

Nuclear Engineering Laboratory
M O N T E C U C C O L I N O
technical report

Nuclear Reactor Physics
Department of Energy and Nuclear Engineering
and of Environmental Control (DIENCA)
University of Bologna

Doc. ID: LIN-R04.2008
Subject: NJoyInputMaker v3
Date: April 30, 2008

Author: Giacomo Grasso
Phone: +39-051-6441708
e-mail: giacomo.grasso@mail.ing.unibo.it

NJoyInputMaker version 3: a graphical tool for automatically processing cross section libraries with NJoy

The present work describes the evolution of the original script *njoyinputmaker.sh* for automatic processing of cross section libraries. A new graphical, menu-driven interface has been introduced for a more user-friendly approach to the creation of cross sections datasets for nuclear systems analysis tools, like MCNP, by means of the well-known NJoy nuclear data processing code.

The Author
Giacomo Grasso

The Supervisor
prof. Marco Sumini

Revisions List:

Date	Author

NJoyInputMaker version 3: a graphical tool for automatically processing cross section libraries with NJoy

Giacomo Grasso

April 30, 2008

Abstract

The present work describes the evolution of the original script *njoyinputmaker.sh* for automatic processing of cross section libraries. A new graphical, menu-driven interface has been introduced for a more user-friendly approach to the creation of cross sections datasets for nuclear systems analysis tools, like MCNP, by means of the well-known NJoy nuclear data processing code.

Contents

1	Introduction	3
2	Script features	4
2.1	Execution parameters	4
2.2	Error handling	5
3	Usage of the new graphical interface	6
A	njoyinputmaker.sh source	10
B	Help menu	35
C	Sample input file	36
D	Error list	37

1 Introduction

The cross section libraries are the most important dependencies for the simulation of a nuclear system: they retrieve all the informations about the behavior of neutrons within the materials composing the system for the criticality study of the latter.

The base format of cross section libraries is the tape one: point data are collected in energy dependent tables referring to 0 K. In order to use libraries within nuclear codes,

they have to be converted in a compatible format and processed to the system's working temperatures. The NJoy [1] nuclear data processing tool has therefore been developed in order to achieve those goals: to both convert the cross section tapes into suitable libraries and to modify the cross sections behaviors depending on the reference temperature.

A graphical, menu-driven automatic script has been developed, for the sequential processing of a list of isotopes at the same temperature: the desired libraries pointed out in a simple input file are individually retrieved together with the corresponding MAT number [2], then an appropriated input script [3] is compiled for running the NJoy code; finally the processed library file stored in a folder and the corresponding entry updated into the proper xsdir library database (the script has been specifically developed for ENDF [4] library output format for use in the MCNP [5] nuclear code).

The Graphical User Interface (GUI) introduced extends the previous version of the script [6] by letting the user tune all the parameters needed for the NJoy execution intuitively: several dialogs (menu, input-box, file-select or edit) guide the user among the options easing the previous command line mode. Several checks have moreover been added in order to verify the correctness of the parameters values before running NJoy.

2 Script features

The “njoyinputmaker.sh” script (whose source code is fully presented in appendix A) is a bash script developed by the author for as general as possible applicability to nuclear data processing. The dialog based GUI is provided by either the **dialog** or **whiptail** packages, and drives the user through the configuration of all the options needed to correctly run NJoy for cross section libraries processing. Wether neither package is available on the host system, the script is run in command line mode as in the previous versions of the script. A typical help message summarizes both the mandatory and optional script parameters, extended in the actual version of the script; it can be retrieved by typing:

```
./njoyinputmaker.sh -h
```

or equivalently

```
./njoyinputmaker.sh --help
```

The output capture of the help message is presented in appendix B.

2.1 Execution parameters

When run in command line mode, three mandatory arguments are to be specified: the file which lists the isotopes IDs to be processed, the temperature of interest, expressed in Kelvin, and the output library file. The input file must be compiled as an ASCII text file, with isotopes IDs provided in column (a sample input file is presented in appendix C). No

mandatory argument is needed in the dialog guided mode, as it is explained in the next section.

In addition to the mandatory arguments, a list of optional variables can be set, in order to tune the script to correctly locate all the needed dependencies as well as to specify the output form.

Among the optional parameters to be set, there is a number of fundamental variables the user is allowed to specify: the default values of such variables have been chosen assuming the script is placed in a folder containing all is needed to process cross section libraries. It is fundamental therefore to edit such variables to reflect the user's environment configuration:

- the NJoy executable location (default `./`); before the script execution, the `xnjoy` program is checked to retrieve the exact version to comment the processed libraries with;
- the path to the input library tapes folder (the default is `./library`).

Some other optional parameters are directed to define both the output library and `xmdir` files locations, as well as the new libraries suffix to be used as a reference in MCNP. Within the output folder (located by default in `./processed_library`) the three new subfolders `out`, `libs` and `xsdirs` are created. The NJoy code produces indeed, for each nuclide to be processed, an output summary, the result library and its corresponding `xmdir` entry as separate files. The “`njoyinputmaker.sh`” script therefore collects all the output files in the `out` subfolder, the produced libraries in the `libs` subfolder and the `xmdir` entry files in the remaining `xsdirs` subfolder.

The name of the treated libraries, which is used to reference the desired library file within MCNP through the `xmdir` database, is determined by the `zaid` number of the respective nuclide, appended of a suffix which univocally identifies the belonging library. The user can therefore easily modify the suffix (default value: `.60`) to avoid duplicate entries consistently with the installed cross sections database.

At the end of the cross section tapes processing, all the new treated libraries are collected in the final comprehensive library (which is a mandatory argument) as well as the `xmdir` entries, modified to point the just produced library, are collected in a global `xmdir` file (which default name, `xmdir`, can be properly edited by the user).

An optional flag can be toggled to force overwriting of eventually existent files and folders.

The remaining optional parameters collect informations to customize the format of the output library. It is possible indeed to specify the input library name and version (the default values, `JEF` and `2.2`, have been chosen depending on the author default choice) and the name of the user (the default value is set by retrieving the output of the `whoami` shell command) to mark the new produced library with its author's name.

2.2 Error handling

The execution of the script with bad parameters values may cause the stop of the script and the possible overwriting of existing files. In the script it is present therefore a deep testing section where all the input parameters (entered either via command line or graphically) are checked before launching the automatic cross sections processing.

Wherever a bad parameter is detected, the script reports an error message to the log file and exits without accessing the files creation/editing phase. The main errors which can be reported by the code, and the causes which determined such errors, are here presented:

- 2 the script was unable to find the specified processing nuclides list input file;
- 3 the script was unable to find the NJoy executable in the specified path;
- 4 the script was unable to find all the cross section libraries specified in the input file: this could be due to missing libraries in the specified path or to an error in the nuclides list file;
- 8 the script was unable to find the input cross section libraries folder.

A full listing of the error messages is presented in appendix [D](#).

3 Usage of the new graphical interface

The GUI guides the user through a series of dialogs and choices to set up the whole environment, from the same input file and temperature strictly needed to run the processing, to the defining of the paths to the NJoy executable, the input library tapes and the output library file.

The dialogs are organized to separate the main variables setting from that of the optional parameters. The first menu driven dialog (whose capture is depicted in Figure 1) allows the user to access the child dialogs where it is possible to set the main environment variables.

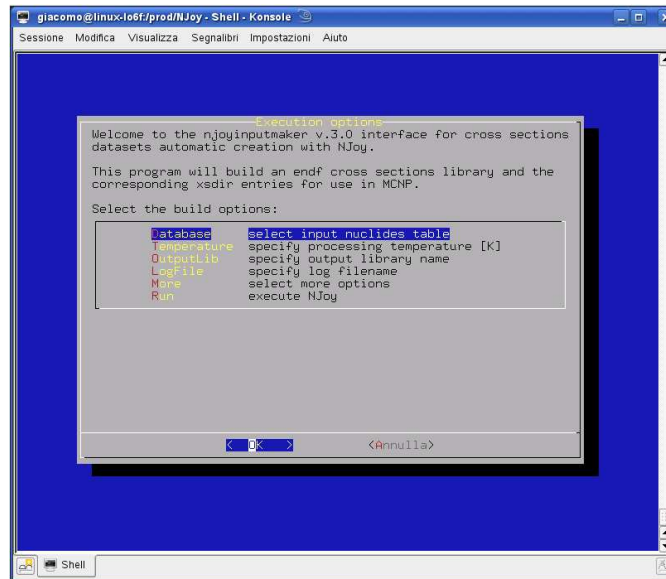


Figure 1: Snapshot of the main dialog.

The first child (corresponding to the **Database** entry in the main dialog) is a file-select dialog, which allows the user to navigate the filesystem looking for the input file listing the processing libraries. The dialog window, captured in Figure 2, is divided in three portions: an editable navigation bar in the bottom part and a folders and a files list selectors in the top part. The user can navigate the filesystem by choosing a folder in the folders view by means of the arrow buttons, and then selecting it with the space bar. The choice is then confirmed by pressing the enter button. The working directory is updated at every selection - and, consequently, the folders and files view are refreshed to show the actual position content - until a file is selected. Alternatively, it is possible to specify manually the input file by typing the whole url to the file itself.

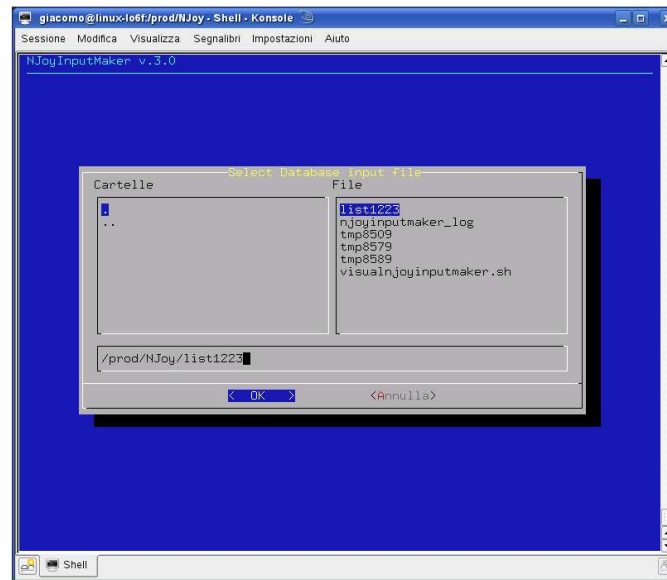


Figure 2: Snapshot of the input list file-select dialog.

The second child (the **Temperature** entry one) is an input-box dialog (Figure 3): the user is simply asked to type the temperature (in Kelvin) to process the cross section libraries at.

Similar dialogs are invoked by the **OutputLib** and **LogFile** entries of the main menu to specify the name of the output library file and of the script log file respectively.

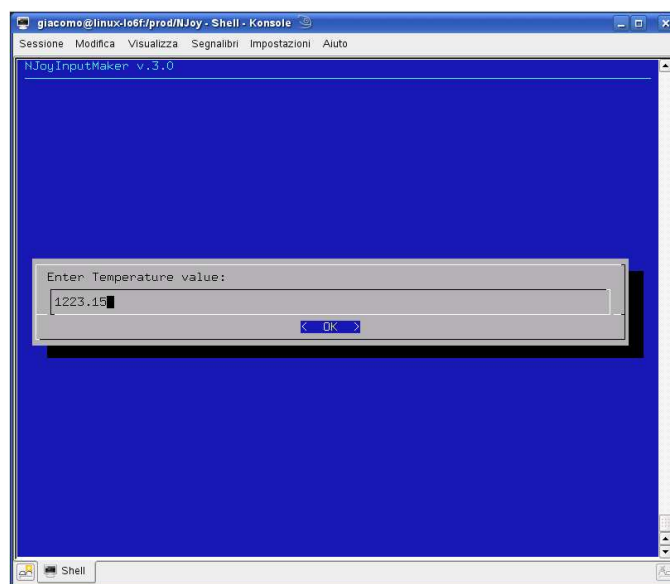


Figure 3: Snapshot of the temperature input-box dialog.

The **More** entry of the main dialog opens the last child dialog, the edit dialog depicted in Figure 4. The optional variables, together with their actual value, are listed. Every variable can be edited by selecting it with the up-down arrows, moving left/right on the edit button and then pressing the enter key. When every field has been tuned, it is possible to accept the whole changes by selecting the ok button.

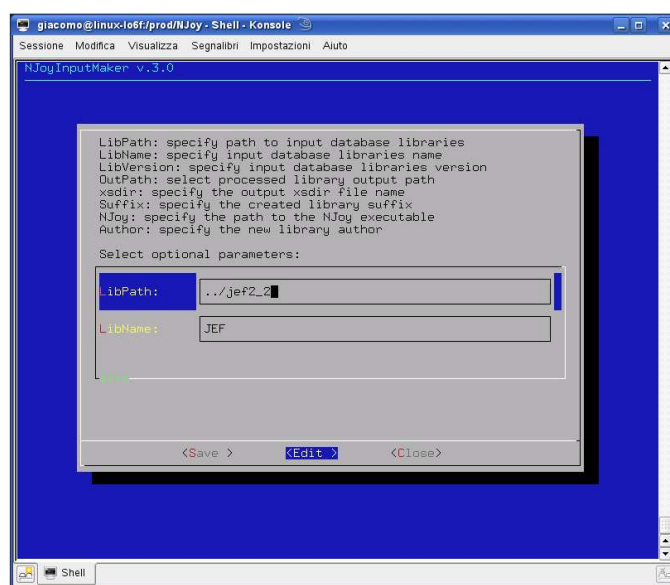


Figure 4: Snapshot of the optional parameters edit dialog.

Finally when the **Run** option is chosen in the Main menu, before executing the script kernel (for the processing of the cross sections libraries), a summary dialog (Figure 5) is proposed to the user, which is asked to verify the correctness of the choices operated.

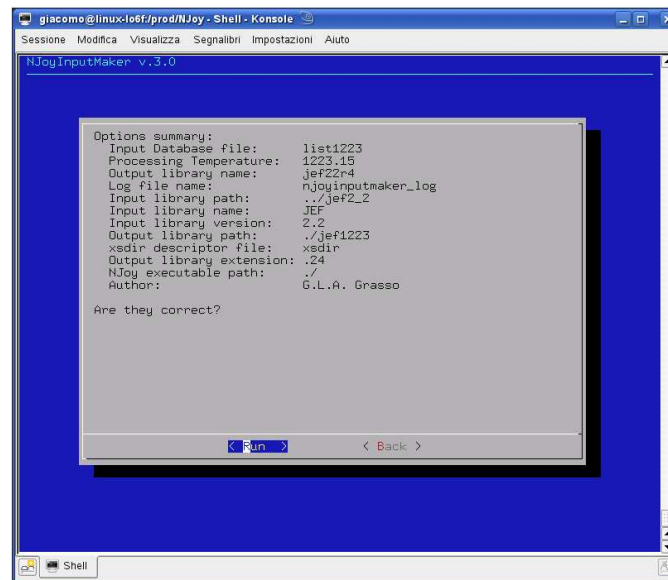


Figure 5: Snapshot of the options summary dialog.

A njoyinputmaker.sh source

```
#!/bin/bash
# *****
#   Script: njoyinputmaker.sh
#
#   Purpose:
#       A script that supplies a graphical intuitive, menu-driven interface
#       to the existing njoyinputmaker.sh program, to simplify the automatic
#       interaction of the core script with NJoy in creating cross sections
#       datasets.
#
#   Nuclear Reactor Physics group
#   Nuclear Engineering Laboratory (LIN) of MONTECUCCOLINO
#   Department of Energy and Nuclear Engineering and of
#   Environmental Control (DIENCA)
#   University of Bologna
#
#   Nuclear Physics (FISNUC) group
#   Research Center "E. Clementel" - BO MMS
#   Department of Fusion, technologies and Nuclear Aid (FPN)
#   National Agency for the New Technologies,
#   Energy and Environment (ENEA)
#
#   Author:      Giacomo L.A. Grasso (with large portions borrowed from
#               build_visit, maintained by Hank Childs and
#               BUILD_NOTES, maintained by Eric Brugger)
#   Creation:    April 10, 2008
#   Version:     3.0
#   Copyright:   LIN   Giacomo Grasso 2008
#
#   Warning:
#       This script is only expected to work for Linux platforms provided of
#       either required package dialog or whiptail.
#       If none of the previous is found on the host machine, the script will
#       run in command line mode, as in version 1.0.
#
#   Layout:
#   - Section 1:          setting up graphics environment variables
#   + Section 2:          setting up NJoy
#       - function MATfinder:  find MAT number for given isotope
#       - function MATcomputer: compute MAT number for given isotope
#       - function NJoyInput:  create NJoy input file for given isotope
#       - function MakeLib:    collect outputs and build xsec library
#       - function NJoyer:     process cross sections libraries
#   + Section 3:          setting up script
#       - function log_msg:    direct messages to log file
#       - function info_box:   show an information box with a message
#       - function check_environment: check main environment variables
#       - function check_njoy:  check existence of NJoy executable
#       - function check_variables: check variables for library creation
#       - function check_more_options: check user specified auxiliary options
#       - function check_files: check input libraries in specified path
#   + Section 4:          executing script
#       - function Usage:      print help menu
```

```

# - function run_njoy:          check all parameters and run NJoy
#
# Modifications:
#
#
# *****

script_version="3.0"

# ***** #
#                               Section 1, setting up graphics                               #
# ----- #
# This section sets up the graphics environment. This is where the system is #
# checked for dialog component and dialog variables are defined.             #
# ***** #
GRAPHICAL="yes"
DLG="dialog"
DLG_BACKTITLE="NJoyInputMaker v.$script_version"
DLG_HEIGHT="5"
DLG_DBLE_HEIGHT="10"
DLG_HEIGHT_WIDE="28"
DLG_TINY_WIDTH="30"
DLG_WIDTH="68"
DLG_WIDTH_WIDE="80"

DLG=$(which dialog)
# Guard against bad "which" implementations that return
# "no <exe> in path1 path2 ..." (these implementations also
# return 0 as the exit status).
if [[ "${DLG#no}" != "${DLG}" ]] ; then
    DLG=""
fi
if [[ "$DLG" == "" ]] ; then
    DLG=$(which whiptail)
    if [[ "${DLG#no}" != "${DLG}" ]] ; then
        DLG=""
    fi
    if [[ "$DLG" == "" ]] ; then
        GRAPHICAL="no"
        echo ""
        echo "Unable to find utility for graphical build..."
        echo "Continuing in command line mode..."
        echo ""
    fi
fi

# ***** #
#                               Section 2, setting up NJoy                               #
# ----- #
# This section sets up the NJoy environment. This is where are defined the #
# main execution variables and all the functions to set up and run Njoy.    #
# ***** #
#
# OPTIONS

```

```

#
libpath="./library"           # Original library source folder
libname="JEF"                 # Original library name...
libversion="2.2"              # ...and version
library="newlib"              # New library name
outpath="./processed_library" # New library output folder
xsdir=xsdir                   # xsdir file for processed libraries entries
suffix=".60"                  # New library suffix
njoy="./"                     # Path to NJoy executable
xnjoy="xnjoy"                 # Name of NJoy executable (program's default)
njoyversion="99.0"            # Available NJoy version
author="'whoami'"
byauthor="by $author"

# ***** #
# Function: MATcomputer #
# #
# Purpose: MAT number finder #
# #
# Programmer: Giacomo L.A. Grasso, #
# Date: Oct. 24th, 2007 15:33:29 GMT +0100 #
# #
# ***** #
MATcomputer() {
  case "$zz" in
    01)
      main=001
      ;;
    02)
      main=003
      ;;
    03)
      main=006
      ;;
    04)
      main=009
      ;;
    05)
      main=010
      ;;
    06)
      main=012
      ;;
    07)
      main=014
      ;;
    08)
      main=016
      ;;
    09)
      main=019
      ;;
    11)
      main=023
      ;;
  )

```

```
12)
    main=024
;;
13)
    main=027
;;
14)
    main=028
;;
15)
    main=031
;;
16)
    main=032
;;
17)
    main=035
;;
18)
    main=036
;;
19)
    main=039
;;
20)
    main=040
;;
21)
    main=045
;;
22)
    main=046
;;
23)
    main=051
;;
24)
    main=050
;;
25)
    main=055
;;
26)
    main=054
;;
27)
    main=059
;;
28)
    main=058
;;
29)
    main=063
;;
31)
```

```
    main=069
;;
32)
    main=070
;;
33)
    main=075
;;
34)
    main=074
;;
35)
    main=079
;;
36)
    main=078
;;
37)
    main=085
;;
38)
    main=084
;;
39)
    main=089
;;
40)
    main=090
;;
41)
    main=093
;;
42)
    main=092
;;
43)
    if [ "$libname" == "JEF" ] ; then
        main=097
    else
        main=099
    fi
;;
44)
    main=096
;;
45)
    main=103
;;
46)
    main=102
;;
47)
    main=107
;;
48)
```

```
    main=106
;;
49)
    main=113
;;
50)
    main=112
;;
51)
    main=121
;;
52)
    main=120
;;
53)
    main=127
;;
54)
    main=124
;;
55)
    main=133
;;
56)
    main=130
;;
57)
    main=138
;;
58)
    main=136
;;
59)
    main=141
;;
60)
    main=142
;;
61)
    main=139
;;
62)
    main=144
;;
63)
    main=151
;;
64)
    main=152
;;
65)
    main=159
;;
66)
    main=156
```

```
;;
67)
    main=165
;;
68)
    main=162
;;
71)
    main=175
;;
72)
    main=174
;;
73)
    main=180
;;
74)
    main=180
;;
75)
    main=185
;;
77)
    main=191
;;
79)
    main=197
;;
80)
    main=196
;;
82)
    main=204
;;
83)
    main=209
;;
88)
    main=223
;;
89)
    main=225
;;
90)
    main=227
;;
91)
    main=229
;;
92)
    main=234
;;
93)
    main=230
;;
```

```

    94)
        main=235
    ;;
    95)
        main=235
    ;;
    96)
        main=240
    ;;
    97)
        main=240
    ;;
    98)
        main=240
    ;;
    99)
        main=257
    ;;
esac
if [ "$aaa" == "000" ] ; then
    at=00
else
    at=$(( (${aaa#0}-${main#0})*3+25 ))
fi
mat=${zz#0}$at
if [ "$ism" == "m" ] ; then
    mat=$(( $mat+1 ))
fi
if [ "$zz" == "99" ] ; then
    case "$aaa" in
        253)
            mat=9913
            ;;
        254)
            mat=9914
            ;;
        255)
            mat=9916
            ;;
    esac
fi
}

# ***** #
# Function: NJoyInput #
# # #
# Purpose: NJoy input file creation #
# # #
# Programmer: Giacomo L.A. Grasso, #
# Date: Oct. 24th, 2007 15:33:29 GMT +0100 #
# # #
# ***** #

NJoyInput() {
    if [ "$zz" -le "82" ] ; then
        #structural or FP
    fi
}

```

```

    cat>input <<EOF
moder
20 -21
reconr
-21 -22
'Pendf tape for ZZ-Ss-AAAm from LIB-LVER T=TEMPK'/
NMAT 3 / mat, ncards
.002 0. 5 .005 / err, tempr, errmax, errint
'ZZ-Ss-AAAm from LIB-LVER T=TEMPK BYAUTHOR'/
'Processed with the NJOYNVER nuclear data processing system'/
'by means of njoyinputmaker v.SVER (copyright LIN c 2008 G.L.A. Grasso)'/
0 /
broadr
-21 -22 -23
NMAT 1 0 0 0. / mat1, ntemp2, istart, istrap
.002 / errthn = 0.2% thinning (1.0%max)
TEMP / temp2 (K)
0 /
unresr
-21 -23 -24
NMAT 1 1 1 / matd, ntemp, nsigz, iprint
TEMP / temp (K)
1.E10 / sigz
0 /
thermr
0 -24 -30
0 NMAT 8 1 1 0 1 221 2 / matde,matdp,nbin,ntemp,iinc,icoh,natom,mtref,iprint
TEMP / tempr
0.001 1.0 / tol, emax(for thermal treatment)
heatr
-21 -30 -31/
NMAT 2 0 0 0 2 / matd, npk, nqa, ntemp, local, iprint
443 444 / kermatot, dpatot
gaspr
-21 -31 -25
acer / Generate the Ace formatted library
-21 -25 0 31 32 / nendf, npend, ngend, nace, ndir
1 1 1 .SF 0 /iopt, iprint, ntype, suff, nextra
'ZZ-Ss-AAAm from LIB-LVER T=TEMPK BYAUTHOR'/
NMAT TEMP / mat, temp
1 1 /newfor, iopp
/
stop
EOF

else
#fuel
    cat>input <<EOF
moder
20 -21
reconr
-21 -22
'Pendf tape for ZZ-Ss-AAAm PT from LIB-LVER T=TEMPK'/
NMAT 3 / mat, ncards
.001 0. 5 .005 / err, tempr, errmax, errint

```

```

'ZZ-Ss-AAAm PT from LIB-LVER T=TEMPK BYAUTHOR'/
'Processed by the NJOYNVER nuclear data processing system'/
'by means of njoyinputmaker v.SVER (copyright LIN c 2008 G.L.A. Grasso)'/
0 /
broadr
-21 -22 -23
NMAT 1 0 0 0. / mat1, ntemp2, istart, istrap, temp1
.001 / errthn = 0.1% thinning (<1.0%max)
TEMP / temp2 (K)
0 /
unresr
-21 -23 -24
NMAT 1 7 1 / matd, ntemp, nsigz, iprint
TEMP / temp (K)
1.e10 1.e4 1.e3 300. 100. 30. 10./sigz
0 /
heatr
-21 -24 -25/
NMAT 4 0 0 0 2 / matd, npk, nqa, ntemp, local, iprint
302 318 402 443 / kerma: el,fiss, disapp, total
purrr
-21 -25 -26
NMAT 1 7 20 32 1 0/ matd,ntemp,nsigz,nbin, nladr,iprint,nunx
TEMP / ntemp
1.e10 1.e4 1.e3 300. 100. 30. 10./sigz
0 /
thermr
0 -26 -27
0 NMAT 8 1 1 0 1 221 2 / matde,matdp,nbin,ntemp,iinc,icoh,natom,mtref,iprint
TEMP / tempr
0.001 1.0 / tol, emax(for thermal treatment)
gaspr
-21 -27 -28
acer / Generate Ace formatted library
-21 -28 0 31 32 / nendf, npend, ngend, nace, ndir
1 1 1 .SF 0 /iopt, iprint, ntype, suff, nxtra
'ZZ-Ss-AAAm PT from LIB-LVER T=TEMPK BYAUTHOR'/
NMAT TEMP / mat, temp
1 1 /newfor, iopp
/
stop
EOF

```

```

fi

```

```

sed {s/"NMAT"/"$mat"/} input > itemp1
sed {s/"TEMP"/"$temp"/} itemp1 > itemp2
sed {s/"ZZ"/"$zz"/} itemp2 > itemp3
sed {s/"Ss"/"$ss"/} itemp3 > itemp4
sed {s/"AAAm"/"$aaa$ism"/} itemp4 > itemp5
sed {s/"LIB"/"$libname"/} itemp5 > itemp6
sed {s/"LVER"/"$libversion"/} itemp6 > itemp7
sed {s/"NVER"/"$njoyversion"/} itemp7 > itemp8
sed {s/"SVER"/"$script_version"/} itemp8 > itemp9
sed {s/"BYAUTHOR"/"$byauthor"/} itemp9 > itemp10

```

```

sed {s/".SF"/"$suffix"/} itemp10 > park
rm itemp* input
mv park input
}

# ***** #
# Function: MakeLib #
# #
# Purpose: Outputs collection and xsec library build #
# #
# Programmer: Giacomo L.A. Grasso, #
# Date: Oct. 24th, 2007 15:33:29 GMT +0100 #
# #
# ***** #
MakeLib() {
  tot_lib_lines=1
  cd "$outpath"/xsdirs
  for processed_lib in `ls *.xsdir`
  do
    temp_name=./libs/${processed_lib}/xsdir/"$suffix"c}
    lib_name=${temp_name:0:10}${temp_name:12:8}
    temp_lines=`wc -l $lib_name`
    lines=${temp_lines/$lib_name/}
    processing_file=`cat $processed_lib`
    libname=${processing_file/filename/$library}
    route=${libname/route/0}
    startline=${route/"0 1 1"/"0 1 $tot_lib_lines"}
    echo "$startline" >> "../$xsdir"
    cat $lib_name >> "../$library"
    let "tot_lib_lines += $lines"
  done
}

# ***** #
# Function: NJoyer #
# #
# Purpose: Cross section libraries automatic processing #
# #
# Programmer: Giacomo L.A. Grasso, #
# Date: Oct. 24th, 2007 15:33:29 GMT +0100 #
# #
# ***** #
NJoyer() {
  echo " "
  echo "Processing library $libname-$libversion with NJoy$njoyversion at $temp K"
  cp $selection ./
  exec<$fname
  while read line
  do
    zz=${line:0:2}
    ss=${line:2:2}
    aaa=${line:4:3}
    if [ ${#line} == 8 ] ; then
      ism=m
    else

```

```

        ism=
    fi
    MATcomputer
    echo "  getting endf tape $line"
    cp "$libpath/$line" tape20
    NJoyInput
    echo "  running njoy..."
    log_msg "Processing isotope $line at $temp"
    "$njoy$xnjoy"<input >> $LOG_FILE
    log_msg " "
    log_msg " ----- "
    log_msg " "
    echo "  saving output and pendf files"
    echo " "
    mv output "$outpath"/out/out"$ism"
    mv tape31 "$outpath"/libs/"$zz$aaa$ism$suffix"c
    mv tape32 "$outpath"/xsdirs/"$zz$ss$aaa$ism".xsdir
    rm tape* input
done
if [ "$library" == "" ] ; then
    lowercase_libname=$(echo $libname | tr [:upper:] [:lower:])
    syncopatelibversion=${libversion/"./"}
    library=${lowercase_libname$syncopatelibversionr}
fi
MakeLib
cd -
echo " ----- "
echo "Library processing complete."
echo "NJoy processing output written in file \"$LOG_FILE\""
}

# ***** #
#                               #
#           Section 3, setting up script           #
# ----- #
# This section sets up the njoyinputmaker script.  This is where all the #
# environment variables are checked before execution. #
# ***** #
LOG_FILE=${LOG_FILE-"njoyinputmaker_log"}
FORCE="no"

# ***** #
# Function: log_msg #
# # #
# Purpose: Direct messages to log file #
# # #
# Programmer: Giacomo L.A. Grasso, #
# Date: Apr. 12th, 2008 20:22:37 GMT +0100 #
# # #
# ***** #
function log_msg
{
    echo "$1" >> $LOG_FILE
    return 0
}

```

```

# ***** #
# Function: info_box #
# #
# Purpose: Show an information box with a message #
# #
# Programmer: Giacomo L.A. Grasso, #
# Date: Oct. 16th, 2007 17:18:52 GMT +0100 #
# #
# ***** #
function info_box
{
    if [[ "$GRAPHICAL" == "yes" ]] ; then
        $DLG --backtitle "$DLG_BACKTITLE" --infobox "$1" $DLG_HEIGHT $DLG_WIDTH
    fi
    return 0
}

# ***** #
# Function: check_environment #
# #
# Purpose: Check to make sure that the main environment variables are set. #
# #
# Programmer: Giacomo L.A. Grasso, #
# Date: Oct. 16th, 2007 17:18:52 GMT +0100 #
# #
# ***** #
function check_environment
{
    check_njoy
    response=$?
    if [[ $response != 0 ]]; then
        return $?
    fi

    check_variables
    response=$?
    if [[ $response != 0 ]]; then
        return $response
    fi

    return 0
}

# ***** #
# Function: check_njoy #
# #
# Purpose: Check the existence of NJoy executable and retrieves its version. #
# #
# Programmer: Giacomo L.A. Grasso, #
# Date: Oct. 16th, 2007 17:18:52 GMT +0100 #
# #
# ***** #
function check_njoy
{

```

```

# Show a screen
if [[ "$GRAPHICAL" == "yes" ]] ; then
    info_box "Checking for njoy . . ."
fi
test -f "$njoy$xnjoy" && cyclenjoy="yes" || cyclenjoy="no"
while [ "$cyclenjoy" == "no" ] ;
do
    if [[ "$GRAPHICAL" == "yes" ]] ; then
        trovato="$($DLG --backtitle "$DLG_BACKTITLE" --stdout \
            --title "ERROR! No NJoy executable found on specified path.\nPlease specify a valid one:" \
            --fselect "$njoy" $DLG_DBL_HEIGHT $DLG_WIDTH )"
        case $? in
            0)
                njoy="{trovato%/*}"
                xnjoy="{trovato#"njoy"}";;
            1)
                return 1;;
            255)
                return 255;;
        esac
    else
        echo "ERROR! No NJoy executable found on specified path. Aborting."
        return 3
    fi
done
cat>jnput <<EOF
stop
EOF
"$njoy$xnjoy"<jnput >jutput
string="cat jutput"
string1="{string#*njoy}"
newnjoyversion="{string1%/" "*"
rm jutput jnput
if [[ "$GRAPHICAL" == "yes" ]] ; then
    info_box "NJoy version $newnjoyversion correctly found in specified path."
fi
njoyversion=$newnjoyversion

return 0
}

# ***** #
# Function: check_variables #
# #
# Purpose: Check all the variables necessar for njoy execution and library #
#         creation. #
# #
# Programmer: Giacomo L.A. Grasso, #
# Date: Oct. 16th, 2007 17:18:52 GMT +0100 #
# #
# ***** #
function check_variables
{
    if ! test -f "$selection"; then
        if [[ "$GRAPHICAL" == "yes" ]] ; then

```

```

    fpath="`pwd`"
    oradir="${selection%$fname}"
    selection=$oradir
    fname="."
    while test -d "$fname" ;
    do
        cd "$selection"
        actual="`pwd`/"
        selection="$($DLG --backtitle "$DLG_BACKTITLE" --stdout \
            --title "Select $OPTION input file" \
            --fselect "$actual" $DLG_DBLE_HEIGHT $DLG_WIDTH )"
        retrnval=$?
        case $retrnval in
            0)
                fname="${selection#$actual}";;
            1)
                return 1;;
            255)
                return 255;;
        esac
    done
    cd $fpath
else
    echo "ERROR! Specified input file does not exists. Aborting..."
    return 2
fi
else
    cp $selection $fname
fi

if test -f "$library" ; then
    if [[ "$FORCE" == "no" ]]; then
        if [[ "$GRAPHICAL" == "yes" ]] ; then
            "$DLG" --clear --backtitle "$DLG_BACKTITLE" \
                --yes-label "Yes" --no-label "No" \
                --yesno "$library file already exists.\n\
Overwrite with new library?" $DLG_HEIGHT_WIDE $DLG_WIDTH
            case $? in
                0) rm "$library";;
                1) return 5;;
            esac
        else
            echo "ERROR! Specified output library already exists. Aborting..."
            return 5
        fi
    else
        rm "$library"
    fi
fi

if test -f "$LOG_FILE" ; then
    if [[ "$FORCE" == "no" ]]; then
        if [[ "$GRAPHICAL" == "yes" ]] ; then
            "$DLG" --clear --backtitle "$DLG_BACKTITLE" \
                --yes-label "Yes" --no-label "No" \

```

```

        --yesno "$LOG_FILE file already exists.\n\
Overwrite with new log?" $DLG_HEIGHT_WIDE $DLG_WIDTH
        case $? in
            0) rm "$LOG_FILE";;
            1) return 6;;
        esac
    else
        echo "ERROR! Specified log file already exists. Aborting..."
        echo "Try with \"-f\" option."
        return 6
    fi
else
    rm "$LOG_FILE"
fi
fi

if [[ "$temp" == "" ]]; then
    $DLG --backtitle "$DLG_BACKTITLE" \
        --no-cancel --inputbox \
"Enter processing temperature [K]:" 0 $DLG_WIDTH_WIDE "$temp" 2> timp$$
    temp="$(cat timp$$)"
    if [[ -e "timp$$" ]] ; then
        rm timp$$
    fi
fi

return 0
}

# ***** #
# Function: check_more_options #
# #
# Purpose: Check the consistency of user specified accessory options #
# #
# Programmer: Giacomo L.A. Grasso, #
# Date: Oct. 16th, 2007 17:18:52 GMT +0100 #
# #
# ***** #
function check_more_options
{
    if test -d "$outpath" ; then
        if [[ "$FORCE" == "yes" ]]; then
            rm -fr "$outpath"
            log_msg "WARNING! Output directory already exists. Overwriting...!"
        else
            return 7
        fi
    fi
    mkdir "$outpath"
    mkdir "$outpath"/out
    mkdir "$outpath"/libs
    mkdir "$outpath"/xsdirs

    while ! test -d "$libpath" ;

```

```

do
  if [[ "$GRAPHICAL" == "yes" ]] ; then
    $DLG --backtitle "$DLG_BACKTITLE" \
      --no-cancel --inputbox \
"Enter valid input library path:" 0 $DLG_WIDTH_WIDE "$temp" 2> tomp$$
    libpath="$(cat tomp$$)"
    if [[ -e "$libpath" ]] ; then
      rm tomp$$
    fi
  else
    echo "ERROR! Input libraries path does not exists. Aborting..."
    return 8
  fi
done

check_files
response=$?
if [[ $response != 0 ]]; then
  return $response
fi

return 0
}

# ***** #
# Function: check_files #
# #
# Purpose: Check for input libraries to exist in specified path. #
# #
# Programmer: Giacomo L.A. Grasso, #
# Date: Oct. 16th, 2007 17:18:52 GMT +0100 #
# #
# ***** #
function check_files
{
  missing_libs="0"
  exec<$selection
  while read line
  do
    if [[ ! -f "$libpath/$line" ]]; then
      if [[ "$missing_libs" == "0" ]]; then
        log_msg "ERROR! Found invalid input libraries"
      fi
      missing_libs="1"
      log_msg "Invalid library: $line"
    fi
  done
  if [[ "$missing_libs" != "0" ]]; then
    return 4
  fi
  return 0
}

# ***** #

```

```

#                               Section 4, executing script                               #
# -----#
# This section does some set up needed for creating the cross section library #
# as initializing the variables by either processing the command line stream #
# or visualizing the GUI dialogs, and then runs the njoyinputmaker script #
# kernel. #
# *****#

# *****#
# Function: Usage #
# # #
# Purpose: Display njoyinputmaker help menu #
# # #
# Programmer: Giacomo L.A. Grasso, #
# Date: Oct. 16th, 2007 17:18:52 GMT +0100 #
# # #
# *****#

Usage () {
    echo "njoyinputmaker script v.3.0"
    echo " "
    echo "Usage:"
    echo "      njoyinputmaker.sh [OPTIONS] [library]"
    echo " "
    echo "The [library] argument is mandatory in command line mode."
    echo " "
    echo " "
    echo "OPTIONS:"
    echo "  -a|--author <author>          Specify the name of the author processing"
    echo "                                the library files"
    echo "                                (default: <author>=\"whoami'\")"
    echo "  -e|--executable <xnjoy>       Specify the path to the xnjoy executable"
    echo "                                (default: <xnjoy>=\"./xnjoy\")"
    echo "  -f                             Force overwriting of existing files"
    echo "  -h|--help                     Print out this help menu"
    echo "  -i|--file-input <inputfile>   Specify the file containing the ASCII table"
    echo "                                of zaid numbers for the processing nuclides"
    echo "                                (no default: mandatory argument in command"
    echo "                                line mode)"
    echo "  -I|--input-library-path <libpath> Specify the input directory path for"
    echo "                                original library files"
    echo "                                (default: <libpath>=\"./library\")"
    echo "  -l|--library <libname>        Specify the input library name"
    echo "                                (default: <libname>=\"JEF\")"
    echo "  -O|--output-path <outpath>    Specify the output directory path for"
    echo "                                processed library files"
    echo "                                (default: <outpath>=\"processed_library\")"
    echo "  -s|--suffix <suffix>         Specify the two-digit suffix to append new"
    echo "                                libraries to"
    echo "                                (default: <suffix>=\".60\")"
    echo "  -T|--temp <temperature>       Specify the processing temperature [Kelvin]"
    echo "                                (no default: mandatory argument in command"
    echo "                                line mode)"
    echo "  -v|--version                 Print out the version number of this script"
    echo "  -V|--library-version <libversion> Specify the input library version"
    echo "                                (default: <libversion>=\"2.2\")"

```

```

echo "  -x|--xmdir <xmdir>                Specify the xmdir file to be upgraded with"
echo "                                     new libraries entries"
echo "                                     (default: <xmdir>=\"xmdir\")"
echo " "
exit 0
}

# ***** #
# Function: run_njoy                               #
#                                                  #
# Purpose: Check all parameters before running NJoy #
#                                                  #
# Programmer: Giacomo L.A. Grasso,                #
# Date: Oct. 16th, 2007 17:18:52 GMT +0100        #
#                                                  #
# ***** #
function run_njoy
{
    check_environment
    response=$?
    if [[ $response != 0 ]] ; then
        echo "ERROR! Library creation aborted because of bad input arguments setting error."
        log_msg "ERROR $response. Stopping library creation."
        exit $response
    fi
    check_more_options
    response=$?
    if [[ $response != 0 ]] ; then
        echo "ERROR! Library creation aborted because of bad variable option setting error."
        log_msg "ERROR $response. Stopping library creation."
        exit $response
    fi

    #
    # Log the start time.
    #
    LINES="-----"
    log_msg $LINES
    log_msg "Starting processing libraries at 'date'"
    log_msg $LINES

    NJoyer
    return 0
}

#
# INPUT ARGUMENTS PARSING
#
while [ $# -gt 0 ]; do
    case "$1" in
        -h | --help)
            Usage
            ;;
        -v | --version)
            echo "njoyinputmaker.sh v. $script_version"
            ;;
    esac
    shift
done

```

```
    exit 0
;;
-f)
    FORCE="yes"
    shift
;;
-I | --input-library-path)
    libpath=$2
    shift
    shift
;;
-l | --library)
    libname=$2
    shift
    shift
;;
-V | --library-version)
    libversion=$2
    shift
    shift
;;
-O | --output-path)
    outpath=$2
    shift
    shift
;;
-x | --xsdir)
    xsdir=$2
    shift
    shift
;;
-s | --suffix)
    suffix=$2
    shift
    shift
;;
-e | --executable)
    njoy=$2
    shift
    shift
;;
-a | --author)
    byauthor="by $2"
    shift
    shift
;;
-i | --file-input)
    fname=$2
    shift
    shift
;;
-T | --temp)
    temp=$2
    shift
    shift
```

```

;;
* )
    library=$1
    shift
;;
esac
done

if [[ "$GRAPHICAL" == "no" ]] ; then
    while [ "$fname" == "" ]; do
        echo " "
        echo "WARNING! The input file is a mandatory argument!"
        echo "Please enter input file name:"
        read fname
    done
    while [ "$temp" == "" ]; do
        echo " "
        echo "WARNING! The processing temperature is a mandatory argument!"
        echo "Please enter temperature [K]:"
        read temp
    done
fi

if [[ "$GRAPHICAL" == "yes" ]] ; then
    while [ ${retval:-99} -ne 1 -a ${retval:-99} -ne 250 ] ;
    do
        $DLG --backtitle "$DLG_BACKTITLE" \
        --title "Execution options" \
        --menu \
        "Welcome to the njoyinputmaker v.$script_version interface for cross "\
        "sections datasets automatic creation with NJoy.\n\n"\
        "This program will build an endf cross sections library "\
        "and the corresponding xsdir entries for use in MCNP.\n\n"\
        "Select the build options:" $DLG_HEIGHT_WIDE $DLG_WIDTH 6 \
        "Database"      "select input nuclides table" \
        "Temperature"   "specify processing temperature [K]" \
        "OutputLib"     "specify output library name" \
        "LogFile"       "specify log filename" \
        "More"          "select more options" \
        "Run"           "execute NJoy" 2> tmp$$

        retval=$?

        # Remove the extra quoting, new dialog has --single-quoted
        choice="$(cat tmp$$ | sed 's/"//g' )"
        case $retval in
            0)
                for OPTION in $choice
                do
                    case $OPTION in
                        Database)
                            fpath="`pwd`"
                            fname="."
                            selection="$fpath"
                            while test -d "$fname" ;

```

```

do
  cd "$selection"
  actual="'pwd'/"
  selection="$($DLG --backtitle "$DLG_BACKTITLE" --stdout \
    --title "Select $OPTION input file" \
    --fselect "$actual" $DLG_DBLE_HEIGHT $DLG_WIDTH )"
  retnval=$?
  case $retnval in
    0)
      fname="{selection#"$actual"}";;
    1)
      log_msg "ERROR 1"
      exit 1;;
    255)
      log_msg "ERROR 255"
      exit 255;;
  esac
done
cd $fpath
;;
Temperature)
  $DLG --backtitle "$DLG_BACKTITLE" \
    --no-cancel --inputbox \
"Enter $OPTION value:" 0 $DLG_WIDTH_WIDE "$temp" 2> tmp$$
  temp="$(cat tmp$$)"
  ;;
OutputLib)
  $DLG --backtitle "$DLG_BACKTITLE" \
    --no-cancel --inputbox \
"Enter $OPTION value:" 0 $DLG_WIDTH_WIDE "$library" 2> tmp$$
  library="$(cat tmp$$)"
  ;;
LogFile)
  $DLG --backtitle "$DLG_BACKTITLE" \
    --no-cancel --inputbox \
"Enter $OPTION value:" 0 $DLG_WIDTH_WIDE "$LOG_FILE" 2> tmp$$
  LOG_FILE="$(cat tmp$$)"
  ;;
More)
  newlibpath="$libpath"
  newlibname="$libname"
  newlibversion="$libversion"
  newoutpath="$outpath"
  newxsdir="$xsdir"
  newsuffix="$suffix"
  newnjoy="$njoy"
  newauthor="$author"
  while [ ${returncode:-99} -ne 1 -a ${returncode:-99} -ne 250 ]; do
    value="$("$DLG" --stdout --clear --ok-label "Save" \
      --extra-label "Edit" --cancel-label "Close" --backtitle "$DLG_BACKTITLE" \
      --inputmenu " LibPath: specify path to input database libraries\n \
LibName: specify input database libraries name\n \
LibVersion: specify input database libraries version\n \
OutPath: select processed library output path\n \
xsdir: specify the output xsdir file name\n \

```

```

Suffix: specify the created library suffix\n \
NJoy: specify the path to the NJoy executable\n \
Author: specify the new library author\n\n \
Select optional parameters:" $DLG_HEIGHT_WIDE $DLG_WIDTH 8 \
    "LibPath:"      "$newlibpath" \
    "LibName:"      "$newlibname" \
    "LibVersion:"   "$newlibversion" \
    "OutPath:"      "$newoutpath" \
    "xsdir:"        "$newxsdir" \
    "Suffix:"       "$newsuffix" \
    "NJoy:"         "$newnjoy" \
    "Author:"       "$newauthor" )"
returncode=$?
case $returncode in
0)
    libpath="$newlibpath"
    libname="$newlibname"
    libversion="$newlibversion"
    outpath="$newoutpath"
    xsdir="$newxsdir"
    suffix="$newsuffix"
    njoy="$newnjoy"
    author="$newauthor"
    byauthor="by $author"
    break
;;
3)
    value="${value:8:${#value}}"
    tag="${value%:*}"
    item="${value#*: }"
    case $tag in
        LibPath)
            newlibpath="$item";;
        LibName)
            newlibname="$item";;
        LibVersion)
            newlibversion="$item";;
        OutPath)
            newoutpath="$item";;
        xsdir)
            newxsdir="$item";;
        Suffix)
            newsuffix="$item";;
        NJoy)
            newnjoy="$item";;
        Author)
            newauthor="$item";;
    esac
;;
1)
    "$DLG" --clear --backtitle "$DLG_BACKTITLE" \
        --yesno "All changes will be lost. Really close?" $DLG_DBLE_HEIGHT $DLG_TINY_WIDTH
    case $? in
        0) break;;
        1) returncode=99;;
    esac

```

```

        esac
        ;;
    255)
        break
        ;;
    *)
        log_msg "ERROR 10 ($returncode)"
        exit 10
        ;;
    esac
done
;;
Run)
#
# See if the user needs to modify some variables
#
"$DLG" --clear --backtitle "$DLG_BACKTITLE" \
--yes-label "Run" --no-label "Back" \
--yesno "Options summary:\n \
Input Database file:      $fname\n \
Processing Temperature:  $temp\n \
Output library name:     $library\n \
Log file name:          $LOG_FILE\n \
Input library path:      $libpath\n \
Input library name:      $libname\n \
Input library version:   $libversion\n \
Output library path:     $outpath\n \
xsdir descriptor file:   $xsdir\n \
Output library extension: $suffix\n \
NJoy executable path:    $njoy\n \
Author:                  $author\n\n\
Are they correct?" $DLG_HEIGHT_WIDE $DLG_WIDTH
case $? in
    0)
        echo "Running NJoy..."
        run_njoy
        if [[ -e "tmp$$" ]] ; then
            rm tmp$$
        fi
        exit 0;;
    1)
        retval=99;;
    esac
;;
esac
done
;;
1)
"$DLG" --clear --backtitle "$DLG_BACKTITLE" \
--yesno "Really quit?" $DLG_DBLE_HEIGHT $DLG_TINY_WIDTH
case $? in
    0) log_msg "ERROR 1"; exit 1;;
    1) retval=99;;
    esac;;
255)

```

```
        log_msg "ERROR 255"
        exit 255;;
    *)
        log_msg "ERROR 10 ($returncode)"
        exit 10;;
    esac
done
else
    echo "Running NJoy..."
    run_njoy
    exit 0
fi
```

B Help menu

njoyinputmaker script v.3.0

Usage:

njoyinputmaker.sh [OPTIONS] [library]

The [library] argument is mandatory in command line mode.

OPTIONS:

-a --author <author>	Specify the name of the author processing the library files (default: <author>="username")
-e --executable <xnjoy>	Specify the path to the xnjoy executable (default: <xnjoy>="./xnjoy")
-f	Force overwriting of existing files
-h --help	Print out this help menu
-i --file-input <inputfile>	Specify the file containing the ASCII table of zaid numbers for the processing nuclides (no default: mandatory argument in command line mode)
-I --input-library-path <libpath>	Specify the input directory path for original library files (default: <libpath>="./library/")
-l --library <libname>	Specify the input library name (default: <libname>="JEF")
-O --output-path <outpath>	Specify the output directory path for processed library files (default: <outpath>="processed_library/")
-s --suffix <suffix>	Specify the two-digit suffix to append new libraries to (default: <suffix>=".60")
-T --temp <temperature>	Specify the processing temperature [Kelvin] (no default: mandatory argument in command line mode)
-v --version	Print out the version number of this script
-V --library-version <libversion>	Specify the input library version (default: <libversion>="2.2")
-x --xmdir <xmdir>	Specify the xmdir file to be upgraded with new libraries entries (default: <xmdir>="xmdir")

C Sample input file

080_016
080_017
92U_234
92U_235
92U_236
92U_238
94PU238
94PU239
94PU240
94PU241
94PU242
94PU243
94PU244
95AM241
95AM242
95AM242M
95AM243
96CM242
96CM243
96CM244
96CM245
96CM246
96CM247
96CM248
93NP237
93NP239

D Error list

+-----+-----+-----+		
ERROR #	Description	
+-----+-----+-----+		
1	Cancel pressed.	
+-----+-----+-----+		
2	Specified input file does not exists (mandatory).	
+-----+-----+-----+		
3	No NJoy executable found on specified path.	
+-----+-----+-----+		
4	Missing input libraries in specified path.	
+-----+-----+-----+		
5	Specified output library already exists (and no force option).	
+-----+-----+-----+		
6	Specified log file already exists (and no force option).	
+-----+-----+-----+		
7	Output directory already exists (and no force option).	
+-----+-----+-----+		
8	Input libraries path does not exists.	
+-----+-----+-----+		
10	Unexpected return code from dialog window (code #)	
+-----+-----+-----+		
255	ESC pressed.	
+-----+-----+-----+		

References

- [1] R. E. MacFarlane and D. W. Muir. The NJOY Nuclear Data Processing System Version 91. Technical Report LA-12740-M, Los Alamos National Laboratories, October 1994. [4](#)
- [2] G. Grasso. MAT numbers table for isotope processing with NJoy. Technical Report LIN-M01.2007, LIN, 2007. [4](#)
- [3] *NJOY99.0 – Code System for Producing, Pointwise and Multigroup Neutron and Photon Cross Sections from ENDFB Data*, 2000. [4](#)
- [4] M. B. Chadwick *et al.* ENDF/B-VII.0: Next Generation Evaluated Nuclear Data Library for Nuclear Science and Technology. *Nucl. Data Sheets*, 107(12):2931–3118, 2006. [4](#)
- [5] F. B. Brown *et al.* MCNP Version 5. *Trans. Am. Nucl. Soc.*, 87:273, 2002. [4](#)
- [6] G. Grasso. NJoyInputMaker: a tool for automatically processing cross section libraries with NJoy. Technical Report LIN-R06.2007, LIN, 2007. [4](#)