IAEA

60 Years

Atoms for Peace and Development

endf-parserpy: A Python package for working with ENDF-6 files

Georg Schnabel

Nuclear Data Section
Division of Physical and Chemical Sciences NAPC
Department for Nuclear Sciences and Applications
IAEA, Vienna

CM on Information Exchange on Developments and Operations of Nuclear Data Dissemination Services

16 January 2024

The digital divide

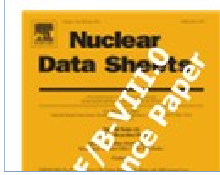
Nuclear data libraries

Nuclear Data Center
Japan Atomic Energy Agency

Top Page in Japanese
Top Page in English

JENDL-5

Evaluated Nuclear Data File (ENDF)



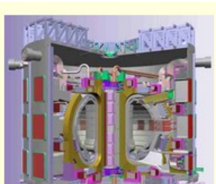
ENDF/B-VIII.0 released February 2, 2018

ENDF/B-VIII.0 fully incorporates the new Neutron Data Standards, includes improved thermal neutron scattering data and uses new evaluated data from the Coordinated International Evaluation Library Organization (CIELO) pilot project for neutron reactions on ^{1}H, ^{16}O, ^{56}Fe, ^{235}U, ^{238}U and ^{239}Pu.

Joint Evaluated Fission and Fusion (JEFF) Nuclear Data Library

JEFF-DOCS 1982 - To date
EFF-DOCS 1984 - To date
JEFF Reports 1985 - To date

CENDL, ROSFOND, ...



Fusion Evaluated Nuclear Data Library - FENDL-3.2b
(Nuclear data supersede all previous versions of FENDL-2.x and 3.x libraries)

And many more ...

Major Libraries

1) ENDF/B-VIII.0 (USA,2018)
2) JEFF-3.3 (Europe,2017)
3) JENDL-5 (Japan,2021)
4) CENDL-3.2 (China,2020)
5) BROND-3.1 (Russia,2016)
6) TENDL-2021 (TALYS, 2021)

IAEA Project Libraries

7) FENDL-3.2b, Fusion, 2022
8) INDEN-Aug2023
9) INDEN-Oct2022
10) IAEA-Med radioisot.prod.2019
11) IAEA-Med diagnostic ri.prod.2001
12) IAEA-Med therapeutical ri.prod.2009
13) IAEA-Photonuclear, 2019
14) IAEA Standards, 2017
15) IAEA Reference cross sections, 2017
16) IAEA High-Energy fission ref., 2015
17) ADS-HE High energy, 2013
18) IRDFF-II Dosimetry, 2019
19) INDL/TSL Thermal Scattering Law, 2006
20) IBA-EVAL diff.data for ion beam analysis, 2013
21) Wind, U,Np,Pu (up to 100 MeV), 1996
22) HE fission by Yashits for Pb-Pu, 2000

Special Libraries

23) JENDL/DDF-2015, JENDL Decay Data File 2015 (Japan)
24) UKDD-2020 decay library, UK, 2020 (UK)
25) JENDL/DEU-2020, Deuteron Reaction Data File 2020 (Japan)
26) JENDL/PD-2016.1, Photonuclear Data File 2016 revision 1, 2020 (Japan)
27) JENDL/ImPACT-2018, LLFP Transmutation Cross Section File (Japan)
28) EAF-2010: European Activation File /816MAT,60MeV/, UK+Netherlands
29) ENDF/HE-VI (High Energy)
30) EPICS-2014 Electron and Photon Interaction Cross Sections (USA,2014)
31) IRDFF-II (auxiliary files), IAEA 2019
32) JEFF-3.1/A (Activation)
33) JENDL-4.0/HE, High Energy File 2015 (neutron, proton)
34) JENDL/AC-2008, JENDL Actinoid File 2008
35) JENDL/AD-2017, Activation Cross Section (Japan,2017)
36) JENDL/AN-2005, (alpha,n) Reaction Data File
37) MINKS-ACT, Actinides Library (Maslov et al.), 1996-2011
38) PADF-2007, Proton Activation Data File, 2007
39) TENDL-2019.s60 /n:2788mat,200MeV/ (by TALYS, 2019)
40) TENDL-2015.s60 /n:19mat,200MeV/ (by TALYS, 2015)
41) W3000, Proton activation cross section data on W (3 GeV), KIT, Germany, 2012

Archival

Derived

ENDF-6 format

```
8.201440+5 2.500000+0 1.704000+2 7.200000-1 0.000000+0 0.000000+02631 2151  320
8.286320+5 2.500000+0 6.790000+2 7.200000-1 0.000000+0 0.000000+02631 2151  321
8.357980+5 2.500000+0 5.364000+2 7.200000-1 0.000000+0 0.000000+02631 2151  322
8.383400+5 2.500000+0 4.450000+1 7.200000-1 0.000000+0 0.000000+02631 2151  323
8.454520+5 2.500000+0 8.830000+2 7.200000-1 0.000000+0 0.000000+02631 2151  324
8.503770+5 2.500000+0 6.340000+1 7.200000-1 0.000000+0 0.000000+02631 2151  325
8.519200+5 2.500000+0 4.060000+1 7.200000-1 0.000000+0 0.000000+02631 2151  326
8.654000+5 2.500000+0 2.673330+2 7.200000-1 0.000000+0 0.000000+02631 2151  327
0.000000+0 0.000000+0          0          0          0          02631 2  099999
0.000000+0 0.000000+0          0          0          0          02631 0  0     0
2.605600+4 5.545440+1          0          0          0          02631 3  1     1
0.000000+0 0.000000+0          0          0          1     118862631 3  1     2
     11886          2                                            2631 3  1     3
1.000000-5 0.000000+0 8.500000+5 0.000000+0 8.500000+5 1.501370+02631 3  1     4
8.501600+5 1.621200+0 8.502720+5 1.830530+0 8.503280+5 1.991440+02631 3  1     5
8.503840+5 2.266760+0 8.504960+5 3.105180+0 8.505520+5 3.398490+02631 3  1     6
8.506080+5 3.240010+0 8.506640+5 2.706120+0 8.507200+5 2.269630+02631 3  1     7
8.508320+5 1.644860+0 8.508890+5 1.420670+0 8.509450+5 1.409280+02631 3  1     8
8.510010+5 1.224200+0 8.510570+5 1.257110+0 8.512250+5 1.190450+02631 3  1     9
8.515050+5 1.016710+0 8.516740+5 1.147350+0 8.517300+5 1.399960+02631 3  1    10
8.517860+5 1.710880+0 8.518420+5 2.202990+0 8.519540+5 3.723320+02631 3  1    11
8.520100+5 4.412230+0 8.520670+5 4.549740+0 8.521230+5 4.276150+02631 3  1    12
8.522350+5 3.192380+0 8.523470+5 2.571110+0 8.525160+5 1.867240+02631 3  1    13
8.525720+5 1.798760+0 8.526850+5 1.560780+0 8.528530+5 1.583620+02631 3  1    14
8.529100+5 1.417930+0 8.530220+5 1.282760+0 8.530780+5 1.381970+02631 3  1    15
8.531350+5 1.183290+0 8.533600+5 1.115840+0 8.536410+5 9.088520-12631 3  1    16
8.538660+5 1.015650+0 8.539230+5 8.782260-1 8.540350+5 8.654520-12631 3  1    17
8.540920+5 7.611950-1 8.543170+5 7.053160-1 8.543740+5 8.481200-12631 3  1    18
8.544860+5 8.733950-1 8.546560+5 6.495240-1 8.549940+5 6.797920-12631 3  1    19
8.550510+5 7.410150-1 8.551070+5 6.239980-1 8.552770+5 7.379170-12631 3  1    20
8.556150+5 8.505940-1 8.560110+5 1.205440+0 8.561240+5 1.366770+02631 3  1    21
8.561810+5 1.343680+0 8.564640+5 1.917050+0 8.568600+5 2.501140+02631 3  1    22
8.569170+5 2.523750+0 8.573130+5 3.187040+0 8.577670+5 3.452750+02631 3  1    23
8.581640+5 3.595140+0 8.586750+5 3.461960+0 8.589020+5 4.070410+02631 3  1    24
8.590730+5 4.755850+0 8.592430+5 4.043290+0 8.593000+5 3.663900+02631 3  1    25
8.593570+5 3.626010+0 8.594140+5 3.518320+0 8.598120+5 3.168320+02631 3  1    26
8.599260+5 3.156840+0 8.599830+5 3.040850+0 8.600400+5 3.161470+02631 3  1    27
8.600970+5 3.007280+0 8.602680+5 3.095120+0 8.603250+5 2.966430+02631 3  1    28
8.603820+5 3.057550+0 8.605530+5 2.975790+0 8.607810+5 2.898040+02631 3  1    29
8.612370+5 2.614040+0 8.613510+5 2.682170+0 8.614080+5 2.580580+02631 3  1    30
8.615800+5 2.506820+0 8.616370+5 2.566630+0 8.618650+5 2.417590+02631 3  1    31
8.620000+5 2.418840+0 8.621510+5 2.420220+0 8.622700+5 2.622580+02631 3  1    32
```

Brookhaven National Laboratory

CSEWG Document ENDF-102
Report BNL-224854-2023-INRE
Git Revision SHA1: 3576914

ENDF-6 Formats Manual

Data Formats and Procedures for the Evaluated Nuclear Data Files
ENDF/B-VI, ENDF/B-VII and ENDF/B-VIII

Written by the Members of the Cross Sections Evaluation Working Group

Edited by
D. A. Brown

September 28, 2023

National Nuclear Data Center
Brookhaven National Laboratory
Upton, NY 11973-5000
www.nndc.bnl.gov

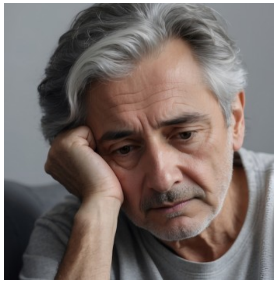
Notice: The work at Brookhaven National Laboratory was sponsored by the Office of Nuclear Physics, Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-98CH10886 with Brookhaven Science Associates, LLC. This work was also upported by the Nuclear Criticality Safety Program, funded and managed by the National Nuclear Security Administration for the Department of Energy.

> 400 pages

The nuclear digital divide

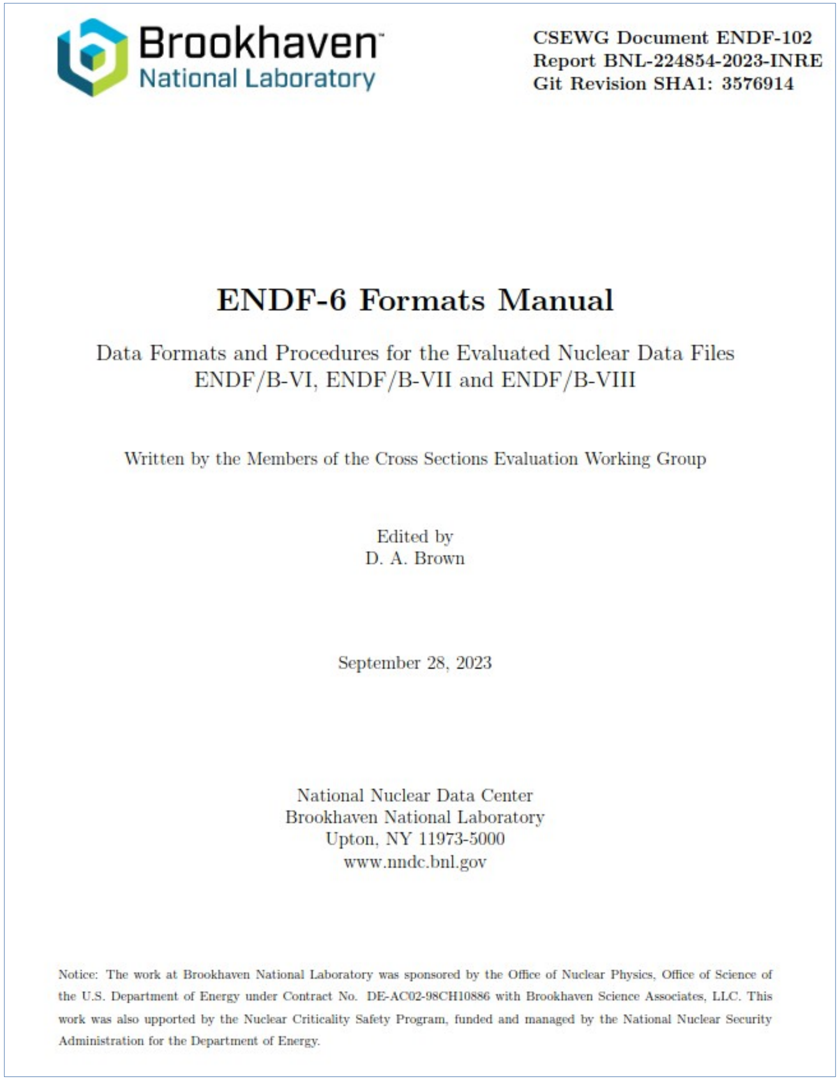
Insiders

Outsiders



Bridging the digital divide

- Read
- Write
- Validate
- Translate
- Everything



Brookhaven National Laboratory

CSEWG Document ENDF-102
Report BNL-224854-2023-INRE
Git Revision SHA1: 3576914

ENDF-6 Formats Manual

Data Formats and Procedures for the Evaluated Nuclear Data Files
ENDF/B-VI, ENDF/B-VII and ENDF/B-VIII

Written by the Members of the Cross Sections Evaluation Working Group

Edited by
D. A. Brown

September 28, 2023

National Nuclear Data Center
Brookhaven National Laboratory
Upton, NY 11973-5000
www.nndc.bnl.gov

Notice: The work at Brookhaven National Laboratory was sponsored by the Office of Nuclear Physics, Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-98CH10886 with Brookhaven Science Associates, LLC. This work was also upported by the Nuclear Criticality Safety Program, funded and managed by the National Nuclear Security Administration for the Department of Energy.

> 400 pages

Bridging the digital divide

- Read
- Write
- Validate
- Translate
- Everything
- Provably correct

How to not mess up?

Brookhaven National Laboratory

CSEWG Document ENDF-102
Report BNL-224854-2023-INRE
Git Revision SHA1: 3576914

ENDF-6 Formats Manual

Data Formats and Procedures for the Evaluated Nuclear Data Files
ENDF/B-VI, ENDF/B-VII and ENDF/B-VIII

Written by the Members of the Cross Sections Evaluation Working Group

Edited by
D. A. Brown

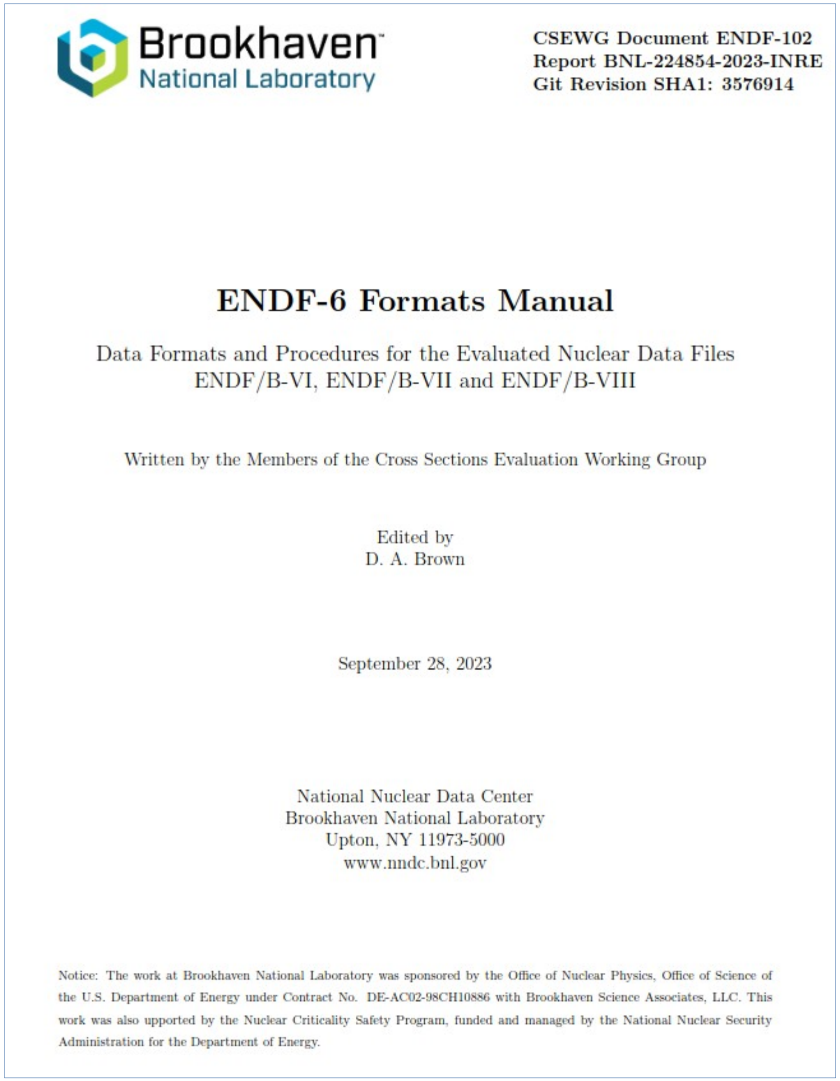
September 28, 2023

National Nuclear Data Center
Brookhaven National Laboratory
Upton, NY 11973-5000
www.nndc.bnl.gov

Notice: The work at Brookhaven National Laboratory was sponsored by the Office of Nuclear Physics, Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-98CH10886 with Brookhaven Science Associates, LLC. This work was also upported by the Nuclear Criticality Safety Program, funded and managed by the National Nuclear Security Administration for the Department of Energy.

> 400 pages

Make manual computer readable



CSEWG Document ENDF-102
Report BNL-224854-2023-INRE
Git Revision SHA1: 3576914

ENDF-6 Formats Manual

Data Formats and Procedures for the Evaluated Nuclear Data Files
ENDF/B-VI, ENDF/B-VII and ENDF/B-VIII

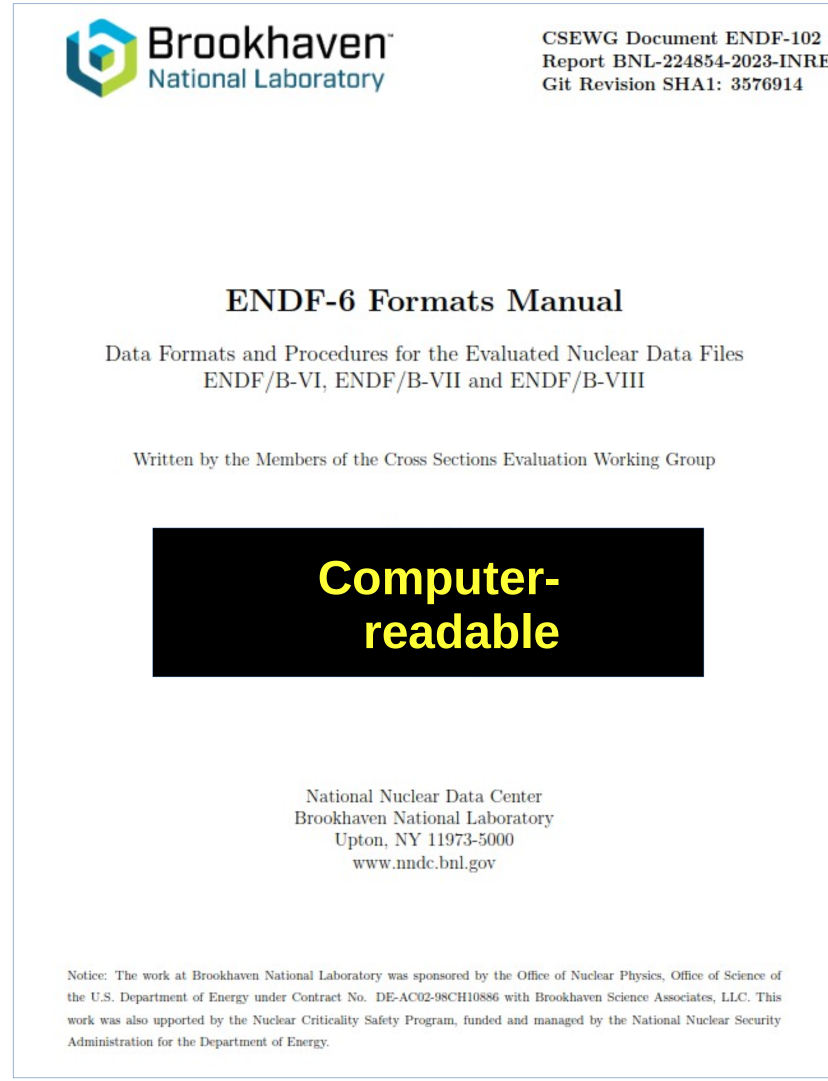
Written by the Members of the Cross Sections Evaluation Working Group

Edited by
D. A. Brown

September 28, 2023

National Nuclear Data Center
Brookhaven National Laboratory
Upton, NY 11973-5000
www.nndc.bnl.gov

Notice: The work at Brookhaven National Laboratory was sponsored by the Office of Nuclear Physics, Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-98CH10886 with Brookhaven Science Associates, LLC. This work was also upported by the Nuclear Criticality Safety Program, funded and managed by the National Nuclear Security Administration for the Department of Energy.



CSEWG Document ENDF-102
Report BNL-224854-2023-INRE
Git Revision SHA1: 3576914

ENDF-6 Formats Manual

Data Formats and Procedures for the Evaluated Nuclear Data Files
ENDF/B-VI, ENDF/B-VII and ENDF/B-VIII

Written by the Members of the Cross Sections Evaluation Working Group

Computer-readable

National Nuclear Data Center
Brookhaven National Laboratory
Upton, NY 11973-5000
www.nndc.bnl.gov

Notice: The work at Brookhaven National Laboratory was sponsored by the Office of Nuclear Physics, Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-98CH10886 with Brookhaven Science Associates, LLC. This work was also upported by the Nuclear Criticality Safety Program, funded and managed by the National Nuclear Security Administration for the Department of Energy.

Basic building blocks

0.6.4.2 CONT Records

The smallest possible record is a control (CONT) record. For convenience, a CONT record is denoted by

```
[MAT,MF,MT/C1,C2,L1,L2,N1,N2]CONT
```

```
 1.000000+0 2.000000+8          3          0         10          34125 1451
```

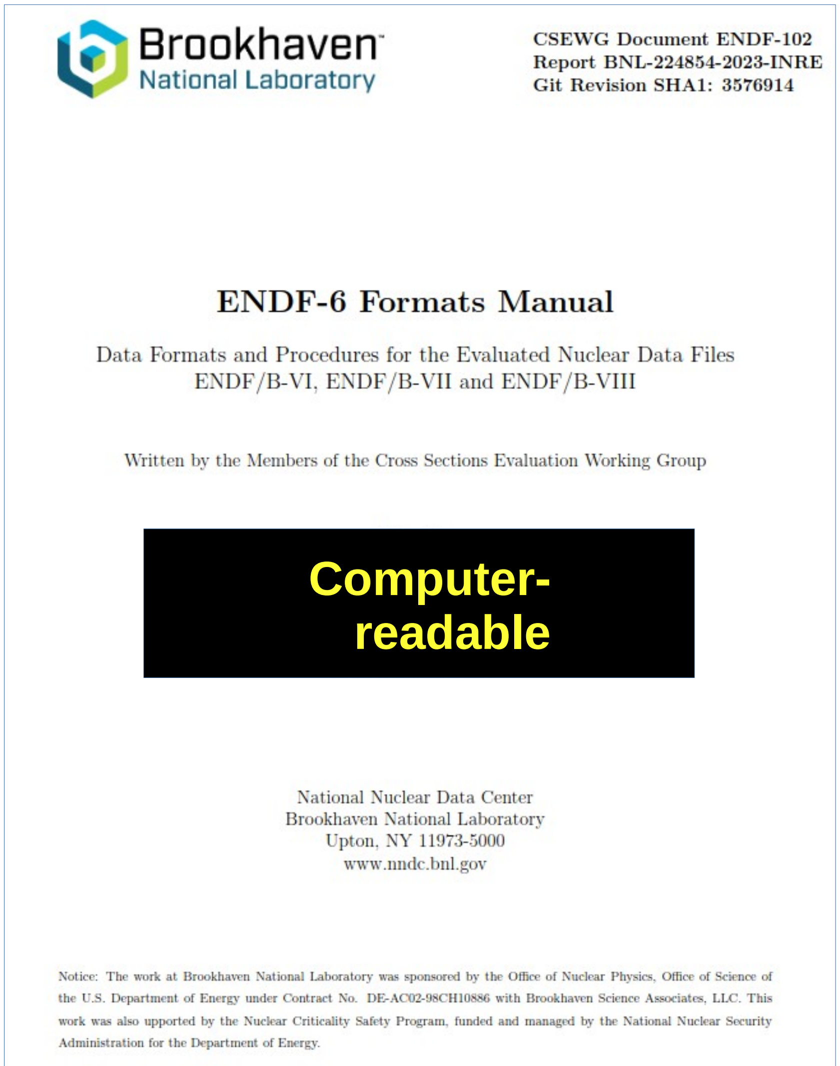
0.6.4.1 TEXT Records

This record is used either as the first entry on an ENDF tape (TPID), or to give the comments in File 1. It is indicated by the following shorthand notation

```
[MAT, MF, MT/ HL] TEXT
```

```
This evaluated data file is based on a software system built       4125 1451
```

Formal description language


Brookhaven National Laboratory

CSEWG Document ENDF-102
Report BNL-224854-2023-INRE
Git Revision SHA1: 3576914

ENDF-6 Formats Manual

Data Formats and Procedures for the Evaluated Nuclear Data Files
ENDF/B-VI, ENDF/B-VII and ENDF/B-VIII

Written by the Members of the Cross Sections Evaluation Working Group

Computer-readable

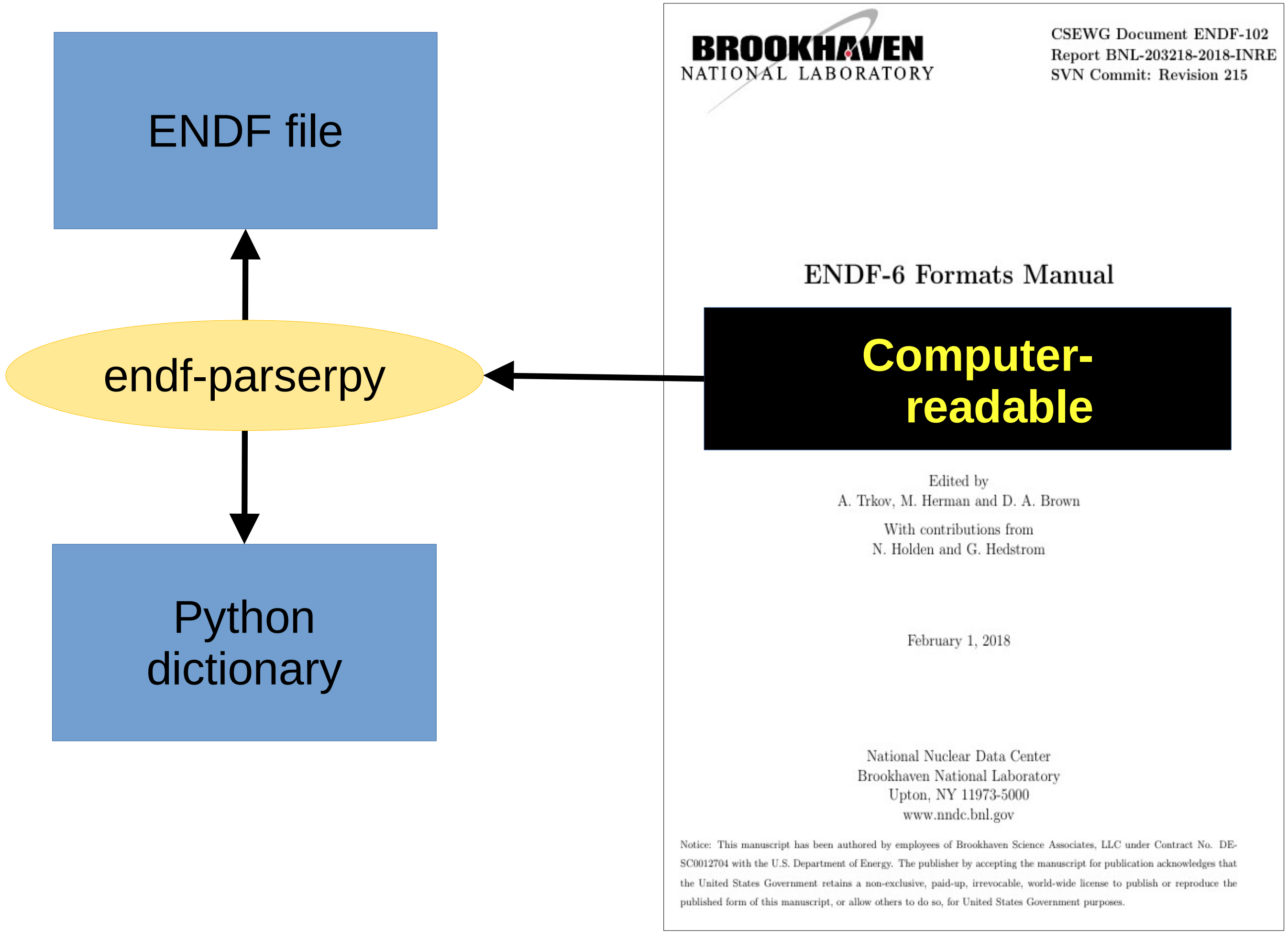
National Nuclear Data Center
Brookhaven National Laboratory
Upton, NY 11973-5000
www.nndc.bnl.gov

Notice: The work at Brookhaven National Laboratory was sponsored by the Office of Nuclear Physics, Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-98CH10886 with Brookhaven Science Associates, LLC. This work was also upported by the Nuclear Criticality Safety Program, funded and managed by the National Nuclear Security Administration for the Department of Energy.

```
[MAT, 4, MT/ ZA, AWR, 0, LTT, 0, 0]HEAD
if LTT==3 and LI==0 [lookahead=1]:
    [MAT, 4, MT/ 0.0, AWR?, LI, LCT, 0, NM]CONT
else:
    [MAT, 4, MT/ 0.0, AWR?, LI, LCT, 0, 0]CONT
endif

# Legendre coefficients
if LTT == 1 and LI == 0:
    [MAT, 4, MT/ 0.0, 0.0, 0, 0, NR, NE/ Eint ]TAB2
    for i=1 to NE:
        [MAT, 4, MT/ T, E[i] , LT, 0, NL[i], 0/ {a[i,l]}{l=1 to NL[i]} ]LIST
    endfor

# Tabulated probability distributions
elif LTT==2 and LI==0:
    [MAT, 4, MT/ 0.0, 0.0, 0, 0, NR, NE/ Eint ]TAB2 (energy_table)
    for i=1 to NE:
        [MAT, 4, MT/ T, E[i] , LT, 0, NR, NP/ mu / f]TAB1 (angtable[i])
    endfor

# Angular distributions over two energy ranges.
elif LTT==3 and LI==0:
    # lower range given by Legendre coefficients
    [MAT, 4, MT/ 0.0, 0.0, 0, 0, NR, NE1/ Eint ]TAB2 (leg_int)
    for i=1 to NE1:
        [MAT, 4, MT/ T, E[i], LT, 0, NL[i], 0/
                     {al[i,j]}{j=1 to NL[i]} ]LIST
    endfor
    # higher range represented by probability distribution
    [MAT, 4, MT/ 0.0, 0.0, 0, 0, NR, NE2/ Eint ]TAB2 (ang_int)
    for i=NE1 to NE1+NE2-1:
        [MAT, 4, MT/ T,  E[i] , LT, 0, NR, NP/ mu / f ]TAB1 (angtable[i])
    endfor
endif
SEND
```


A formal ENDF format description language, ArXiv:2312.08249

Basic design



Using ENDF-6 recipe files

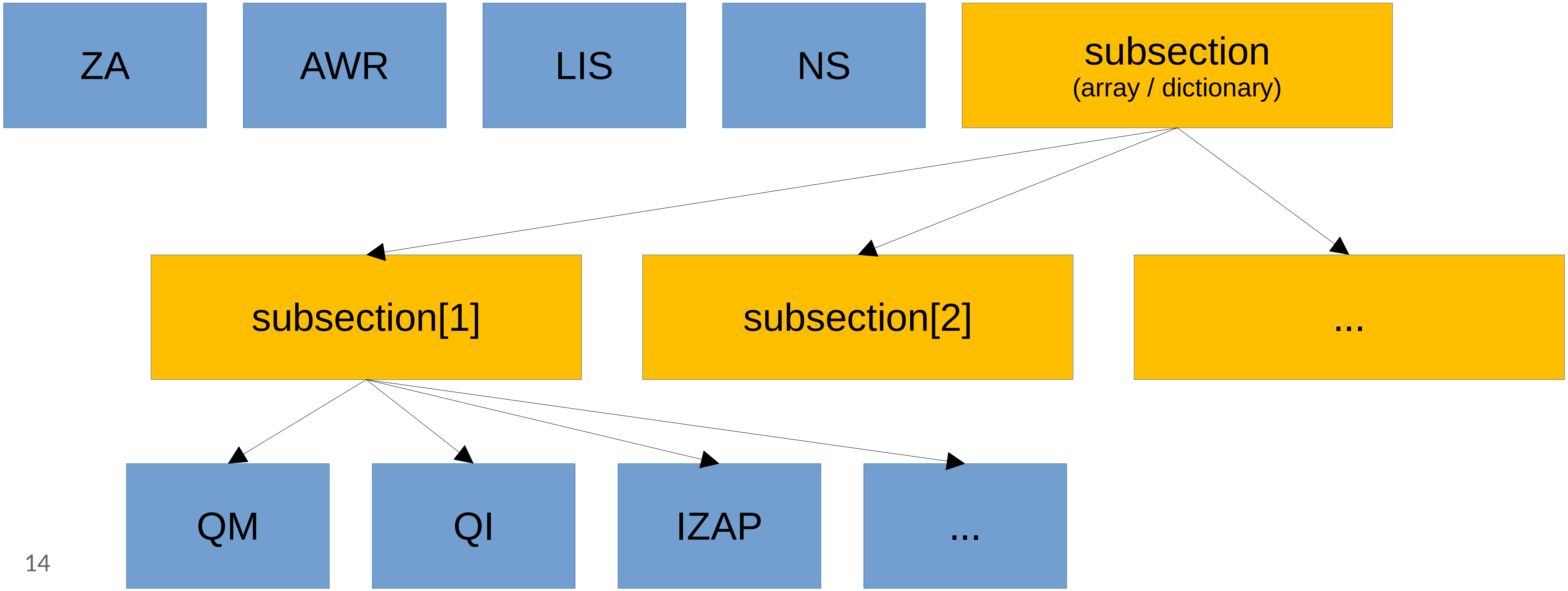
```
[MAT, 1,451/ ZA, AWR, LRP, LFI, NLIB, NMOD]HEAD
[MAT, 1,451/ ELIS, STA, LIS, LISO, 0, NFOR]CONT
[MAT, 1,451/ AWI, EMAX, LREL, 0, NSUB, NVER]CONT
[MAT, 1,451/ TEMP, 0.0, LDRV, 0, NWD, NXC]CONT
for i=1 to NWD:
    [MAT, 1,451/ DESCRIPTION[i]]TEXT
endfor
```

endf-parserpy

```
 2.906300+4 6.238900+1          1          0          0          52925 1451
 0.000000+0 0.000000+0          0          0          0          62925 1451
 1.000000+0 1.500000+8          8          0         10          72925 1451
 0.000000+0 0.000000+0          0          0        481        1152925 1451
 29-Cu- 63 LANL,ORNL  EVAL-FEB98 A.Koning,M.Chadwick,Hetrick       2925 1451
 CH98,CH99            DIST-DEC06 REV4-                20011108     2925 1451
----ENDF/B-VII        MATERIAL 2925         REVISION 4             2925 1451
-----INCIDENT NEUTRON DATA                                         2925 1451
------ENDF-6 FORMAT                                                2925 1451
```

Hierarchical representation

```
[MAT, 10, MT/ ZA, AWR, LIS, 0, NS, 0]HEAD
for k=1 to NS:
(subsection[k])
    [MAT, 10, MT/ QM, QI, IZAP, LFS, NR, NP/ E / sigma ]TAB1
(/subsection[k])
endfor
SEND
```



Defining a formal language

- List of production rules
- Extended Backus-Naur Form (EBNF)

Production rule

```
S : "a" "b" "c"
```

Nonterminal symbol

terminal symbol

“abc”

List of production rules

```
S : "a" "b" "c"
```

start symbol →

```
T : S | "d" "e" "f"
```

“abc”

“def”

List of production rules

```
S : "a" "b" "c"
```

start symbol →

```
T : S | "d" "e" "f"
```

Optional presence **“a”?**

Zero or more repetitions **“a”***

Groups **(“a” “b” “c”)**

Generation of parsing tree

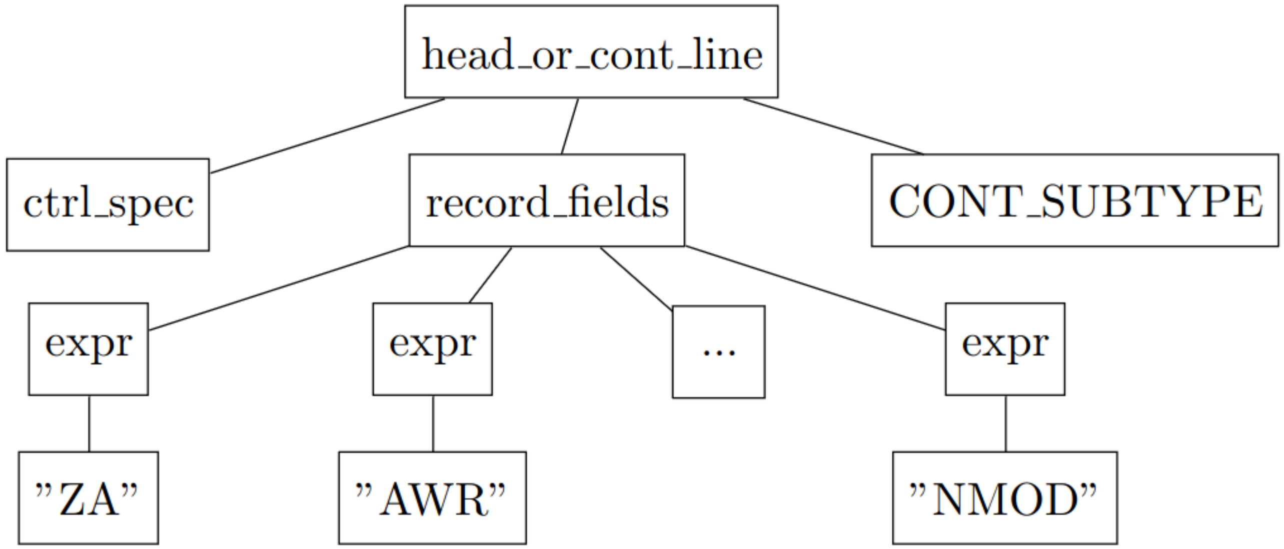
Production rules

```
head_or_cont_line : "[" ctrl_spec "/" record_fields "]" CONT_SUBTYPE
record_fields : expr "," expr "," expr "," expr "," expr "," expr
CONT_SUBTYPE : "CONT" | "HEAD"
```

“Sentence” in formal language

```
[ MAT , 1 ,451/ ZA , AWR , LRP , LFI , NLIB , NMOD ] HEAD
```

Parsing tree



Functioning of endf-parserpy

1) Formal Grammar (production rules) defining formal ENDF description language
2) ENDF-6 recipe files ("computer-readable" ENDF-6 manual)
3) Python package Lark to generate parsing tree based on (1) and (2)
4) endf-parserpy walks through nodes of parsing tree, performing specific actions for mapping low-level ENDF-6 representation to high-level hierarchical representation

https://github.com/lark-parser/lark

https://github.com/iaea-nds/endf-parserpy

Selected examples

Inconsistent variable specification in mathematical expression in ENDF-6 file

```
-------- Line 11 -----------
Template:  [ MAT , 32 , 151 / AJ , PJ , 0 , 0 , 6 * NCH , NCH /
 { PPI [ k ] , L [ k ] , SCH [ k ] , BND [ k ] , APE [ k ] , APT [ k ] } { k = 1 to NCH } ] LIST
Line:      " 5.000000-1 0.000000+0          0          0         18            3 82532151   12"

-------- Line 15 -----------
Template:  [ MAT , 32 , 151 / 0.0 , 0.0 , 0 , NRSA , 12 * NX , NX ? /
 { ER [ k ] , { GAM [ p , k ] } { p = 1 to NCH } PADLINE ,
 DER [ k ] , { DGAM [ p , k ] } { p = 1 to NCH } PADLINE } { k = 1 to NRSA } ] LIST
Line:      " 0.000000+0 0.000000+0          0          4         48            8 82532151   16"

Error message: Expected 4 in the ENDF file but got 8. The value was encountered in a source field named N2
```

$$
\begin{array}{llllll}
[\mathrm{MAT},32,151/\ \mathrm{AJ}, & \mathrm{PJ}, & 0, & 0, & 6{*}\mathrm{NCH}, & \mathrm{NCH}/ \\
\mathrm{PPI}_1, & \mathrm{L}_1, & \mathrm{SCH}_1, & \mathrm{BND}_1, & \mathrm{APE}_1, & \mathrm{APT}_1, \\
\mathrm{PPI}_2, & \mathrm{L}_2, & \mathrm{SCH}_2, & \mathrm{BND}_2, & \mathrm{APE}_2, & \mathrm{APT}_2, \\
\multicolumn{6}{c}{\text{-----------------------------------}} \\
\mathrm{PPI}_{NCH}, & \mathrm{L}_{NCH}, & \mathrm{SCH}_{NCH}, & \mathrm{BND}_{NCH}, & \mathrm{APE}_{NCH}, & \mathrm{APT}_{NCH}]\mathrm{LIST} \\
\multicolumn{6}{c}{\text{-----------------------------------}} \\
[\mathrm{MAT},32,151/0.0, & 0.0, & 0, & \mathrm{NRSA}, & 12{*}\mathrm{NX}, & \mathrm{NX}/ \\
\mathrm{ER}_1, & \mathrm{GAM}_{1,1}, & \mathrm{GAM}_{2,1}, & \mathrm{GAM}_{3,1}, & \mathrm{GAM}_{4,1}, & \mathrm{GAM}_{5,1}, \\
\mathrm{GAM}_{6,1}, & \ldots\ \ldots & \ldots\ \ldots & \ldots\ \ldots & \ldots\ . & ,\ \mathrm{GAM}_{NCH,1},
\end{array}
$$

Wrong/unclear description of MF2/MT151/LRF=7 in ENDF-6 manual

$$
\begin{array}{llllll}
\texttt{[MAT,2,151/ 0.0,} & \texttt{0.0,} & \texttt{0,} & \texttt{NRS,} & \texttt{6*NX,} & \texttt{NX/} \\
\texttt{ER}_1\texttt{,} & \texttt{GAM}_{1,1}\texttt{,} & \texttt{GAM}_{2,1}\texttt{,} & \texttt{GAM}_{3,1}\texttt{,} & \texttt{GAM}_{4,1}\texttt{,} & \texttt{GAM}_{5,1}\texttt{,} \\
\texttt{GAM}_{6,1}\texttt{,} & \texttt{-------------------------} & \texttt{GAM}_{NCH,1}\texttt{,} & & & \\
\texttt{ER}_2\texttt{,} & \texttt{GAM}_{1,2}\texttt{,} & \texttt{GAM}_{2,2}\texttt{,} & \texttt{GAM}_{3,2}\texttt{,} & \texttt{GAM}_{4,2}\texttt{,} & \texttt{GAM}_{5,2}\texttt{,} \\
\texttt{GAM}_{6,2}\texttt{,} & \texttt{--------------------------} & \texttt{GAM}_{NCH,2}\texttt{,} & & & \\
\texttt{---} & & & & & \\
\texttt{ER}_{NRS}\texttt{,} & \texttt{GAM}_{1,NRS}\texttt{,} & \texttt{GAM}_{2,NRS}\texttt{,} & \texttt{GAM}_{3,NRS}\texttt{,} & \texttt{GAM}_{4,NRS}\texttt{,} & \texttt{GAM}_{5,NRS}\texttt{,} \\
\texttt{GAM}_{6,NRS}\texttt{,} & \texttt{------------------------} & \texttt{GAM}_{NCH,NRS} & & & \texttt{]LIST}
\end{array}
$$

Correct description

```
NX := (1+NCH + (5-NCH) % 6) * NRS / 6
num_zeros := (5-NCH) % 6
[MAT,2,151/ 0.0, 0.0, 0, NRS, 6*NX, NX /
    { ER[n], {GAM[m,n]}{m=1 to NCH},
      {0.0}{p=1 to num_zeros} }{n=1 to NRS} ]LIST
```

Provenance checking
FENDL proton sublibrary

#	Material	Source	Emax [MeV]
1	1-H-1	JENDL-1	3000
2	1-H-2	ENDF/B-VII	150
3	1-H-3	ENDF/B-VII	20
4	2-He-3	ENDF/B-VII	20
5	3-Li-6	JENDL-4.0/HE	200
6	3-Li-7	JENDL-4.0/HE	200
7	4-Be-9	ENDF/B-VII	113
8	5-B-10	ENDF/B-VII	3
9	5-B-11	ENDF/B-?????	200

FENDL 3.2b = ENDF/B.VII.0

```
from endf_parserpy.endf_parser import BasicEndfParser
from endf_parserpy.debugging_utils import compare_objects
parser = BasicEndfParser()
fendl_endf = parser.parsefile(fendl_filename)
other_endf = parser.parsefile(other_endffile)
del fendl_endf[1][451]
del other_endf[1][451]
compare_objects(fendl_endf, other_endf, fail_on_diff=False)
```

```
---- difference for MAT 131 ------
Value mismatch at .3.50.QI (-0.76387 vs -763870.0)
Value mismatch at .3.50.QM (-0.76387 vs -763870.0)
Value mismatch at .3.650.QI (-4.0329 vs -4032900.0)
Value mismatch at .3.650.QM (-4.0329 vs -4032900.0)
```

Comparison with ENDF/B-VII.1

Convenient access to variables

- Browse ENDF-6 files like a filesystem
- Create, change and delete variables like files

Sketch of how access should be possible:

```
mkdir -p 33/18
cd 33/18
echo 236.9986 > AWR
echo 2 > subsection[1]/NI
...
```

```python
outendf = EndfDict({})
outendf['33/18'] = {}
p = outendf['33/18']
p['MAT'] = 9437
p['MF'] = 33
p['MT'] = 18
p['ZA'] = 94239.0
p['AWR'] = 236.9986
p['MTL'] = 0
p['NL'] = 1
p['subsection[1]/XMF1'] = 0.0
p['subsection[1]/XLFS1'] = 0.0
p['subsection[1]/MAT1'] = 0
p['subsection[1]/MT1'] = 18
p['subsection[1]/NC'] = 0
p['subsection[1]/NI'] = 1
p['subsection[1]/ni_subsection[1]/LS'] = 1
p['subsection[1]/ni_subsection[1]/LB'] = 5
NE = len(midE)
p['subsection[1]/ni_subsection[1]/NE'] = NE
p['subsection[1]/ni_subsection[1]/E'] = \
    {k+1: float(midE[k] * 1e6) for k in range(len(midE))}
p['subsection[1]/ni_subsection[1]/F'] = {}
F = p['subsection[1]/ni_subsection[1]/F']
for k in range(1, len(midE)):
    for kp in range(k, len(midE)):
        F[k, kp] = float(rel_cov_lib[k-1, kp-1])
```

“Create filesystem”

```
outendf = EndfDict({})
outendf['33/18'] = {}
p = outendf['33/18']
p['MAT'] = 9437
p['MF'] = 33
p['MT'] = 18
p['ZA'] = 94239.0
p['AWR'] = 236.9986
p['MTL'] = 0
p['NL'] = 1
p['subsection[1]/XMF1'] = 0.0
p['subsection[1]/XLFS1'] = 0.0
p['subsection[1]/MAT1'] = 0
p['subsection[1]/MT1'] = 18
p['subsection[1]/NC'] = 0
p['subsection[1]/NI'] = 1
p['subsection[1]/ni_subsection[1]/LS'] = 1
p['subsection[1]/ni_subsection[1]/LB'] = 5
NE = len(midE)
p['subsection[1]/ni_subsection[1]/NE'] = NE
p['subsection[1]/ni_subsection[1]/E'] = \
    {k+1: float(midE[k] * 1e6) for k in range(len(midE))}
p['subsection[1]/ni_subsection[1]/F'] = {}
F = p['subsection[1]/ni_subsection[1]/F']
for k in range(1, len(midE)):
    for kp in range(k, len(midE)):
        F[k, kp] = float(rel_cov_lib[k-1, kp-1])
```

mkdir -p 33/18

```
outendf = EndfDict({})
outendf['33/18'] = {}
p = outendf['33/18']
p['MAT'] = 9437
p['MF'] = 33
p['MT'] = 18
p['ZA'] = 94239.0
p['AWR'] = 236.9986
p['MTL'] = 0
p['NL'] = 1
p['subsection[1]/XMF1'] = 0.0
p['subsection[1]/XLFS1'] = 0.0
p['subsection[1]/MAT1'] = 0
p['subsection[1]/MT1'] = 18
p['subsection[1]/NC'] = 0
p['subsection[1]/NI'] = 1
p['subsection[1]/ni_subsection[1]/LS'] = 1
p['subsection[1]/ni_subsection[1]/LB'] = 5
NE = len(midE)
p['subsection[1]/ni_subsection[1]/NE'] = NE
p['subsection[1]/ni_subsection[1]/E'] = \
    {k+1: float(midE[k] * 1e6) for k in range(len(midE))}
p['subsection[1]/ni_subsection[1]/F'] = {}
F = p['subsection[1]/ni_subsection[1]/F']
for k in range(1, len(midE)):
    for kp in range(k, len(midE)):
        F[k, kp] = float(rel_cov_lib[k-1, kp-1])
```

cd 33/18

```
outendf = EndfDict({})
outendf['33/18'] = {}
p = outendf['33/18']
p['MAT'] = 9437
p['MF'] = 33
p['MT'] = 18
p['ZA'] = 94239.0
p['AWR'] = 236.9986
p['MTL'] = 0
p['NL'] = 1
p['subsection[1]/XMF1'] = 0.0
p['subsection[1]/XLFS1'] = 0.0
p['subsection[1]/MAT1'] = 0
p['subsection[1]/MT1'] = 18
p['subsection[1]/NC'] = 0
p['subsection[1]/NI'] = 1
p['subsection[1]/ni_subsection[1]/LS'] = 1
p['subsection[1]/ni_subsection[1]/LB'] = 5
NE = len(midE)
p['subsection[1]/ni_subsection[1]/NE'] = NE
p['subsection[1]/ni_subsection[1]/E'] = \
    {k+1: float(midE[k] * 1e6) for k in range(len(midE))}
p['subsection[1]/ni_subsection[1]/F'] = {}
F = p['subsection[1]/ni_subsection[1]/F']
for k in range(1, len(midE)):
    for kp in range(k, len(midE)):
        F[k, kp] = float(rel_cov_lib[k-1, kp-1])
```

Create files named MAT, MF, MT, ... with specific content

```
outendf = EndfDict({})
outendf['33/18'] = {}
p = outendf['33/18']
p['MAT'] = 9437
p['MF'] = 33
p['MT'] = 18
p['ZA'] = 94239.0
p['AWR'] = 236.9986
p['MTL'] = 0
p['NL'] = 1
p['subsection[1]/XMF1'] = 0.0
p['subsection[1]/XLFS1'] = 0.0
p['subsection[1]/MAT1'] = 0
p['subsection[1]/MT1'] = 18
p['subsection[1]/NC'] = 0
p['subsection[1]/NI'] = 1
p['subsection[1]/ni_subsection[1]/LS'] = 1
p['subsection[1]/ni_subsection[1]/LB'] = 5
NE = len(midE)
p['subsection[1]/ni_subsection[1]/NE'] = NE
p['subsection[1]/ni_subsection[1]/E'] = \
    {k+1: float(midE[k] * 1e6) for k in range(len(midE))}
p['subsection[1]/ni_subsection[1]/F'] = {}
F = p['subsection[1]/ni_subsection[1]/F']
for k in range(1, len(midE)):
    for kp in range(k, len(midE)):
        F[k, kp] = float(rel_cov_lib[k-1, kp-1])
```

Create files named XMF1, XLFS1, ... in a subdirectory named subsection[1]

```
outendf = EndfDict({})
outendf['33/18'] = {}
p = outendf['33/18']
p['MAT'] = 9437
p['MF'] = 33
p['MT'] = 18
p['ZA'] = 94239.0
p['AWR'] = 236.9986
p['MTL'] = 0
p['NL'] = 1
p['subsection[1]/XMF1'] = 0.0
p['subsection[1]/XLFS1'] = 0.0
p['subsection[1]/MAT1'] = 0
p['subsection[1]/MT1'] = 18
p['subsection[1]/NC'] = 0
p['subsection[1]/NI'] = 1
p['subsection[1]/ni_subsection[1]/LS'] = 1
p['subsection[1]/ni_subsection[1]/LB'] = 5
NE = len(midE)
p['subsection[1]/ni_subsection[1]/NE'] = NE
p['subsection[1]/ni_subsection[1]/E'] = \
    {k+1: float(midE[k] * 1e6) for k in range(len(midE))}
p['subsection[1]/ni_subsection[1]/F'] = {}
F = p['subsection[1]/ni_subsection[1]/F']
for k in range(1, len(midE)):
    for kp in range(k, len(midE)):
        F[k, kp] = float(rel_cov_lib[k-1, kp-1])
```

```python
outendf = EndfDict({})
outendf['33/18'] = {}
p = outendf['33/18']
p['MAT'] = 9437
p['MF'] = 33
p['MT'] = 18
p['ZA'] = 94239.0
p['AWR'] = 236.9986
p['MTL'] = 0
p['NL'] = 1
p['subsection[1]/XMF1'] = 0.0
p['subsection[1]/XLFS1'] = 0.0
p['subsection[1]/MAT1'] = 0
p['subsection[1]/MT1'] = 18
p['subsection[1]/NC'] = 0
p['subsection[1]/NI'] = 1
p['subsection[1]/ni_subsection[1]/LS'] = 1
p['subsection[1]/ni_subsection[1]/LB'] = 5
NE = len(midE)
p['subsection[1]/ni_subsection[1]/NE'] = NE
p['subsection[1]/ni_subsection[1]/E'] = \
    {k+1: float(midE[k] * 1e6) for k in range(len(midE))}
p['subsection[1]/ni_subsection[1]/F'] = {}
F = p['subsection[1]/ni_subsection[1]/F']
for k in range(1, len(midE)):
    for kp in range(k, len(midE)):
        F[k, kp] = float(rel_cov_lib[k-1, kp-1])
```

Create empty covariance matrix

Define shortcut for variable F in ENDF-6 hierarchy

```
outendf = EndfDict({})
outendf['33/18'] = {}
p = outendf['33/18']
p['MAT'] = 9437
p['MF'] = 33
p['MT'] = 18
p['ZA'] = 94239.0
p['AWR'] = 236.9986
p['MTL'] = 0
p['NL'] = 1
p['subsection[1]/XMF1'] = 0.0
p['subsection[1]/XLFS1'] = 0.0
p['subsection[1]/MAT1'] = 0
p['subsection[1]/MT1'] = 18
p['subsection[1]/NC'] = 0
p['subsection[1]/NI'] = 1
p['subsection[1]/ni_subsection[1]/LS'] = 1
p['subsection[1]/ni_subsection[1]/LB'] = 5
NE = len(midE)
p['subsection[1]/ni_subsection[1]/NE'] = NE
p['subsection[1]/ni_subsection[1]/E'] = \
    {k+1: float(midE[k] * 1e6) for k in range(len(midE))}
p['subsection[1]/ni_subsection[1]/F'] = {}
F = p['subsection[1]/ni_subsection[1]/F']
for k in range(1, len(midE)):
    for kp in range(k, len(midE)):
        F[k, kp] = float(rel_cov_lib[k-1, kp-1])
```

```python
outendf = EndfDict({})
outendf['33/18'] = {}
p = outendf['33/18']
p['MAT'] = 9437
p['MF'] = 33
p['MT'] = 18
p['ZA'] = 94239.0
p['AWR'] = 236.9986
p['MTL'] = 0
p['NL'] = 1
p['subsection[1]/XMF1'] = 0.0
p['subsection[1]/XLFS1'] = 0.0
p['subsection[1]/MAT1'] = 0
p['subsection[1]/MT1'] = 18
p['subsection[1]/NC'] = 0
p['subsection[1]/NI'] = 1
p['subsection[1]/ni_subsection[1]/LS'] = 1
p['subsection[1]/ni_subsection[1]/LB'] = 5
NE = len(midE)
p['subsection[1]/ni_subsection[1]/NE'] = NE
p['subsection[1]/ni_subsection[1]/E'] = \
    {k+1: float(midE[k] * 1e6) for k in range(len(midE))}
p['subsection[1]/ni_subsection[1]/F'] = {}
F = p['subsection[1]/ni_subsection[1]/F']
for k in range(1, len(midE)):
    for kp in range(k, len(midE)):
        F[k, kp] = float(rel_cov_lib[k-1, kp-1])
```

Save covariance matrix in hierarchical ENDF-6 structure

```python
p['MAT'] = 9437
p['MF'] = 33
p['MT'] = 18
p['ZA'] = 94239.0
p['AWR'] = 236.9986
p['MTL'] = 0
p['NL'] = 1
p['subsection[1]/XMF1'] = 0.0
p['subsection[1]/XLFS1'] = 0.0
p['subsection[1]/MAT1'] = 0
p['subsection[1]/MT1'] = 18
p['subsection[1]/NC'] = 0
p['subsection[1]/NI'] = 1
p['subsection[1]/ni_subsection[1]/LS'] = 1
p['subsection[1]/ni_subsection[1]/LB'] = 5
NE = len(midE)
p['subsection[1]/ni_subsection[1]/NE'] = NE
p['subsection[1]/ni_subsection[1]/E'] = \
    {k+1: float(midE[k] * 1e6) for k in range(len(midE))}
p['subsection[1]/ni_subsection[1]/F'] = {}
F = p['subsection[1]/ni_subsection[1]/F']
for k in range(1, len(midE)):
    for kp in range(k, len(midE)):
        F[k, kp] = float(rel_cov_lib[k-1, kp-1])
```

```python
parser.writefile('pu9_nf_mf33.endf', outendf, overwrite=True)
```

Summary

- endf-parserpy: read, write, validate, compare ENDF-6 files
- Translation to other formats easy (e.g. to JSON)
- endf-6 files can be browsed similar to a filesystem
- Using a formal grammar to define a format specification language (or sometimes the format itself) is a powerful approach
- For example, derived parsing tree can be used to provide guidance to user what variables are expected
- Higher-level functionality (e.g. various interpolation schemes) will be added in the future