ACE format at a specific temperature for a given nuclide. User gives to NJOY code an input file properly inform. After run, one get lots of files, but only few of them are really useful for the final objective (data in ACE format).

The main idea of the **ENDF2ACE** program is to limit the number of parameters given by the user to construct the NJOY input file, then run NJOY and keep only files use by MCNP. It is also possible to convert these ASCII ACE data in binary using the **MAKXSF** program which is distributed with MCNP.

2.2 How convert nuclear data into the appropriate form for my application ?

ENDF2ACE utility is an simple interface to the different step for generating ACE files for MCNP : build the NJOY input file, run NJOY, and run MAKXSF. The user can think that there is no physics in NJOY, but it is not the truth, there is physics (NJOY doc with it's 810 pages !)

2.2.1 NJOY

2.2.1.1 Introduction :

The NJOY nuclear data processing system is a modular computer code used for converting evaluated nuclear data in the ENDF format into libraries useful for applications calculations. Because ENDF is used all around the world (ENDF/B-VI and B-VII in the US, JEFF-3.1 in Europe, JENDL-3.3 in Japan, BROND-2.2 in Russia, CENDL-2 in China), NJOY gives access to a wide variety of the most up-to-date nuclear data. These libraries represent the underlying nuclear data from a physics point of view, but practical calculations usually require special libraries. NJOY provides comprehensive capabilities for processing evaluated data, and it can serve applications ranging from continuous energy Monte Carlo (like MCNP), through deterministic transport code, to reactor lattices codes. This is the mission of NJOY - to take the basic data from the nuclear data library and convert it into the forms needed for applications.

NJOY handles a wide variety of nuclear effects, including resonances, Doppler broadening, heating (KERMA: Kinetic Energy Release in MAterial), radiation damage, thermal scattering (even cold moderators), gas production, neutrons

NJOY	Directs the flow of data through the other modules and contains a library of common functions and subroutines used by the other modules
moder	Convert between ENDF standard coded mode and the NJOY blocked binary mode
reconr	Reconstruct point wise (energy-dependent) cross-sections from ENDF resonance parameters and interpolation schemes
broadr	Doppler-broadens and thins point-wise cross-sections
heatr	Generate point-wise heat production cross-sections (KERMA factors) and radiation damage production cross-sections
gaspr	Add gas production (mt203-207) to PENDF
unresr	Compute effective point-wise self-shielded cross-sections in the unresolved energy range
purr	Prepare unresolved region probability tables for the MCNP continuous energy Monte Carlo code
leapr	Calculates the thermal scattering law, $S(\alpha,\beta)$, in the incoherent and Gaussian approximations
thermr	Generate neutron scattering cross-sections and point-to-point scattering kernels in the thermal range for free or bound atoms
acer	Prepare libraries in ACE format for the Los Alamos continuous-energy Monte Carlo code MCNP
viewr	View plots generated by different modules in postscript

Table 2.2: List of used NJOY modules

and charged particles, photo-atomic interactions, self-shielding, probability tables, photon productions, and high-energy interactions (to 150 MeV). Output can include printed listings, special library files for applications, and Postscript graphics.

NJOY consists of a set of modules, each performing a well-defined processing task. Each of these modules is essentially a separate computer program linked together by input and output files and a few common constants.

All developments have been made using version NJOY99.90 (patch up90) using a Linux system.

2.2.1.2 NJOY Modules :

I give here just few comments about used NJOY modules used in ENDF2ACE.

2.2.1.3 Data Flow in NJOY

The modules of NJOY can be linked in a number of different ways to prepare libraries for various nuclear applications. We will briefly describe here only a basic input structure for create a ACE file for MCNP. Next, we find a typical input file ($^{235}_{92}\text{U}$ at 600 Kelvin) need for run NJOY and we will comment it.

```

moder
1 -21
*PENDF tape for ENDFB-6.8 92235(0)*

```



```

20 9228
0/
reconr
-21 -22
'PENDF tape for ENDFB-6.8 92235(0)'
9228 1 0 /
.001/
'92235(0) from ENDFB-6.8'/
0/
broadr
-21 -22 -23
9228 1/
.001/
600/
0/
heatr
-21 -23 -24/
9228 6 0 1 0 2/
302 303 304 402 443 444
gaspr
-21 -24 -25
thermr
0 -25 -42
0 9228 8 1 1 0 1 221 2
600
0.001 1.0
purr
-21 -42 -41
9228 1 7 20 16/
600
1.e10 1.e5 1.e4 1000. 100. 10. 1
0/
moder
-41 28
acer
-21 -41 0 26 27
1 0 1 .06/
'92235(0) from ENDFB-6.8 njoy99'/
9228 600/
/
/
acer
0 26 33 34 35
7 1 2 .06/
'92235(0) from ENDFB-6.8 njoy99'/
viewr
33 36/
stop

```

The word **tape** is used, by convention, for each module ; the input and output

file are named **tapeXX** (*tape20*, *tape21* ...).

All modules parameters are clearly described in the NJOY doc and summary ; here just few comments about the NJOY input used by ENDF2ACE are given.

- **moder** reads the ENDF file (*tape20*) which contains the material (for $^{235}_{92}\text{U}$ the material number is 9228), extracts it and put the ENDF file in binary mode (negative sign) in *tape21*. **moder** is usually used at the beginning of an NJOY run to convert ENDF library files into binary for calculation efficiency. It can also (it is the case here and in every instance) be used to extract desired materials from a multi-material library.
- **reconr** module reconstructs point-wise cross-sections: read *tape21*, extract material 9228 and reconstruct cross-sections at 0 Kelvin with a reconstruction tolerance of 0.001. The resulting point-wise cross-sections are written in a PENDF (for Point-ENDF) *tape22* for future use.
- **broadr** module reads *tape22* and Doppler-broadens at 600 Kelvin (only one temperature) the data using the accurate point-kernel method for material 9228. The union grid allows all resonance reactions to be broadened simultaneously, resulting in a great savings of processing time. After broadening and thinning (with a tolerance of 0.001), the summation cross-sections are again reconstructed from their parts. **broadr** needs in input the ENDF file (*tape21*) and the PENDF file (*tape22*). The results are written out in a new PENDF *tape23* for future use.
- **heatr** module compute energy-balance heating and damage energy for material 9228 for the 600 Kelvin temperature. Input files are *tape21* (ENDF file) and *tape23*. Line (302 303 304 402 443 444) are associated to the partial KERMA for reactions which must be added. Output file is *tape24* which is for the results for all the added processes.
- **gaspr** adds gas production reactions (mt203-207, see appendix G of MCNP doc[1]) to the PENDF tape. Any old gas sections on the input tape are deleted. Input tape : *tape21* is the ENDF file in binary mode, *tape24* is the PENDF tape ; Output tape is *tape25* (PENDF tape).
- **thermr** generates neutron scattering cross-sections and point-to-point scattering kernels in the thermal range. Input file is only *tape25*, which is the previous PENDF file, output file is *tape42*. The generation of neutron scattering cross-sections is done for material 9228 with 8 equi-probable angles at one temperature (600K), parameter for inelastic options is 1 (compute as free gas). Next parameter is 0 for the elastic options (no elastic cross-sections in the thermal range). In this case, we consider only 1 principal atom. The **mt** value for inelastic reaction is 221. A tolerance of 0.001 is chosen and the maximum energy for thermal treatment is 1 eV.
- **purr** calculate probability tables for treating unresolved-resonance self-shielding for Monte-Carlo codes (see NJOY documentation[3] page 548). Input : *tape21* and *tape42*, PENDF output : *tape41*. The calculation is done for material 9228 for one temperature (600K) with 7 sigma0 (1E10, 1E5, 1E4, 1000, 100, 10, 1). A study was done with MCNP for different

value of `sigma0`: choosing value 1 didn't give the correct results for the self-shielding effect, 1E10 is more representative but can be not adapted for all material ; only a table from 1 up to 1E10 gives systematically the good results. M. Mac Farlane from Los Alamos says that it is dependent of the considered material (i.e. the cross-section). One has chosen 20 probability bins and 16 resonance ladders. It can be noticed that the ENDF file must be contain some definition of probability tables for treating unresolved resonance, if it is not the case, NJOY don't crash but just write a warning message. Here is the answer of Mr Mac Farlane concerning the **purr** module and chosen parameters :

“The background cross-section is important for multigroup methods, but it is not needed for Monte Carlo. It can be convenient just to judge how important self shielding might be for a given material in a particular energy range. The purr module takes a lot of time to do its calculation, and that time is determined by `nladr`. If you can afford it, values on the order of 16 to 32 should be OK. Look at how closely the average over the ladders matches the infinitely dilute cross-section to get an idea if you are getting sufficient convergence. I haven't done many studies on the optimum size for `nbin-I` just use 20.

A problem here is with the occasional very small cross-section in interference minima. It is hard to get enough samples into the minima. As a result, the accuracy may begin to suffer for deep penetration, which is equivalent to very low sigma zero values. Examples might be very thick depleted uranium or thick iron with strong flux in the 200 keV range. Anyway, there is always doubt about such low cross-sections between resonances because they are hard to measure and because single-level Breit-Wigner theory doesn't represent them well.”

- The second call to **moder** converts binary PENDF file (*tape41*) in ASCII PENDF file (*tape28*)
- The first call to **acer** module prepares a data library for MCNP (in ASCII). Input files are *tape21* (ENDF file) and *tap41* (binary PENDF), output files are *tape26* (the ACE file) and *tape27* (the line will be copy in the XSDIR file of MCNP). These data are designed by fast data (value 1), with a minimal printed info (next value 0) in output file of **xnjoy**. The ACE output type for the file (*tape26* in ASCII) is defined by the value 1, with the *id* suffix for ZAID .06 (default in NJOY is .00). The material to process is 9228 at 600K.¹
- The second call to **acer** module reads *tape26* (ASCII ACE file) and prepare *tape33* (c.f. **viewr** module).

¹The next line with a “/” can be very important in some case. Some times one must replace the “/” by “0 1/” because NJOY crash with a error message “**too many angles**”. This is the case for nuclides with $Z \geq 90$ and $A \geq 233$. See more precisely **acer** section in NJOY doc[3] page 457-491.

- **viewr** module generate postscript file (*tape36*) from input file (*tape33*). postscript file of cross-section and other reaction can be very useful for verify quickly all the NJOY execution.

For point-wise cross-sections in the thermal range, **thermr** module is used. Energy-to-energy incoherent inelastic scattering matrices can be computed for free-gas scattering or for bound scattering using a precomputed scattering law $S(\alpha, \beta)$ in ENDF format. These last ENDF data can be produced with the **leapr** module (with a long list of parameters) and special parameters for **thermr** and **acer** (thermal data) modules which are material dependent (graphite, light water, heavy water, polyethylene, H in ZrH, Zr in ZrH). Comment about input file for thermal data are not given because it is specific to each material (see **leapr** doc [5, 6, 7]).

2.2.2 MAKXSF

MAKXSF program is fully described in the MCNP manual[1] (see appendix C section IV). MAKXSF is an auxiliary program distributed with MCNP to convert cross-section libraries from on format to another (ASCII ACE files to be converted in binary or vice versa). MCNP can read cross-section data from two types of files: the type 1 files (ACE ASCII files) are formatted and have sequential access, type 2 files are unformatted and have direct access. Binary files are often preferred because their reading (by MCNP) is faster and they have a smaller size on the disk.

MAKXSF need as input a ACE file (in our case, the final *tape26*), the XSDIR header with an adapted copy of directory line (*tape27*) and a *specs* file properly structured (c.f. In *endf2ace.cxx* source, the function *BuildSpecs* and see table C.5 of the MCNP manual).

Chapter 3

The ENDF2ACE utility

The basic idea of the **endf2ace** utility is to read a ENDF file and create the ACE file for a nuclide (a material) at a given temperature. It generates a input file for NJOY, run NJOY, convert the ASCII ACE output file in binary using MAKXSf, and finally generates a directory tree with needed files, for checks and libraries for MCNP.

3.1 Data Base in ACE format

The data base in the ENDF format, the characteristics of the nuclide (Z the proton number, A and I the isomer) and the wanted temperature are the minimum parameters to give to **endf2ace**. Other parameters are less important and have default values. Next chapter describes in more details input parameters ; here different steps are presented:

- With Z, A and I values, the *endf2ace* finds the material number in the *ZAI_mat.dat* file (*SetMaterialAttribute* method). User can also give only the material number, the method will find Z,A and I values in *ZAI_Mat.dat* file. For Thermal data, user can give only value associated to material (1 for Graphite, 2 for H in H₂O ...), method return directly Z,A,I and material number. The material number is crucial, it is the value that allows a nuclide to be identified in the ENDF file¹. We assume that this value is given by convention (see ref. [2]). If a material number is not found, the code stop with an appropriate error message (“*Material Number not found*”). Others options are possible (see sections 4.2.2 and 4.4).
- Different values are set (see chapter 4), like self-shielding calculation, suffix for the ACE file (default .00, see *SetSuffix* method), if a binary ACE file is desired (using MAKXSf, default is true), the path for output directory (default is local directory), NJOY executable location, MAKXSf executable location, name and version of the data base (default is ENDFB 6.8), the temperature (default is 300 K), the ENDF input file name and a

¹for ENDF/B, JEFF and JENDL data bases, all material number are the same for Z, A, I sets. It can differ for BROND and CENDL data bases, and in this case, it will be important to give to the code directly the material number (see next chapter for input parameters).

boolean for keeping specials output files from NJOY (postscript, PENDF file and NJOY output).

- In the next step, *BuildScript* method is called.

3.1.1 BuildScript method

The *BuildScript* method is the kernel code. the method does:

1. Print on standard output all parameters needed by the code : input file name for **xnjoy** (default is input), ENDF input file name location, name and version of the data base, the nuclide name (construct by Z, A and I), the material number, the temperature, the suffix, boolean for ACE binary file and unresolved resonance tables (**purr** module of NJOY is called if a “true” value is used), value for thermal Neutron scattering data (see section 3.2) and output path (from local directory).
2. Verify if user does not forget to give the file where material will be found
3. Call *SetPath* method which construct the complete and comprehensive path where output files will be copy. Two cases can be distinguished : ACE data for Continuous-Energy Tables (contain total, elastic, absorption cross-sections ...) called in a most simple way continuous data, and ACE data for Thermal $S(\alpha,\beta)$ tables (the thermal Neutron scattering data) called directly thermal data.
The path for continuous data is defined by : the output path (default is ./)+data base name+data base version+Z+A+I+Temperature. For example : ./ENDFB/6.8/92/235/0/300 for Uranium 235 at 300 Kelvin in ENDFB6.8 data base.
4. Check if *ace* and *dir* file already exist in the path construct previously. If yes, the code stops.
5. *BuildInput* method is called. The input file for NJOY is constructed using the different conditions (thermal data, self-shielding table, convert in ASCII the PEND file (*tape28*)). For more information about NJOY modules see section 2.2.1.2 and ref. [3]. The name of the table is define as convention in MCNP : ZZZAAA.nnC (see ref. [1] appendix F-II and table F-3), example : 92235.00C for $^{235}_{92}\text{U}$.
6. Next, a script file is built and is executed. The script performs different steps :
 - (a) link the ENDF input file to *tape20* which is the input tape for the first module **moder** call by NJOY.
 - (b) Run NJOY with the input file (*xnjoy < input*). All intermediate tapes are created in the directory where the program is executed (i.e. local directory), a maximum of 14 tapes are constructed.
 - (c) We verify if *tape26* and *tape27* exist in the directory. These files are respectively the ASCII ACE and directory line needed for MCNP *xmdir* file. The test is needed because NJOY can stop or crash for any reason and *endf2ace* don't stop. The best test is to check if final

needed ACE file is present in the directory. If it is not the case, remove all tapes (except input for NJOY, script and *xnjoy* output file)². If *tape26* and *tape27* exist, *endf2ace* does :

- i. create the output directory and two subdirectories named *asc* and *bin*,
 - ii. Move NJOY output file and *tape28* (the final PENDF file) in the *asc* subdirectory (if user has chosen this option),
 - iii. Modify *tape27* (containing the directory line for *xmdir* file of MCNP) in the correct version,
 - iv. If user want the ACE binary file, one needs to convert the ASCII ACE file using MAKXSf (see section 2.2.2 and 3.1.2),
 - v. ACE ASCII file (*tape26*), directory line for *xmdir* (*tape27*), input file for NJOY and the script file are moved in the final *asc* subdirectory,
 - vi. Postscript file generated with **viewr** module and output of NJOY are moved also (if user has chosen this option),
 - vii. all others intermediate files or not useful tapes are removed,
7. Finally, the right of the script file are modified and the script is execute.

3.1.2 Converting ASCII ACE in binary

As seen in section 2.2.2, MAKXSf is used to convert type 1 file (ASCII format) in type 2 (binary file). If user chooses to convert final ACE ASCII file (by default, the parameter is *true*), in the script file, some lines are added and others methods are called:

1. *BuildSpecs* method is called. For converting type 1 file in type 2, MAKXSf needs 2 files:
 - (a) The method called *BuildXSDIRHeader* which generates in the local directory a file named **xmdir1**. *xmdir1* contains the header of the classical *xmdir* file (section with “*atomic weight ratios*” and finish with the first line of the last *xmdir* section “*directory*”, see ref. [1] appendix F-II).
 - (b) The method creates a file named **specs**. Details structures of *specs* file are given in the ref. [1] appendix C-IV table C.5. Output files of MAKXSf are *tprint* (error messages), ACE type 2 file named *ace2* and *xmdir2* (with “*header*” and type 2 description data table). In the line “*ace ace2 2 2048 512*” in *specs* file, the value 2 is for type 2 libraries (binary), 2048 is the record length value (can be different from a machine to another, for example between 32 and 64 bits) and 512 is the number of entries per record.
2. Then, in the script file, the *tape27* (the directory line for MCNP *xmdir* in type 1 format) is added to *xmdir1* ,

²If NJOY crash, it will be useful to check the NJOY output file and the NJOY manual (ref. [3]) to identify troubles.

3. ACE output *tape26* is copy in a file called *ace* (file in ASCII type 1 format),
4. **makssf** is execute,
5. Binary directory in the directory tree is created,
6. *ace2* file is moved in the binary final directory with a new name *ace*,
7. name *ace2* is replace by *ace* in *xmdir2* before extracting the directory line from *xmdir2* and copying output file and moving that file,
8. remove others files not needed anymore from type 1 to type 2 ACE conversion.

3.2 The Thermal Neutron Scattering data

The procedure is almost the same than for continuous data in ACE format (see section 3.1). As input parameter, material must be specify by the user. In this case, a material is either Graphite, light Water, heavy Water, Beryllium Metal, Beryllium Oxyde, H in ZrH, Zr in ZrH, or Polyethylene ... 8 different materials are actually defined to construct ACE data :

1. Graphite
2. Water (H in H₂O)
3. H in ZrH
4. Zr in ZrH
5. Be (Beryllium Metal)
6. BeO (Beryllium Oxide)
7. Heavy Water (d in d₂O)
8. Polyethylene (H in CH₂)

We have seen that *SetMaterialAttribute* sets Z,A,I and material number directly for each value of material number given by user. For Carbon isotope (¹²C), user must precise Z=6, A=12 for Zirconium isotope (⁹⁰Zr,⁹¹Zr,⁹²Zr,⁹³Zr,⁹⁴Zr,⁹⁵Zr,⁹⁶Zr), user must give Z=40, A=90...96. If user do not give Z,A,I for thermal data of Carbon and Zirconium, natural material is taking.

With the appropriated parameter, the *BuildScript* method does:

1. Using *SetPath* method, the output path is defined for each material (example : *./THERMAL/CGPH* for Graphite). Note that the output path don't take into account the name and version of data base.
2. The structure of the script file is the same than for continuous data (see section 3.1.1).

3. In the same way, the *BuildInput* method is called. First part of input file is the same than for continuous data (**moder**, **reconr**, **broadr** modules are called). *AddThermalRangeModules* method is called which adds to input file for NJOY the appropriated next modules for thermal data (**leapr** which constructs the *mf7* part in ENDF format, **moder**, **thermr** and **acer** which construct the ACE file *tape26* and directory *tape27* as for continuous data). For example, for Graphite thermal data, *AddThermal-GRAPHITE* method is called. The last part of the input file for NJOY is added like for continuous data.
4. If type 2 ACE file is wanted, the process described in section 3.1.2 applies.
5. Remark : the name of the table (see directory output files *tape27*) is defined to avoid a mixing with default name in xsdir of MCNP in the form : *cgph.nnt*, *hh2o.nnt*, *hzrh.nnt*, *zrzrh.nnt*, *beth.nnt*, *beoth.nnt*, *dd2o.nnt* and *hch2.nnt* ...

Chapter 4

Using the `endf2ace`

In this chapter, some recommendations are given in order to use `endf2ace` in a correct way.

4.1 Input

We will discuss now what we need as input and what we must know about the input of the code.

1. Probably the most important thing is to have **ENDF file** for the chosen nuclide. A lot of different solutions exist for obtaining files in ENDF format from different data bases. The END2ACE utility will link on the file when you call it, the link name will be *tape20*.
 - (a) We can consider today that 5 major data bases exist :
 - i. the US library ENDF/B (actual version is VII),
 - ii. the Japanese JENDL library, version 3.2 and 3.3,
 - iii. the European library : JEF-2.2 and JEFF-3.1 version,
 - iv. the Russian library : BROND-2.2 (BROND-3 exist but not accessible)¹,
 - v. the Chinese library : CENDL-2².
 - (b) These libraries can be found on the web in a lot of official web sites. Principal sites wherein ENDF files can be found are given here:
 - i. <http://t2.lanl.gov/data/data.html> (the reference Los Alamos National Laboratory site),
 - ii. <http://www.nndc.bnl.gov/point2004/original/zaget.htm>,
http://www.nndc.bnl.gov/sg21/fp_cendl3.htm from National Nuclear Data Center of Brookhaven National laboratory,
 - iii. <http://iaeand.iaea.org/endl/endfframe.html>,
<http://www-nds.iaea.org/> from Nuclear Data Services of International Atomic Energy Agency,

¹you must probably verify the material number for your ZAI because material number in case of BROND library is not conventional (in sense, as ENDF/B, JEF and JENDL libraries)

²c.f. note 1

- iv. <http://www-nds.iaea.org/cd-catalog.html> for receive libraries on CD-ROM
 - v. <http://www.nea.fr/dbforms/data/eva/evatapes/> for JEFF3.1 library (and also others)
 - vi. <http://wwwndc.tokai.jaeri.go.jp/nuclldata/index.html> for all on JENDL libraries (JENDL3.2 and JENDL3.3)
 - vii. <http://ukrndc.kinr.kiev.ua/databases/Brond-2/>,
<http://ukrndc.kinr.kiev.ua/databases/Cendl-2/> ,
<http://ukrndc.kinr.kiev.ua/databases/databases.html> for Russian and Chinese libraries
 - viii. <http://www.ippe.obninsk.ru/podr/cjd/data/brond/brond.htm> for Russian library
 - ix. <http://www.google.fr> (or your preferred search engine)
2. You need also the **ZAI_Mat.dat** file which is provided with the code. The code will find the material number with Z, A and I value given by user. A second solution is to give directly the material number as parameter for the code. With the material value for thermal data, *SetMaterialAttribute* method will set Z,A,I and material number needed for running NJOY. The *ZAI_Mat.dat* file is constructed with the set of all material number find in US, European and Japanese libraries (ENDF/B-VI8, JEF3.0 and JENDL3.3). For Russian and Chinese (BROND and CENDL) libraries, please, verify the associated material number for Z,A and I, it can differ for these two last data bases.
 3. You must specify which nuclide you want : Z A I (examples : $^{235}_{92}\text{U}$ is Z=92 A=235 I=0, $^{58}_{27}\text{Co}^m$ is Z=27 A=58 I=1).
 4. You must also specify the temperature (in Kelvin).
 5. You can specify also :
 - (a) the name and version of the library,
 - (b) the material number (directly, do not use *ZAI_Mat.dat* file),
 - (c) generation of unresolved-resonance probability tables (using **purr** module of NJOY), the default is *true*,
 - (d) the suffix of the final data table, value from 00 up to 99 : example : 92235.10c,
 - (e) if you want convert ACE type 1 file in ACE binary type 2 file using MCNP auxiliary code MAKXSF. It can be useful, to check output datas after a NJOY run, to keep ASCII file (type 1). (default is *true* for binary),
 - (f) if you want keep the postscript file generated with **viewr** module of NJOY, the PEND file generated with last **moder** module of NJOY and output file *info* of NJOY run (default is *false* : files are removed)
 - (g) the path where you want put your NJOY results (default is local directory)
 - (h) the material for generating neutron scattering XS and point-to-point scattering kernels in the thermal range.
 6. Never forget: **you must know what you want !**

4.2 The package

You will find in the package: sources files of the code, documentation:

- **endf2ace.cxx** and **endf2ace.hxx** are the ENDF2ACE class source files
- **conv.cxx** is the main file for running the code
- **ZAI_Mat.dat** file gives you material number for each nuclide
- executable file **build_conv**, run *build_conv* will create the **conv** executable
- **ENDF2ACE_User_Guide.pdf** is this present guide.
- **Replace.cxx** is used to make string replacement ; because on some Unix/Linux system, the “*replace*” command is not present, we provide a small (even if not efficient) method to do the same job.

4.2.1 Installation

The package has been compiled with GCC 3.2 and after.

- Edit **conv.cxx** and modify the 2 string variables *Path2NJOYExec* and *Path2MaksfExec* (line 4 and 5) according to your installation.
- compile the code using the exec *build_conv*

```
> build_conv
```

Compilation options can be modified by editing *build_conv* file. Normal output are

```
> build_conv
Compiling Replace program
Compiling ENDF2ACE Class
Compiling conv program
```

4.2.2 The command line

After describing quickly the different parameters, some documented examples are given.

4.2.2.1 Beginners

To have the list of all possible parameters, type

```
> conv -h
=====
You should use the following options :
-p path and name of the endf file, e.g: /My_endf_library/endf_file [NEEDED]
-d data base name, e.g. endfb, jeff, jendl, brond, cendl [OPTIONAL, default endfb]
-v data base version, e.g., 6.8, 3.0, 3.3, 2.2 [OPTIONAL, default 6.0]
-z ZZ, e.g., 90 [NEEDED] if -m is not used
-a AAA, e.g., 232 [NEEDED] if -m is not used
-i I, the isomeric state, e.g., 0 for ground state [OPTIONAL, default=0]
-m num, material number in endf, e.g., 7925 for 197Au [OPTIONAL, (see -z and -a)]
```

```

-t T, the wanted temperature (in kelvin), [OPTIONAL, default=300]
-u num, self-shielded cross-section in the URR [OPTIONAL, default=1 (true)]
-s XX, the suffixe for material in acer like ZZAAA.00c [OPTIONAL, default=00]
-b num, binary ace translation from ascii ace [OPTIONAL, default=1 (true)]
-o path, path of the output directory [OPTIONAL, default=./]
-k num, keep or not postscript, njoy_output and pend_file from NJOY [OPTIONAL, default=0 (false)]
-n num, generate neutron scattering in the thermal range for material num, call leapr and thermr modules
  Thermal data available for :
    0 - do not generate thermal data
    1 - GRAPHITE
    2 - WATER (H in H2O)
    3 - H in ZrH
    4 - Zr in ZrH
    5 - Be (Beryllium Metal)
    6 - BeO (Beryllium Oxide)
    7 - HEAVY WATER (D in D2O)
    8 - Polyethylene (H in CH2)
=====

```

4.2.2.2 Examples

example 1 : Generate the continuous data for $^{235}_{92}\text{U}$. The ENDF file (named *myfile.endf*) exists in local directory. Defaults are chosen:

```
> conv -p myfile.endf -z 92 -a 235
```

By default, the data base name and version is ENDFB 6.0, value for the isomeric state is 0, i.e., ground state (there are few cases with a true isomeric state, see *ZAI_Mat.dat* file), temperature is 300K, suffix is 00 (i.e. the final MCNP code will be 92235.00c), self-shielding cross-sections are calculated, type 1 ASCII ACE file are converted in type 2 (binary), output path is local and no evaluation cross-section in the thermal range is performed.

example 2 : Generate the continuous data for $^{235}_{92}\text{U}$ at 400 Kelvin :

```
> conv -p myfile.endf -z 92 -a 235 -t 400
```

example 3 : Generate the continuous data for $^{235}_{92}\text{U}$ at 400 Kelvin, the suffix for the name of the data table (in xsdir file) is 04 (to produce 92235.04c) :

```
> conv -p myfile.endf -z 92 -a 235 -t 400 -s 04
```

Differents possibilities for suffix are: "04" , ".04" or "4".

example 4 : Generate the continuous data for $^{235}_{92}\text{U}$, at 400 Kelvin without self-shielding cross-section in the unresolved resonance region :

```
> conv -p myfile.endf -z 92 -a 235 -t 400 -s 04 -u 0
```

"-u 0" specifies that input file for NJOY don't add **purr** module (**purr** is used to generate unresolved-resonance probability tables for MCNP). In the other hand, if you want specify absolutely the boolean to generate unresolved-resonance probability tables, you just need to do "-u 1".

example 5 : Generate the continuous data for $^{235}_{92}\text{U}$, without self-shielding cross-section in the unresolved resonance region and only ASCII ACE file is wanted:

```
> conv -p myfile.endf -z 92 -a 235 -t 400 -s 04 -u 0 -b 0
```

“-b 0” specifies that no type 2 binary ACE file must be generate. In that case, MAKXSf is not called. By default, binary file is generated (using MAKXSf).

example 6 : Generate the continuous data for $^{235}_{92}\text{U}$, putting results in a specific output path (see item 3 of section 3.1.1):

```
> conv -p myfile.endf -z 92 -a 235 -t 400 -s 04 -o /path
```

“-o /path” specify the path where results must be copied. In this example, final files from NJOY and MAKXSf will be put in :
/path/ENDFB/6.0/92/235/0/400/asc/ and
/path/ENDFB/6.0/92/235/0/400/bin/

example 7 : Generate the continuous data for $^{235}_{92}\text{U}$, but specifying directly the material number (assuming that the *ZAI_Mat.dat* file is present in the local directory):

```
> conv -p myfile.endf -m 9228
```

Code will find in *ZAI_Mat.dat* file the Z, A and I for $^{235}_{92}\text{U}$. If you type wrong numbers like:

```
> conv -p myfile.endf -m 600 -z 92 -a 235
```

Code will run normally but *ZAI_Mat.dat* file is read, taking [material number found int the file ZAI_Mat.dat](#) (here 9228) and not the one provided by the user (here 600).

example 8 : If *ZAI_Mat.dat* file does not exist, one must give the material number, Z, A and I:

```
> conv -p myfile.endf -m 9228 -z 92 -a 235 -i 0
```

Code use only the material number, Z and A value are used for the output path construction. Giving a wrong material number:

```
> conv -p myfile.endf -m 600 -z 92 -a 235 -i 0
```

The code will run assuming evry thing is OK....This could be useful to specify different material number (if it is not standard).

example 9 : Generate the continuous data for $^{235}_{92}\text{U}$ and saying that *myfile.endf* is taken from JENDL 3.3 data base:

```
> conv -p myfile.endf -z 92 -a 235 -d JENDL -v 3.3
```

These two parameters will be used principally for the output path. In this case, final files will be put in : `./JENDL/3.3/92/235/0/300` (default temperature, 300K, is used and as well as local directory).

example 10 : To keep final postscript file of cross-sections generated with **viewr** module of NJOY, PEND file generated with last **moder** module of NJOY and output file information of the NJOY run:

```
> conv -p myfile.endf -z 92 -a 235 -d JENDL -v 3.3 -k
```

example 11 : These next examples show how to generate thermal neutron scattering data (see section 4.2.2.1) :

- 1 Graphite
- 2 Water (H in H₂O)
- 3 H in ZrH
- 4 Zr in ZrH
- 5 Be (Beryllium Metal)
- 6 BeO (Beryllium Oxide)
- 7 Heavy Water (d in d₂O)
- 8 Polyethykene (H in CH₂)

To construct thermal data for Graphite at 400 Kelvin, suffix will be 04 and ENDF file will be *myfile.endf* (we suppose that file for Carbon is present in *myfile.endf*), *ZAI_Mat.dat* exists :

```
> conv -p myfile.endf -t 400 -s 04 -n 1
```

For Graphite, option is “-n 1”. *SetMaterialAttribute* associate Graphite to Z=6 and A=0 which is the natural Carbon. For Carbon and Zirconium, user can give Z,A for isotope. As an example:

```
> conv -p myfile.endf -t 400 -s 04 -n 1 -z 6 -a 12
```

In case of thermal scattering data, generate unresolved-resonance probability table using **purr** module of NJOY have no sense, but you don't need to precise the option “-u 0”. For generate thermal scattering data, code needs the corresponding ENDF file, cross-sections are reconstructed, but final ACE file is only dependent from the file generate by **leapr** module and the effect of the data base chosen at the beginnig is negligible. That's why there is no need to precise the name and version of data base.

4.3 Standard message

We will comment now some elements write on the standard output during a run. As an example, we choose our *myfile.endf* for the ENDF/B-6.8 data base. We want continuous data for ²³⁵U at 600 Kelvin, with suffix 06, probability tables in unresolved resonance region (default parameter), type 2 binary ACE file using MAKXSF and output path is local directory:

```
> conv -p myfile.endf -d ENDFB -v 6.8 -z 92 -a 235 -t 600 -s 06
```

In chapter 3, we have studied how to the code works for create ACE (type 1 and type 2) files from the ENDF file, here, the different steps are supposed known. We will just comment for user standard messages written on standard output :

1- Material number identification from ZAI_Mat.dat (from SetMaterialNumber method)

```
=====
Material file ZAI_Mat.dat found
Success !      Z=92 A=235 I=0 of U ,
                Material number is : 9228
```

2- Summarization of parameters choosen by user :

```
-----
The input file name for xnjoy program is : input
The path name+file is : myfile.endf
The data base + version is : ENDFB 6.8
The nucleide is : 92235(0)
The material number is : 9228
The temperature is : 600
The suffix is : .06
Binary file wanted : 1
SelfShielding : 1
ThermalRange : 0
Keep postscript, PENDF, NJOY output : 0
Output path : ./
-----
```

It will be sometimes useful to check these values. For example, some NJOY run are long (${}_{92}^{238}\text{U}$ or ${}_{90}^{232}\text{Th}$ for example), thus don't loose your time with wrong parameters.

3- Prepare NJOY run :

```
-----
Continuous-energy Neutron transport table for Material
Number 9228 (nucleide 92235)
-----
Check if nuclide already done : No, continue ...
-----
NJOY endf to ace format
getting endf tape for 92235(0) from myfile.endf
```

Output path is created, code checks if nuclide is already done. This last line is just a *tape20* link on *myfile.endf* (*tapeXX* is NJOY file transfert).

4- The NJOY run (all outputs write here are output from NJOY) :

```
njoy 99.90                                09/20/04 14:48:30
*****
moder...                                  0.0s
processing endf or PENDF tape.
reconr...                                  8.3s
broadr...                                  451.7s
heatr...                                   549.0s
```



```

---message from hinit---mt19 has no spectrum
                        mt18 spectrum will be used.
---message from hinit---mf6, mt 16 does not give recoil za=92234
                        one-particle recoil approx. used.
---message from hinit---mf6, mt 17 does not give recoil za=92233
                        one-particle recoil approx. used.
---message from hinit---mf6, mt 37 does not give recoil za=92232
                        one-particle recoil approx. used.
---message from hinit---mf6, mt 91 does not give recoil za=92235
                        one-particle recoil approx. used.
---message from nheat---changed q from 1.9372E+08 to 1.8089E+08
                        for mt 18

gaspr...                571.7s
thermr...              573.7s
wrote thermal data for temp= 6.000E+02      581.7s
purr...                581.7s
mat = 9228             581.7s
acer...                774.8s
                        783.3s
*****

```

As you see, we have used for this example NJOY version 99.90. Run has started the 20th of septembre 2004 at arround 14h48. Standard output of NJOY are the list of modules used in the input file, the right column is the running time in second. Some specific message can be given by a module. If NJOY does an error, the run stops with a message and nothing is done by ENDF2ACE.

5- When type 1 ASCII ACE is converted in type 2 binary file, one get:

```

tape27 converted
tape27 converted
tape27 converted
For MAKXSf, you need specs and xsdir1 files
Starting makxsf....
makxsf done.
xsdir2 converted

```

6- If everything have worked fine, you will obtain this final lines :

```

-----
ASCII file for MCNP :
ace dir input njoy.csh
xsdir line to be added in local xsdir :
92235.06c 233.025000 ace 0 1 1 621851 0 0 5.170E-08 ptable
Replace "ace" by the complete path and name of the ace file
WARNING: line in the xsdir file are limited to 80 columns, the continuous mark is +
Same method for binary files
You will find files in ./ENDFB/6.8/92/235/0/600
-----

```

Second line is the list of files moved in the ASCII directory :
./ENDFB/6.8/92/235/0/600/asc/ (c.f. item 6c of section 3.1.1). njoy.csh is the script file used for generate ACE files. input is the input file for NJOY. ace file is the type 1 ASCII ACE file (this file is readable by MCNP) and dir file contains line which must be copied (and mod-

ified to take into account the full path) in your *xmdir*³. The directory `./ENDFB/6.8/92/235/0/600/bin/` contains respectively type 2 binary ACE file (*ace*) and associated directory line for MCNP (*dir*).

4.4 Ouput

When the run is finished with success (we use “-k” option in **conv**), output files are written in output directory (c.f. last description of section 4.3) :

```
> ls -lh ./ENDFB/6.8/92/235/0/600*/
./ENDFB/6.8/92/235/0/600/asc/:
total 13M
-rw-r--r--  1 perrot  pacs          12M Sep 20 15:02 ace
-rw-r--r--  1 perrot  pacs           66 Sep 20 15:02 dir
-rw-rw-r--  1 perrot  pacs          535 Sep 20 14:48 input
-rwxrwxr-x  1 perrot  pacs          2.5k Sep 20 14:48 njoy.csh
./ENDFB/6.8/92/235/0/600/bin/:
total 2.4M
-rw-r--r--  1 perrot  pacs          2.4M Sep 20 15:02 ace
-rw-r--r--  1 perrot  pacs           80 Sep 20 15:03 dir
```

Four files are found in the *asc* directory and two in *bin* directory.

To give an order of disk space for a complete data base, considering 100 nuclides with 10 temperatures (taking a 12 Mb per nuclide on average), you need a minimum total disk space arround 12 Gb. Perhaps, it’s a lot, but we are at the beginning of the 21th century and we can buy large disk for low cost...

4.5 Error messages

Lots of error message can appear, here major typicals errors due to user (principally) and few others are given:

- SWrong path to NJOY or MAKXSf executable:


```
##### Error, NJOY executable /yourpathNJOY/xnjoy does not found (The code
will stop immediately)
##### Error, MAKXSf executable /yourpathMAKXSf/makxsf does not found (No
type 2 binary ACE file created)
```
- You can also forget to give the ENDF file (option -p), you will get :


```
##### Error, You have probably forget to give the path+name of you ENDF file
(see doc for -p option). Stop
```
- You give a wrong ENDF file. Everything will start properly but NJOY will abort during the first module **moder**. In fact, **moder** module can’t extract ENDF file from the given file:

³For special user who want keep : postscript 92235.ps file is a useful postscript file, PENDF file is the last file from NJOY which contain everything (to be used by expert but not really important). File `njoy_output` can be very important : it contains details from the NJOY run. Use option -k.

```

njoy 99.90                                09/21/04 13:48:50
*****
moder...                                  0.0s

processing endf or PENDF tape.
open: No such file or directory
apparent state: unit 20 named tape20
lately writing sequential formatted external IO
Abort
-----
Problem occurs during the xnjoy execution, no directory structure and ace file created
Please look into njoy's output file to solve the problem
-----

```

OR

```

njoy 99.90                                09/21/04 13:52:23
*****
moder...                                  0.0s

processing endf or PENDF tape.
fmt: end of file
apparent state: unit 20 named tape20
last format: (16a4,a2,i4,i2,i3,i5)
lately reading sequential formatted external IO
Abort
-----
Problem occurs during the xnjoy execution, no directory structure and ace file created
Please look into njoy's output file to solve the problem
-----

```

- The file given is a true ENDF file but the nuclide chosen doesn't not found by NJOY (**moder** module). NJOY abort and you will get :

```

njoy 99.90                                09/21/04 14:08:05
*****
moder...                                  0.0s
processing endf or PENDF tape.
***error in findf*** mat2525 mf 0 mt 0 not on tape 20

STOP 77 statement executed
-----
Problem occurs during the xnjoy execution, no directory structure and ace file created
Please look into njoy's output file to solve the problem
-----

```

- It can be append also for reason not yet determine error during acer module execution :

```

acer...                                    795.6s
***error in ptleg2***too many angles
STOP 77 statement executed

```

It can be solved by uncomment and comment lines in *BuildInput* method where you will find “****error in ptleg2***too many angles*”. Then **you need recompile the code**. This part concern just a NJOY method for define the coupled energy-angle distributions (see **acer** section page 457-491 in ref. [3] and specially card 6 in **acer** input specifications).

- In all case, when you have this error message :

```

-----
Problem occurs during the xnjoy execution, no directory structure and ace file created

```

Please look into njoy's output file to solve the problem

It will be a very good idea to check your command line, the input file for NJOY, the script file and "output" file from NJOY (files not removed) for localise the error.

- You want to create output directory somewhere you can't (case of use option "-o"). You will get probably a message of that kind :

```
mkdir: cannot create directory '/BLABLA': Permission denied
```

- If user choose a nuclide at a given temperature who already exist in the considering output path, you will get this kind of message :
Check if nucleide already done : Yes, not needed to run another time, stop !

```
./ENDFB/6.8/92/235/0/600/asc/ace  
./ENDFB/6.8/92/235/0/600/asc/dir  
./ENDFB/6.8/92/235/0/600/asc/input  
./ENDFB/6.8/92/235/0/600/asc/njoy.csh
```

The test is done on *ace* and *dir* files in output ASCII directory are listing and code stop.

- Use of Z, A, I or material number parameters associated with *ZAI_Mat.dat* file have been commented in section 4.2.2.2. You will find next typical error message from these parameters :

1. You forget to give Z or/and A or you give Z and A but wrong I, *ZAI_Mat.dat* exists (without material number value):

```
Values given are : Material number is -1 , Z=0 , A=55 , I=0. It is not  
sufficient !
```

```
Nothing for work ! Give me something ! Stop
```

2. You give Z and A but wrong material number, the code run but give this warning message, *ZAI_Mat.dat* exists :

```
Material file ZAI_Mat.dat found  
Warning, Material number found (2525) in ZAI_Mat.dat and number given (252)  
are different
```

```
Material number choosen is Material number found : 2525
```

```
Success !      Z=25 A=55 I=0 of MN ,
```

```
Material number is : 2525
```

3. *ZAI_Mat.dat* is not present, you give Z, A and I but not the material number

```
Material file ZAI_Mat.dat not found. Stop
```

4. *ZAI_Mat.dat* is not present, you give Z, A and I but a wrong material number. There are 2 cases:

- you give a material number which not exist in the ENDF file, **moder** modules stops the execution.
- you give a material number which exist in the ENDF file, you have clearly a problem, but nothing stops. NJOY will work with the material number and the structure of the output path and name of the final table will be wrong. Message will be :

```
Warning, working with Z=25 A=55 I=0 and Material number 9040
given by user
Values can be not compatible
Success !           Z=25 A=55 I=0 of ZZ ,
                    Material number is : 9040
```

5. *ZAI_Mat.dat* is not present, you don't give Z, A and I but you give the material number :

I need more, please give me also Z,A and I (minimum is Z,A). Stop

- In case of thermal data, if you give a wrong value in the command line :

```
myprompt> conv -p ../myfile.endf -d ENDFB -n 9
=====
You ask for a value for thermal data which is not implemented, verify carefully
your parameters
Thermal data available for :
  1 - GRAPHITE
  2 - WATER (H in H2O)
  3 - H in ZrH
  4 - Zr in ZrH
  5 - Be (Beryllium Metal)
  6 - BeO (Beryllium Oxide)
  7 - HEAVY WATER (d in d2O)
  8 - Polyethylene (H in CH2)
=====
```

Bibliography

- [1] J.F. Briesmeister. MCNP - A General Monte Carlo N-Particle Transport Code, Version 4C, 2000. LA-13709-M
- [2] Cross Section Evaluation Working Group. ENDF-102 Data Formats and Procedures for the Evaluated Nuclear Data File ENDF-6. Informal Report. Revised version April 2001. BNL-NCS-44945-01/04-Rev (see also : <http://www.nndc.bnl.gov/nndescr/documents/endl>).
- [3] R.E. Mac Farlane. RSICC Peripheral Shielding Routine Collection : NJOY99.0, Code System for Producing Pointwise and Multigroup Neutron and Photon Cross Sections from ENDF/B Data. Oak Ridge National Laboratory. PSR-480 NJOY99.0
- [4] R.E. Mac Farlane. User Input for NJOY. It is a very useful ASCII file place in the NJOY directory, the file is named Userinp. It contains a basic description of each module used by the NJOY code (see also : <http://t2.lanl.gov/codes/njoy99/Userinp>).
- [5] R.E. Mac Farlane. New Thermal Neutron Scattering Files for ENDF/B-VI release 2. <http://t2.lanl.gov/data/thermal.html> , <http://t2.lanl.gov/publications/thermal/thermal.ps> . August 24, 1994.
- [6] J. Keinert, M. Mattes IKE Stuttgart. JEF-1 Scattering Law Data. JEF Report 2, JEF/DOC 41.2, IKE 6-147. September 1984. http://www.nea.fr/html/dbdata/nds_jefreports/jefreport-2.pdf , http://www.nea.fr/html/dbdata/nds_jefreports/nea3249-jef-report-cd.html
- [7] J.U. Koppel and D.H. Houston. Reference Manual for ENDF Thermal Neutron Scattering Data. National Nuclear Data Center, GA-8774 Revised (ENDF - 269). July 1978. On request to services@bnl.gov
- [8] F. Frohner. Evaluation and Analysis of Nuclear Resonance Data, JEFF Report 18. NEA-OCDE 2000. http://www.nea.fr/html/dbdata/nds_jefreports/jefreport-18/jeff18.pdf , exist in French : http://www.nea.fr/html/dbdata/nds_jefreports/jefreport-18/jeff18fr.pdf .