

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-R06.2007
Subject: NJoyInputMaker
Date: October 28, 2007

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

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

The present work describes the creation of a script for automatic processing of cross section libraries in order to create useful datasets for nuclear systems analysis tools, like MCNP, by means of the well-known NJoy nuclear data processing code.

The resulting “njoyinputmaker.sh” script allows the user to automatically process a given list of isotopes at a given temperature by creating multiple appropriated input script for the NJoy code and running the latter to process them.

The Author
Giacomo Grasso

The Supervisor
prof. Marco Sumini

Revisions List:

Date	Author

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

Giacomo Grasso

October 28, 2007

Abstract

The present work describes the creation of a script for automatic processing of cross section libraries in order to create useful datasets for nuclear systems analysis tools, like MCNP, by means of the well-known NJoy nuclear data processing code.

The resulting “njoyinputmaker.sh” script allows the user to automatically process a given list of isotopes at a given temperature by creating multiple appropriated input script for the NJoy code and running the latter to process them.

Contents

1	Introduction	3
2	Description of the script features	4
A	njoyinputmaker.sh source	6

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 temperature of reference.

The scope of this work is to develop an automatic script for the sequential processing of a list of isotopes at the same temperature, retrieving the original library and the corresponding MAT numbers [2], compiling an appropriated input script [3] and running the NJoy code and finally storing the final processed library file in a folder and updating the xsdir library database (the script has been specifically developed for ENDF [4] library output format for use in the MCNP [5] nuclear code).

2 Description of the 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. On this leading guide, it has been oriented to let the user specify a variety of input parameters for the characterization of the processed libraries. A help menu has been added in order to guide the user to a correct learning of the script syntax and options; it can be retrieved by tapping:

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

or

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

The output capture is the following:

```
njoyinputmaker script v.2.0
```

Usage:

```
njoyinputmaker.sh [options] -f|--file-input <inputfile> -T|--temp <temperature>
where <inputfile> and <temperature> are mandatory arguments.
```

The <inputfile> must be provided as an ascii table of zaid numbers.

The <temperature> argument must be expressed in Kelvin.

OPTIONS:

-i --input-library-path <libpath>	to specify the input directory path for original library files (default: <libpath>="./library/")
-l --library <libname>	to specify the input library name (default: <libname>="JEFF")
-v --library-version <libversion>	to specify the input library version (default: <libversion>="2.2")
-o --output-path <outpath>	to specify the output directory path for processed library files (default: <outpath>="processed_library/")
-x --xmdir <xmdir>	to specify the xmdir file to be upgraded with new libraries entries (default: <xmdir>="xmdir")
-s --suffix <suffix>	to specify the two-digit suffix to append new libraries to (default: <suffix>=".60")
-e --executable <execpath>	to specify the path to the NJoy executable (xnjoy) (default: <execpath>="./")
-h --help	to print out this help menu

Two mandatory arguments are the file which lists the isotopes zaid to be processed and the temperature of interest, expressed in Kelvin degrees. The input file must be compiled as an ASCII text file, with isotopes zaid provided in column.

The user is then able to tune the script execution by specifying the path (the default is `./library/`) to the folder containing the original libraries to be processed, by means of the `-i|--input-library-path` flag. The input library name and version can be also specified by, respectively, the two flags `-l|--library` and `-v|--library-version` (the default values, JEFF and 2.2, have been chosen depending on the author default choice).

The NJoy executable can be localized elsewhere than default `./` path by means of the `-e|--executable` flag. The script will then check the xjoy program to retrieve the exact version to mark the processed libraries with.

At last, the output libraries can be placed in a user defined folder (using the `-o|--output-path` input flag), the informations needed for MCNP to use the produced libraries can be put in a user defined xsdir file (`-x|--xsdir`) and the suffix to be appended to the libraries for identification specified with the `-s|--suffix` input argument.

A njoyinputmaker.sh source

```
#####
# njoyinputmaker                                     #
#                                                     #
# A shell script for automatic processing of neutron cross section #
# libraries with NJoy                                     #
#                                                     #
#                                                     #
# 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                             #
# Department of Fusion, technologies and                     #
#   Nuclear Aid (FPN)                                         #
# National Agency for the New Technologies,                 #
#   Energy and Environment (ENEA)                             #
#                                                     #
# njoyinputmaker script v.2.0                               #
# author: Giacomo L. A. Grasso                             #
# date: Oct. 24th, 2007                                     #
# LIN (c) Giacomo Grasso 2007                               #
#####
#
#!/bin/bash

# Input parameters check
HelpMenu() {
    echo " "
    echo "njoyinputmaker script v.2.0"
    echo " "
    echo "Usage:"
    echo "    njoyinputmaker.sh [options] -f|--file-input <inputfile> -T|--temp <temperature>"
    echo "where <inputfile> and <temperature> are mandatory arguments."
    echo " "
    echo "The <inputfile> must be provided as an ascii table of zaid numbers."
    echo " "
    echo "The <temperature> argument must be expressed in Kelvin."
    echo " "
    echo " "
    echo "OPTIONS:"
    echo "  -i|--input-library-path <libpath>  to specify the input directory path for original library files"
    echo "                                         (default: <libpath>=\"./library/\")"
    echo "  -l|--library <libname>              to specify the input library name"
    echo "                                         (default: <libname>=\"JEFF\")"
    echo "  -v|--library-version <libversion>   to specify the input library version"
    echo "                                         (default: <libversion>=\"2.2\")"
    echo "  -o|--output-path <outpath>          to specify the output directory path for processed library files"
    echo "                                         (default: <outpath>=\"processed_library/\")"
    echo "  -x|--xsdir <xmdir>                  to specify the xsdir file to be upgraded with new libraries entries"
    echo "                                         (default: <xmdir>=\"xsdir\")"
}
```

```

    echo "  -s|--suffix <suffix>          to specify the two-digit suffix to append new libraries to"
    echo "                                (default: <suffix>=".60")"
    echo "  -e|--executable <execpath>       to specify the path to the NJoy executable (xnjoy)"
    echo "                                (default: <execpath>="./\")"
    echo "  -h|--help                          to print out this help menu"
    echo " "
    exit 1
}

libpath="./library/"
libname="JEFF"
libversion="2.2"
outpath="./processed_libraries/"
xmdir=xmdir
suffix=".60"
njoy="./"
njoyversion="99.0"
while [ $# -gt 0 ]; do
    case "$1" in
        -h | --help)
            HelpMenu
            ;;
        -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
            if test -d $outpath ; then
                echo " "
                echo "WARNING! The specified output directory already exists. Possible data overwriting."
            else
                mkdir $outpath
            fi
            if ! test -d "$outpath"/out ; then
                mkdir "$outpath"/out
            fi
            if ! test -d "$outpath"/xsdirs ; then
                mkdir "$outpath"/xsdirs
            fi
            shift
            shift
            ;;
    esac
done

```

```

    -x | --xmdir)
        xmdir=$2
        shift
        shift
    ;;
    -s | --suffix)
        suffix=$2
        shift
        shift
    ;;
    -e | --executable)
        njoy=$2
        shift
        shift
    ;;
    -f | --file-input)
        fname=$2
        shift
        shift
    ;;
    -T | --temp)
        temp=$2
        shift
        shift
    ;;
esac
done

while [ "$fname" == "" ]; do
    echo " "
    echo "WARNING! The input file is a mandatory argument! Please enter input file name:"
    read fname
done

while [ "$temp" == "" ]; do
    echo " "
    echo "WARNING! The processing temperature is a mandatory argument! Please enter temperature [K]:"
    read temp
done

cat>input <<EOF
    stop
EOF
"$njoy">njoy<input >output
string=$(cat output)
string1=${string#"vers: "}
njoyversion=${string1%" "*}
rm output input

# MAT number finder
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" == "JEFF" ] ; 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
}

# NJoy input file creation
NJoyInput() {
    if [ "$zz" -le "82" ] ; then
        #structural or FP
        cat>input <<EOF
moder
20 -21
reconr
-21 -22
'Pendf tape for ZZ-Ss-AAAm from LIB-LVER T=TEMPK'/
NMAT 2 / mat, ncards
.002 0. 5 .005 / err, tempr, errmax, errint
'ZZ-Ss-AAAm from LIB-LVER T=TEMPK by G.L.A.Grasso'/
'Processed by the NJOYNVER nuclear data processing system'/
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 by G.L.A.Grasso'/
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 2 / mat, ncards
.001 0. 5 .005 / err, tempr, errmax, errint
'ZZ-Ss-AAAm PT from LIB-LVER T=TEMPK by G.L.A.Grasso'/
'Processed by the NJOYNVER nuclear data processing system'/
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
purr
-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 by G.L.A.Grasso'/
NMAT TEMP / mat, temp
1 1 /newfor, iopp
/
stop
EOF

fi

sed {s/"NMAT"/"$mat"/} input > temp1
sed {s/"TEMP"/"$temp"/} temp1 > temp2
sed {s/"ZZ"/"$zz"/} temp2 > temp3
sed {s/"Ss"/"$ss"/} temp3 > temp4
sed {s/"AAAm"/"$aaa$ism"/} temp4 > temp5
sed {s/"LIB"/"$libname"/} temp5 > temp6
sed {s/"LVER"/"$libversion"/} temp6 > temp7
sed {s/"NVER"/"$njoyversion"/} temp7 > temp8
sed {s/".SF"/"$suffix"/} temp8 > park
rm temp* input
mv park input
}

echo " "
echo "Processing library $libname-$libversion with NJoy$njoyversion at $temp K"
exec<$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..."
  echo "Processing isotope $line at $temp" >> outnjoy
  "$njoy"xnjoy<input >> outnjoy
  echo " " >> outnjoy
  echo " ----- " >> outnjoy
  echo " " >> outnjoy
  echo " saving output and pendf files"
  echo " "
  mv output "$outpath"out/out"$mat"
  mv tape31 "$outpath$zz$aaa$ism$suffix"c
  mv tape32 "$outpath"xsdire/"$zz$ss$aaa$ism".xsdir
  cat "$outpath"xsdire/"$zz$ss$aaa$ism" >> $xsdir
  rm tape* input

```

```
done
lowcaselibname=$(echo $libname | tr [:upper:] [:lower:])
syncopatelibversion=${libversion}/"./"
sed {s/"filename"/"$lowcaselibname$syncopatelibversion"r/} $xsdir > park
sed {s/"route"/"0"/} park > $outpath$xsdir
rm $xsdir park
cd $outpath
for libfile in *"$suffix"
do
    mv $libfile ${libfile/"$suffix"/"$suffix"c}
done
cd -
echo "Library processing complete."
echo "NJoy processing output written in file \"outnjoy\""
```

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.
- [2] G. Grasso. MAT numbers table for isotope processing with NJoy. Technical Report LIN-M01.2007, LIN, 2007.
- [3] *NJOY99.0 – Code System for Producing, Pointwise and Multigroup Neutron and Photon Cross Sections from ENDFB Data*, 2000.
- [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.
- [5] F. B. Brown *et al.* MCNP Version 5. *Trans. Am. Nucl. Soc.*, 87:273, 2002.