General-purpose “EXFOR-driver”: tasks, approach and implementation

by Viktor Zerkin
~ nuclear data software developer ~

Technical Meeting NRDC-2025 of the International Network of Nuclear Reaction Data Centres,
Polytechnic University of Madrid, Madrid, Spain + WebEx, 17 - 20 June 2025

Before any development

Steps to go.

1. Understand/formulate problem and possibly task(s)
2. Three questions to answer:
   a) What already exists and can be used?
   b) What is wrong now?
   c) What do we want to achieve? (define ideal goal)
3. Preparations:
   a) discuss/define main ideas, concept and possible technologies
   b) study and test technologies, select technology
4. Plan: split problem to tasks/sub-tasks, define dates
5. Implementation: development loop

Recent tendencies in ND

1. *More Python*
2. *More JSON for data exchange, operations and storage*
3. *More "open source" for codes and data files, more Git*
4. *Many EXFOR parsers and utility-codes in Python*

What is wrong now?

EXFOR code systems:

1. *Existing codes: Fortran: 2, C: 1, Java:2, Python: 6*
2. *Fortran, C, Java codes: functioning long time, well tested, comprehensive, advanced functionality - produce C5, X5, JSON, XML, X4Pro accessible via Web-GUI and API with Python-examples, but difficult to extend for new tasks by new people*
3. *Most of the systems were built for specific purposes*
4. *Systems are built independently without common approach*
5. *Some codes are very large and specialized*
6. *Python codes not yet cover all EXFOR data types and advance tasks (SG50, SG54)*

Important

There is no general-purpose, low-level, simple "EXFOR-driver" which can be used as common basis for other software built on top of it

1. Translation EXFOR to JSON.

Tasks, basic principles, current limits

Tasks for EXFOR parser:

1. *Step-1: small, general-purpose, low-level EXFOR parser (X4-Driver); not connected to Dictionaries*
2. *To be extended in near future by a new functionality*
3. *To be able translate EXFOR to JSON and back*
4. *Should work on any EXFOR file from single ENTRY to full Master file*
5. *Should allow sequential parsing of a file and full-file parsing to DOM (document object model)*
6. *Q: Should work with old EXFOR files? (support VECTOR-COMMON formalism)*

Basic principles:

1. *Languages: Python and JavaScript**
2. *Based on OOP (Object Oriented Programming), i.e. classes/objects (structures with methods)*
3. *JSON is produced from an object by a method in a class*
4. *Every class in single file with self-test*

* *Structures and algorithms are based on existing X4-Java codes*

Current limitations:

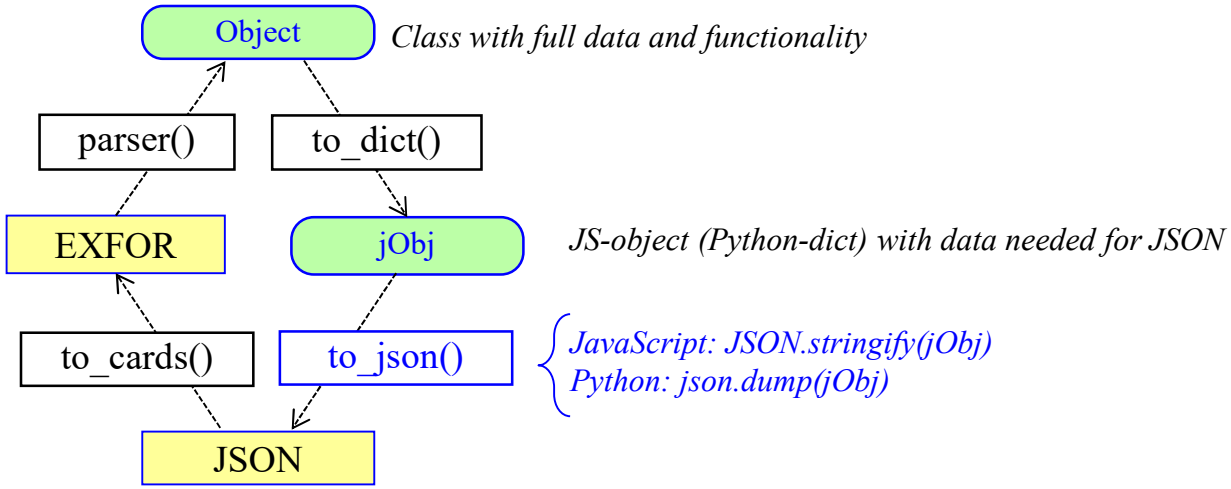
1. *Work on valid EXFOR file, i.e. now not oriented to work with incorrect EXFOR files*
2. *Requires ENTRY+ENDENTRY lines (Q: change to SUBENT+ENDSUBENT?)*

About “basic principles”

Why OOP?

Object Oriented Programming using classes/objects having data structures with methods:
preparing for further extensions beyond the task “translate EXFOR to JSON”

JSON is produced from an object by a method in a class



Every class in single file with self-test
this is how I was writing X4java package, keeping tests inside class-file

General approach

Having experience with programming EXFOR, CINDA, Dictionaries, ENDF, ENSDF-Editor in Java + JSON + JavaScript/Html, I decided to try to rewrite basic low-level part of parsers in Python and JavaScript and see how difficult it would be.

2. Technologies and current implementation

JSON-Tree Editor (2024-2025)

1. *Initial main goal: develop/discuss JSON formats of nuclear data*
2. *Language: JavaScript; works in Web-Browser locally and remotely*
3. *Presents any JSON file as interactive tree*
4. *Provides specialized view for data in X5, ENDF, ENSDF, NSR, IBANDL, MyEnsdf, etc.*
5. *Provides "classic" operations: edit/delete/move/copy/paste text and nodes, new/open/save files, undo/redo, history of operations, return back to any step, etc.*

EXFOR-CINDA-X4TOC5 Dictionary parser (2024)

1. *Language: Python*
2. *Using elements of meta-programming describing input format*
3. *Parsing Dictionaries to classes/objects (~clone of x4dicts.java reproducing functionality)*
4. *Production of JSON file for whole Dictionary release + internal X4toC5 Dictionaries*

ENSDF parser (2024)

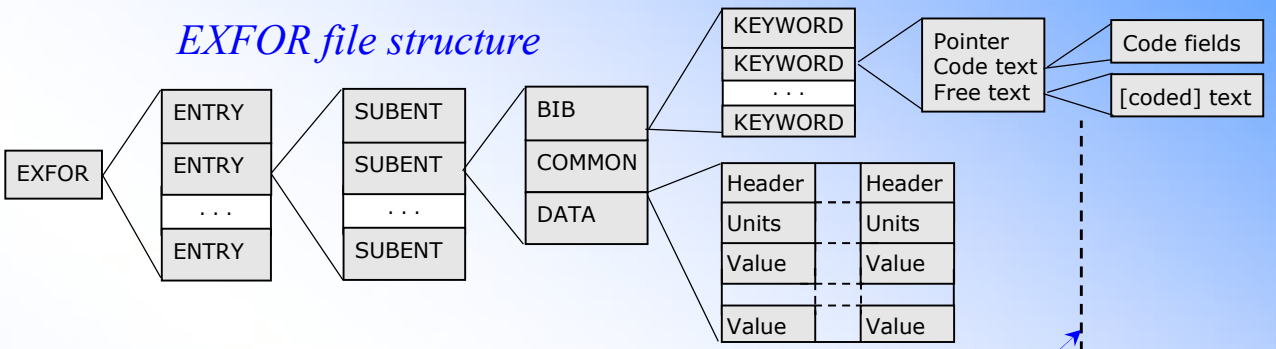
1. *Languages: Python and JavaScript using myMeta-programming approach*
2. *Translating ENSDF file to JSON strictly according to ENSDF-Manual*
3. *Translating JSON to ENSDF and comparison with original*
4. *Web interface (JS only): ENSDF <=> JSON => JSON-Tree Editor*
5. *Command line interface (CLI): Python only*

Low-level EXFOR parser: x4py, x4js (2025)

1. *Implemented on Python and JavaScript*
   1) *Python3. Package x4py: 1.1K lines (10 files, 70-200 lines each)*
   2) *JavaScript-ES5. Module x4dom.js (1K lines), json2x4cards/csv (600 lines)*
2. *Translating EXFOR text to JSON*
3. *Translating JSON to EXFOR and CSV*
4. *Web interface via Web-browser (online, offline): EXFOR <=> JSON => JSON-Tree Editor*
5. *Command line interface (CLI): Python and Node.JS*

Data structures, objects, flows

EXFOR file structure



Exactly follows EXFOR file structure

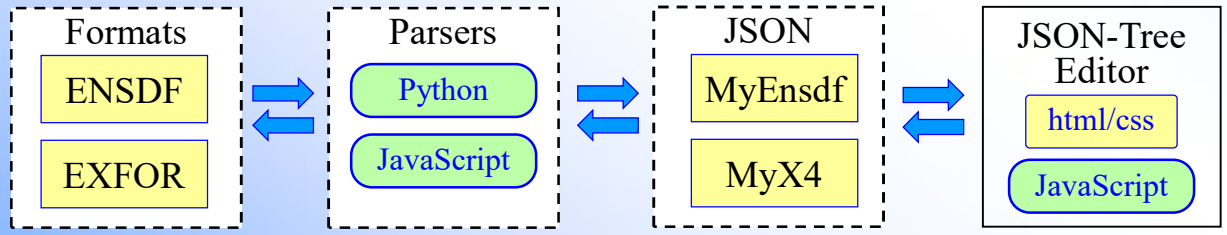
Low-level EXFOR-Driver parser objects

x4dom
↓
x4file → x4entry[] → x4subent[] → x4bib, x4common, x4data
x4bib →x4kw[] →x4code[] →pointer, codeLines[], freeLines[]
x4common, x4data → headers[], pointers[], units[], data[][]

Because now X4-Driver parses file up to this border

Why "low-level"?

Main blocks and data flows



Why JavaScript?

Web-interface
1) Local (offline)
2) Remote (online)

Only Browser is needed
Easy to use/discuss structures

Parser: package/modules can be used to build Desktop and Web Applications

Command line interface (CLI)

1) Python3

```
$ python x4dom.py mm03.x4
$ python x4update.py EXFOR-2023-06-30.bck trans.4213 -o:out1.bck
```

2) Node.JS

```
$ node x4tojs1 mm03.x4
```

ENSDF parser on Python and JavaScript

Java

Codes in Web-tools MyEnsdf ~16K lines

Python

parser() + to_json() ~1.1K ln, json2ensdf: 338 lines

JavaScript

parser() + to_json() ~1.6K ln; json2ensdf: 413 lines

Web-App: ENSDF ⇔ JSON

Translate ENSDF to MyEnsdf-JSON.
by V.Zerkin, 2025-03-08

ENSDF lines:4,597 bytes:372,276 *Examples:[1][2][3][4] [upload][clear][copy][paste][*]*

```
177LU    ADOPTED LEVELS, GAMMAS                                    19NDS    201909
177LU  H TYP=FUL$AUT=F.G. KONDEV$CIT=NDS 159, 1 (2019)$CUT=30-Aug-2019$
177LU  Q 496.8     8 7072.89 166181.5  121447    5     2017WA10
```

Translate >> | indent: 1 | #4 OK | Translate back << | ☑ compact continuation records | ☑ new style of comment: SYM$ | ☐ use dot as flag in field 6 continuation records | #

JSON datasets:8 lines:22,214 bytes:509,550 *[upload][clear][copy][paste][mini][midi][maxi]*

```
{"format": "ens2json-zv1.0.0",
 "now": "2025-03-12T23:54:51Z",
 "title": "ENSDF file in JSON",
 "program": "ens2json.js, by V.Zerkin, Vienna, ver.2025-03-06",
 "purpose": "Interpretation ENSDF to JSON strictly according to ENSDF Manual",
 "nuclides": [
  {"NUCID": "177LU",
   "datasets": [
    {
     "header": {
      "type": "adopted",
      "NUCID": "177LU",
      "DSID": "ADOPTED LEV
      "PUB": " 19NDS",
      "DATE": "201909"
     },
     "history": [
      {"TYP": "FUL",
       "AUT": "F.G. KONDEV
       "CIT": "NDS 159, 1
       "CUT": "30-Aug-2019
      }
     ],
     "Q-values": [
      {"Q": "496.8",
       "DQ": "8",
       "SN": "7072.89",
       "DSN": "16",
       "SP": "6181.5",
       "DSP": "12",
       "QA": "1447",
       "DQA": "5",
       "QREF": "2017WA10"
      }
     ],
     "crossReferences": [
      {"NUCID": "177LU",
       "DSSYM": "C",
       "DSID": "177LU IT D
      },
```

ENSDF-Manual

3 The Q-value Record

Required for adopted data sets.
If there is only one data set for the nuclide then the Q-value record should be given in that data set.
Must precede **L, G, B, E, A, DP** *records.*
If signs are not given, they will be assumed to be +.

Field (Col.)	Name	Description	Reference
1-5	**NUCID**	Nuclide identification	**V.1**
6		Blank	
7		Must be blank	
8	**Q**	Letter 'Q' is required	
9		Must be blank	
10-19	**Q⁻**	Total energy (keV) available for β^- decay of the ground state. ($\mathbf{Q^- > 0}$ if β^- decay is energetically possible. $\mathbf{Q^-} < 0$ represents the Q_ϵ energy of the Z+1 (Z = proton number) isobar.)	**V.10**
20-21	**DQ⁻**	Standard uncertainty in **Q⁻**	**V.11**
22-29	**SN**	Neutron separation energy in keV	**V.10**
30-31	**DSN**	Standard uncertainty in **SN**	**V.11**
32-39	**SP**	Proton separation energy in keV	**V.10**
40-41	**DSP**		
42-49	**QA**		
50-55	**DQA**		
56-80	**QREF**		

Descriptor

```
,"Q-VALUE": {
  "title": "Q-value Record",
  "notes": [
     "Required for adopted data sets.",
     "If there is only one data set for the nuclide then the Q-value r
     "Must precede L, G, B, E, A, DP records."
   ],
  "fields": [
    ["#NUCID",   1,   5, "txt" ,"Nuclide identifcation"            ],
    ["#RTYPE",   8,   8, "Q"   , "Letter 'Q' is required"         ],
    ["Q"     ,  10,  19, "txt" ,"Total energy (keV) available for b- d
    ["DQ"    ,  20,  21, "txt" ,"Standard uncertainty in Q"       ],
    ["SN"    ,  22,  29, "txt" ,"Neutron separation energy in keV"  ],
    ["DSN"   ,  30,  31, "txt" ,"Standard uncertainty in SN"      ],
    ["SP"    ,  32,  39, "txt" ,"Proton separation energy in keV"  ],
    ["DSP"   ,  40,  41, "txt" ,"Standard uncertainty in SP"      ],
    ["QA"    ,  42,  49, "txt" ,"Total energy (keV) available for alph
    ["DQA"   ,  50,  55, "txt" ,"Standard uncertainty in QA"      ],
    ["QREF"  ,  56,  80, "txt" ,"Reference citation(s) for the Q-value
  ]
}
```

JSON-Tree editor

zerkin.usite.pro/edit-json-tree/#5n

File Edit View History Tools Help About myensdf-json example 2025-03-12,23:58

- 177Lu //zv2025 {6}
  - format: ens2json-zv1.0.0
  - now: 2025-02-27T17:15:48.288Z
  - title: ENSDF file in JSON
  - program: ens2json, by V.Zerkin, Vienna, ver.2025-02-27
  - purpose: Interpretation ENSDF to JSON strictly according to ENSDF Manual
  - nuclides [1]
    - nuclides[0] {2} Nuclide:177LU Date:201909 Datasets:8
      - NUCID: 177LU
      - datasets [8] Datasets
        - datasets[0] {7} /A/ Dataset:ADOPTED LEVELS, GAMMAS F.G. Kondev
          - header {5}
          - history [1]
          - Q-values [1]
            - Q-values[0] {9} Q-value:496.8keV SN:7072.89keV SP:6181.5keV QA
              - Q: 496.8 #Total energy (keV) available for b- decay of the groun
              - DQ: 8 #Standard uncertainty in Q
              - SN: 7072.89 #Neutron separation energy in keV
              - DSN: 16 #Standard uncertainty in SN
              - SP: 6181.5 #Proton separation energy in keV
              - DSP: 12 #Standard uncertainty in SP
              - QA: 1447 #Total energy (keV) available for alpha decay of the gr
              - DQA: 5 #Standard uncertainty in QA
              - QREF: 2017WA10 #Reference citation(s) for the Q-values
          - crossReferences [7]
          - Comment-L [33]
          - Comment-G [8]
          - levels [192] Nuclear levels
        - datasets[1] {10} /D/ Dataset:177YB B- DECAY F.G. Kondev Levels:17(β:1
        - datasets[2] {9} /D/ Dataset:177LU IT DECAY (160.4 D) F.G. Kondev Leve
        - datasets[3] {5} /R/ Dataset:176YB(3HE,D),(A,T) F.G. Kondev Levels:36

Use on text-area: <Ctrl/ ...urrect cursor position)

MyEnsdf-JSON: created by Vik
Last updated: 2025-03-11 21:

Colored ENSDF with built-in record editor and JSON-Tree editor

ENSDF ⇔ JSON

JSON ⇔ JSON-Tree editor

Translate ENSDF to MyEnsdf.json

by V.Zerkin, 2025-06-15 /under development/

ENSDF Examples:[1][2][3][4] Do:[upload][clear][copy][paste] grid

```
....|...10....|...20....|...30....|...40....|...50....|...60....|...70....|...80
123456789|123456789|123456789|123456789|123456789|123456789|123456789|123456789|
177LUS G  KC=0.046 10$LC=0.0073 9$MC=0.00166 19
177LUS G  NC=0.00039 5$OC=5.7E-5 8$PC=3.3E-6 8
177LU cG M,MR$|a(K)exp=0.051 {I14} (1996Pe05). Other: |a(K)exp=0.049 (1971Ma45).
177LU  G 552.102   4   74     7 M1+E2     1.8     5     0.019  3
177LU2 G FL=0.0
177LUS G  KC=0.0156 24$LC=0.0028 3$MC=0.00064 6
177LUS G  NC=0.000150 14$OC=2.15E-5 22$PC=1.09E-6 19
177LU cG M,MR$|a(K)exp=0.015 {I4} and |a(L1)exp=0.0022 {I9} (1996Pe05).
177LU2cG Other: |a(K)exp=0.028 (1971Ma45).
177LU  L 569.6721  15   1/2+          155 US     7                        DM2
177LUX L XREF=AE(*576)F(*574)G
177LU cL J$L=0+2 for 1/2+, 3/2+ doublet in {+176}Yb({+3}He,d),(|a,t); 111.715|g
177LU2cL E2 to 5/2+; configuration assignment.
177LU cL T$Weighted average of 150 |ms {I10} (1970F109) and 160 |ms {I10}
177LUxcL (1965He06)
177LU cL CONF$|p1/2[411] (d{-3/2}) Nilsson configuration.
177LU2cL Based on the observed in-band properties, such as alignment
177LU3cL and large signature splitting, and systematics of structures
177LU4cL in neighboring nuclei.
177LU  G 111.715   1  100    10E2                    2.20
177LU2 G FL=457.9568
177LUS G  KC=0.774 11$LC=1.087 16$MC=0.269 4
177LUS G  NC=0.0619 9$OC=0.00752 11$PC=4.07E-5 6
177LU cG M$|a(K)exp=1.0 {I4}, |a(L1)exp=0.39 {I15}, |a(L2)exp=0.7 {I3},
177LU2cG |a(L3)exp=0.8 {I3}, and |a(M3)exp=0.15 {I6} (1996Pe05).
177LU3cG Other: |a(K)exp=0.96, |a(L1)exp=0.1, |a(L2)exp=0.75, |a(L3)exp=0.67,
177LU4cG |a(M)exp=0.42 and |a(N)exp=0.13 (1971Ma45).
177LUB G BE2W=0.00110 5
177LU  G    569.680 9  2.1    5 [M3]                  0.235
177LU2 G FL=0.0
177LUS G  KC=0.183 3$LC=0.0395 6$MC=0.00930 13
177LUS G  NC=0.00220 3$OC=0.000319 5$PC=1.753E-5 25
177LUB G BM3W=0.14 4
177LU  L 573.6203  14   3/2+          3.5 NS     10                       D
177LUX L XREF=AE(*576)F(*574)G
177LU cL J$L=0+2 for 1/2+, 3/2+ doublet with {+176}Yb({+3}He,d),(|a,t); 115.665|g
177LU2cL M1 to 5/2+ and 573.6|g to 7/2+; band assignment.
```

lines:4,597 bytes:372,276

Translate > / < — Including: continuation records, record comments; indent: 1; #1 OK; compact continuation records; new style of comment SYM$; use dot as flag in field 6 continuation records; #

JSON Do:[upload][clear][copy][paste][mini][midi][maxi][tree]

```
 ]
},
{"E": "569.6721",
 "DE": "15",
 "J": "   1/2+",
 "T": "155 US",
 "DT": " 7",
 "C": "D",
 "MS": "M2",
 "continuation":
  "XREF=AE(*576)F
 ],
 "comments": [
  {"C": "c",
   "SYM": "J",
   "CTEXT": [
    "L=0+2 for 1/
    "E2 to 5/2+;
   ]
  },
  {"C": "c",
   "SYM": "T",
   "CTEXT": [
    "Weighted ave
    "(1965He06)"
   ]
  },
  {"C": "c",
   "SYM": "CONF",
   "CTEXT": [
    "|p1/2[411] (
    "Based on the
    "and large si
    "in neighbori
   ]
  }
 ],
 "radiations": [
  {"type": "Gamma
   "E": "111.715"
```

datasets:8

Tree wrap text

File Edit View History Tools Help About myEnsdf 2025-06-18,16:37

- levels[7] {9} Level:457.9568 keV J: 5/2+ T:0.45 NS y:2
- levels[8] {7} Level:552.0960 keV J: 7/2+ y:5
- levels[9] {10} Level:569.6721 keV J: 1/2+ T:155 US y:2
  - E: 569.6721 #Level energy in keV
  - DE: 15 #Standard uncertainty in E

Edit LEVEL - Google Chrome

about:blank

Project MyEnsdf.json 2025-06-18,09:09:57

1/8) Dataset: [177LU] ADOPTED LEVELS, GAMMAS

77/600) Record: [L] Level

Energy (keV): 569.6721

Edit Record: LEVEL

Edit Record

ENSDF Cards

```
....|...10....|...20....|...30....|...40....|...50....|...60....|...70....|...80
123456789|123456789|123456789|123456789|123456789|123456789|123456789|123456789|
177LU  L 569.6721  15   1/2+          155 US     7                        DM2
177LUX L XREF=AE(*576)F(*574)G
177LU cL J$L=0+2 for 1/2+, 3/2+ doublet in {+176}Yb({+3}He,d),(|a,t); 111.715|g
177LU2cL E2 to 5/2+; configuration assignment.
177LU cL T$Weighted average of 150 |ms {I10} (1970F109) and 160 |ms {I10}
177LUxcL (1965He06)
177LU cL CONF$|p1/2[411] (d{-3/2}) Nilsson configuration.
177LU2cL Based on the observed in-band properties, such as alignment
177LU3cL and large signature splitting, and systematics of structures
177LU4cL in neighboring nuclei.
```

Save | Exit | [delete] | Split | Merge

Main Card

Name	Value	Len.	Col.	Description
#NUCID	177LU	5	1:5	*Nuclide identifcation*
#RTYPE	L	1	8	*Letter 'L' is required*
E	569.6721	10	10:19	*Level energy in keV*
DE	15	2	20:21	*Standard uncertainty in E*
J	1/2+	18	22:39	*Spin and parity*
T	155 US	10	40:49	*Half-life of the level; units must be given*
DT	7	6	50:55	*Standard uncertainty in T*
L		9	56:64	*Angular momentum transfer in the reaction*
S		10	65:74	*Spectroscopic strength for this level*
DS		2	75:76	*Standard uncertainty in S*
C	D	1	77	*Comment FLAG used to refer to a particular comment record*
MS	M2	2	78:79	*Metastable state*
Q		1	80	*Questionable level*

Continuation Records

#	Continuation Record	Help
1	XREF=AE(*576)F(*574)G	[?]
2		[?]
3		[?]

Comment Records

#	C	RTYPE	PSYM	[SYM$]CTEXT
1	c			J$L=0+2 for 1/2+, 3/2+ doublet in {+176}Yb({+3}He,d),(\|a,t); 111.715\|g E2 to 5/2+; configuration assignment. J$L=0+2 for 1/2+, 3/2+ doublet in 176Yb(3He,d),(α,t); 111.715γ E2 to 5/2+; configuration assignment.

ENSDF-Manual

9 The Level Record

Optional, although a data set usually has at least one.

Field (Col.)	Name	Description
1-5	NUCID	Nuclide identification
6		Blank Any alphanumeric character other than '1' for continuation records
7		Must be blank
8	L	Letter 'L' is required
9		Must be blank
10-19	E	Level energy in keV - *Must not be blank*
20-21	DE	Standard uncertainty in **E**
22-39	J	Spin and parity
40-49	T	Half-life of the level; units *must* be given. Mean-life expressed as the width of a level, in units of energy, may also be used
50-55	DT	Standard uncertainty in **T**
56-64	L	Angular momentum transfer in the reaction determining the data set. (Whether it is L_n, L_p, $\Delta\mathbf{L}$, *etc.*, is determined from the **DSID** field of the **IDENTIFICATION**

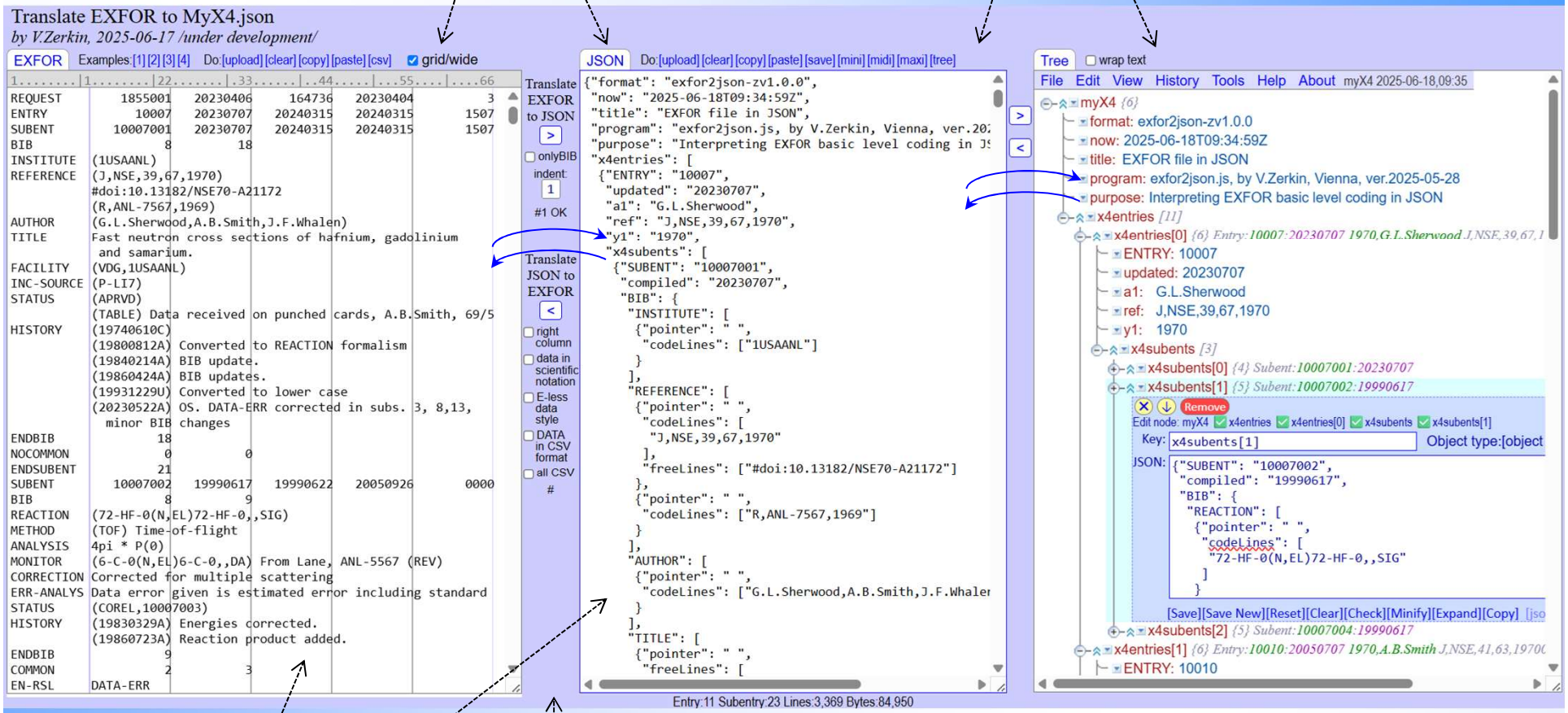
Reproduced via Descriptor

Click to open Popup-Window

Edit and [Save]

Translate EXFOR to MyX4.json via Web interface

EXFOR ⇔ JSON **JSON ⇔ JSON-Tree editor**



Editable text areas:

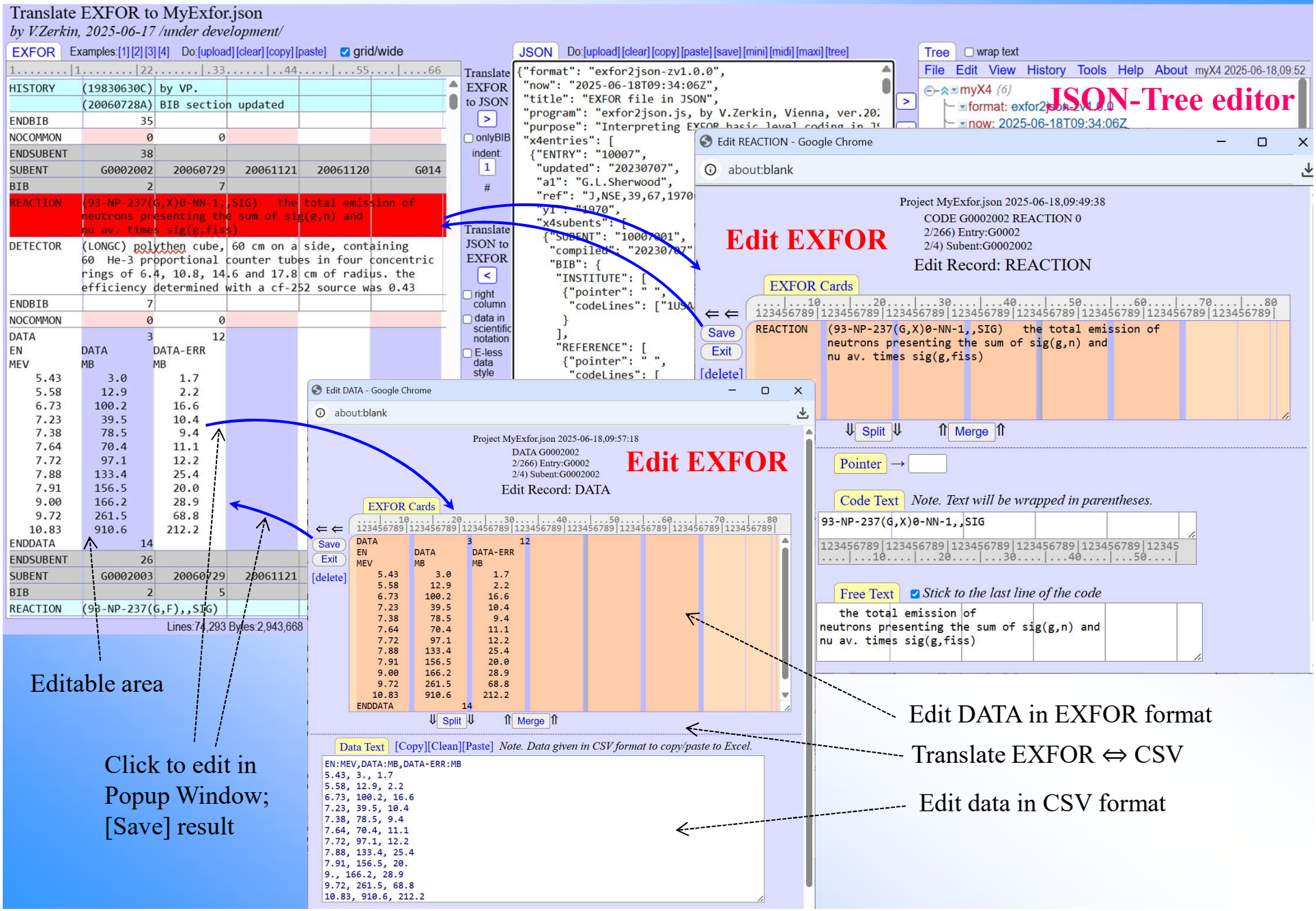
Options:

- EXFOR to JSON:
  - only BIB (no COMMON, no DATA sections)
- JSON to EXFOR:
  - right column (booking-info)
  - data values in scientific notation, e.g. 0.00000123 => 1.23E-6
  - E-less data style, e.g. 1.997512e-19 => 1.997512-19
  - DATA in CSV form
  - All in CSV: BIB (coded information), COMMON and DATA

Python parser: ~1.1K lines
to_cards(): not yet done

JavaScript: parser ~1K lines;
to_cards(): ~600 lines

Colored EXFOR with built-in editors



Why CSV for DATA inside EXFOR?

Original EXFOR: 3 lines per data point

```
STATUS      (TABLE) Table V of Phys.Rev.C81(2010)064604,            23114002   81
                    Appendix B of Sage's thesis (micro correlation)23114002   82
HISTORY     (20130924A) On. ERR-ANALYS, COVARIANCE etc.             23114002   83
ENDBIB                81                                          23114002   84
COMMON                 4          3                               23114002   85
ERR-3       ERR-4       ERR-5      ERR-6                          23114002   86
PER-CENT    PER-CENT    PER-CENT   PER-CENT                       23114002   87
 1.2         0.1         0.3        3.                            23114002   88
ENDCOMMON             3          0                               23114002   89
DATA                 13          9                               23114002   90
EN          EN-ERR      DATA       ERR-T      MONIT-ERR  ERR-1    23114002   91
ERR-2       ERR-7       ERR-8      MISC1      MISC2      MISC3    23114002   92
MISC4                                                           23114002   93
MEV         MEV         MB         PER-CENT   PER-CENT   PER-CENT 23114002   94
PER-CENT    PER-CENT    PER-CENT   NO-DIM     NO-DIM     NO-DIM   23114002   95
NO-DIM                                                          23114002   96
 8.34        0.15        96.8       6.5        1.9        5.      23114002   97
 1.          0.9                    0.9974     0.9925     1.      23114002   98
 1.                                                             23114002   99
 9.15        0.15        162.9      5.7        1.9        4.      23114002  100
 1.          0.6                    1.0731     1.3117     1.      23114002  101
 1.                                                             23114002  102
 13.33       0.15        241.8      4.6        1.6        2.5     23114002  103
 1.          0.4         0.3        0.9168     0.8288     1.      23114002  104
 1.                                                             23114002  105
 16.1        0.15        152.4      4.6        2.         2.1     23114002  106
 1.          0.6         0.3        1.0749     1.2335     1.      23114002  107
 1.                                                             23114002  108
 17.16       0.03        116.1      4.4        2.         1.5     23114002  109
 1.          0.6         0.3        0.9987     0.9878     0.998   23114002  110
 0.997                                                          23114002  111
 17.9        0.1         105.7      4.4        2.2        1.3     23114002  112
 0.7         0.7         0.3        0.969      0.933      0.998   23114002  113
 0.997                                                          23114002  114
 19.36       0.15        89.5       8.2        3.1        6.3     23114002  115
 2.          0.6         1.3        1.0061     1.0157     0.941   23114002  116
 0.926                                                          23114002  117
 19.95       0.07        102.1      5.8        4.1        1.4     23114002  118
 1.          0.6         1.4        0.9822     0.9433     0.922   23114002  119
 0.891                                                          23114002  120
 20.61       0.04        77.9       8.8        5.4        5.7     23114002  121
 1.6         0.6         1.4        0.9938     0.982      0.885   23114002  122
 0.832                                                          23114002  123
ENDDATA              33          0                               23114002  124
ENDSUBENT           123          0                               2311400299999
```

- EXFOR with DATA and COMMON sections presented in CSV form (including headers, units, pointers coded in one line) is fully equivalent to EXFOR
- compact, clearer, easy to observe and make copy/paste
- better if we need significantly extend number of partial uncertainties (may be needed for SG54 curated data)

Largest EXFOR Entry: 14508
- with right-column: 94MB 100%
- without right-column: 51MB 54%
- with DATA in CSV: 31MB 33%

DATA in CSV form (comma separated values):

```
STATUS      (TABLE) Table V of Phys.Rev.C81(2010)064604,            23114002   81
                    Appendix B of Sage's thesis (micro correlation)23114002   82
HISTORY     (20130924A) On. ERR-ANALYS, COVARIANCE etc.             23114002   83
ENDBIB                81                                          23114002   84
COMMON                 4          3                               23114002   85
#h:23114002:COMMON,ERR-3:PER-CENT,ERR-4:PER-CENT,ERR-5:PER-CENT,ERR-6:PER-CENT
#,1.2,0.1,0.3,3
ENDCOMMON             3          0                               23114002   86
DATA                 13          9                               23114002   87
#h:23114002:DATA,EN:MEV,EN-ERR:MEV,DATA:MB,ERR-T:PER-CENT,MONIT-ERR:PER-CENT,ERR-1
#,8.34,0.15,96.8,6.5,1.9,5,1,0.9,,0.9974,0.9925,1,1
#,9.15,0.15,162.9,5.7,1.9,4,1,0.6,,1.0731,1.3117,1,1
#,13.33,0.15,241.8,4.6,1.6,2.5,1,0.4,0.3,0.9168,0.8288,1,1
#,16.1,0.15,152.4,4.6,2,2.1,1,0.6,0.3,1.0749,1.2335,1,1
#,17.16,0.03,116.1,4.4,2,1.5,1,0.6,0.3,0.9987,0.9878,0.998,0.997
#,17.9,0.1,105.7,4.4,2.2,1.3,0.7,0.7,0.3,0.969,0.933,0.998,0.997
#,19.36,0.15,89.5,8.2,3.1,6.3,2,0.6,1.3,1.0061,1.0157,0.941,0.926
#,19.95,0.07,102.1,5.8,4.1,1.4,1,0.6,1.4,0.9822,0.9433,0.922,0.891
#,20.61,0.04,77.9,8.8,5.4,5.7,1.6,0.6,1.4,0.9938,0.982,0.885,0.832
ENDDATA              10          0                               23114002   88
ENDSUBENT            99          0                               2311400299999
```

EXFOR DATA-CSV exported to EXCEL:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	#h:23114002:DATA	EN:MEV	EN-ERR:MEV	DATA:MB	ERR-T:PER-CENT	MONIT-ERR:P	ERR-1:PER	ERR-2:PER	ERR-7:PER	ERR-8:PER	MISC1:NO-DIM	MISC2:NO	MISC3:NO-DI	MISC4:NO-
2	#	8.34	0.15	96.8	6.5	1.9	5	1	0.9		0.9974	0.9925	1	1
3	#	9.15	0.15	162.9	5.7	1.9	4	1	0.6		1.0731	1.3117	1	1
4	#	13.33	0.15	241.8	4.6	1.6	2.5	1	0.4	0.3	0.9168	0.8288	1	1
5	#	16.1	0.15	152.4	4.6	2	2.1	1	0.6	0.3	1.0749	1.2335	1	1
6	#	17.16	0.03	116.1	4.4	2	1.5	1	0.6	0.3	0.9987	0.9878	0.998	0.997
7	#	17.9	0.1	105.7	4.4	2.2	1.3	0.7	0.7	0.3	0.969	0.933	0.998	0.997
8	#	19.36	0.15	89.5	8.2	3.1	6.3	2	0.6	1.3	1.0061	1.0157	0.941	0.926
9	#	19.95	0.07	102.1	5.8	4.1	1.4	1	0.6	1.4	0.9822	0.9433	0.922	0.891
10	#	20.61	0.04	77.9	8.8	5.4	5.7	1.6	0.6	1.4	0.9938	0.982	0.885	0.832

Command line interface: tests, validation, utilities.

ENSDF

```
$ python -B EnsDom.py 177LU.ens
$ python -B EnsJson2Cards.py 177LU.ens.json
$ ens2flag6 177LU.ens > 177LU.ens.cmp
$ ens2flag6 177LU.ens.json.ens > 177LU.ens.json.ens.cmp
$ mycmp 177LU.ens.cmp 177LU.ens.json.ens.cmp
$ cat ens2flag6.c
  #include <ctype.h>
  #include <stdio.h>
  #include <string.h>
  #define LSTR    520
  char str[LSTR];
  int main(int argc,char **argv)
  {
      char ch, *ss, *fgets();
      FILE *in;
      char *infile;
      int  i;
      if (argc<2) {
          printf("At least one parameter needed!\n");
          return(0);
      }
      argv++; argc--;
      if (argc>0) {infile=*argv++; argc--;}
      in=fopen(infile,"r");
      if (in==NULL) return(-1);
      for (i=0; ;i++) {
          ss=fgets(str,LSTR-1,in);
          if (ss==NULL) break;
          if (strlen(str)>6) {
              ch=str[5];
              if (ch!=' ') str[5]='.';
          }
          printf("%s",str);
      }
  }
```

1) Translate ENSDF to MyEnsdf.json
2) Translate MyEnsdf.json to ENSDF
3) Modify flag-6 to "." for continuation records
4) Compare initial and finel ENSDF

G:\projects\zerkin\pdev\x4dev\ens1\177LU.ens.cmp

```
177LU  G     569.680 9  2.1     5 [M3]                          0.235
177LU. G FL=0.0
177LU. G  KC=0.183 3$LC=0.0395 6$MC=0.00930 13
177LU. G  NC=0.00220 3$OC=0.000319 5$PC=1.753E-5 25
177LU. G BM3W=0.14 4
177LU  L 573.6203  14    3/2+             3.5 NS     10
```

G:\projects\zerkin\pdev\x4dev\ens1\177LU.ens.json.ens.cmp

```
177LU  G     569.680 9  2.1     5 [M3]                          0.235
177LU. G FL=0.0$KC=0.183 3$LC=0.0395 6$MC=0.00930 13$NC=0.00220 3
177LU. G OC=0.000319 5$PC=1.753E-5 25$BM3W=0.14 4
177LU  L 573.6203  14    3/2+             3.5 NS     10
```

EXFOR

```
                +--------------------------------------+
                |          Update EXFOR files.         |
                |  Program x4update.py ver.2024-11-13  |
                |        by V.Zerkin, Vienna, 2024     |
                +--------------------------------------+
Program: x4update.py
Package "x4py" version: 1.0.0
Running: 2025-06-18 21:29:50
Help.
Purpose: merge/update EXFOR files
Functions:
  * standalone maintenance of EXFOR Master file
  * join/merge EXFOR files using latest Subentries
  * cut EXFOR text after 66th column and right-trim strings
  * add right column after 66th column to EXFOR text
  * sort Entries in EXFOR file
  * supporting NOENTRY (to exclude Entry from output)
  * split EXFOR file(s) by ENTRY and store one file for one Entry
Algorithm:
  * program in the loop on command line arguments:
    - reads EXFOR file (Backup/Master/TRANS or any other) into buffer
    - inserts/overwrites next Entry/Subentry into the buffer
    - accepts options and formatting parameters
  * output content of buffer to new EXFOR file or directory
Usage: $ python [{flag}] x4update.py {[option|file]}
  * flag: see all Python flags:  $ python --help
    -B        don't write .pyc files on import
  * option:
    -help     print this help-text and exit (also --help)
    -h:<hdr>  set header-line in output file (default: -t:REQUEST)
    -n1:<N1>  set N1 in header-line (default: -n1:777)
    -wide     add to EXFOR file right column (default: cut after 66-col.)
    -o:<file> write final buffer to new EXFOR file
    -d:<dir>  split final buffer by Entry to directory structure
    -sdd      set date of modification by Entry.N2 to <dir>
    -i:<Ent>  include Entries starting with <Ent>
    -x:<Ent>  exclude Entries starting with <Ent>
    -v:<show> verbose - show details of the process
  * file: file name should not start with sign "-"
    <file>    path of an EXFOR file (can be relative or absolute)
Examples:
  1) load Master file, update by Trans-file(s), write new Master file
     $ python3 -B x4update.py EXFOR-2023-06-30.bck trans.4213 -o:out1.bck -h:LIB
  2) insert/replace EXFOR Entries into TRANS file, verbose (trace of the processing)
     $ python3 -B x4update.py trans.4213 mm03.x4 -wide -h:TRANS -n1:4213 -o:trans.4213 -v:1
  3) split EXFOR file into sub-directoriess by Entries, set dir-dates by Entry:N2
     $ python3 -B x4update.py EXFOR-2023-05-23.bck -wide -d:x4all -sdd
  4) join EXFOR files from Area-4 to single file (use with bash or in MinGW/MSYS)
     $ python3 -B x4update.py x4all/4/*/*.x4 -o:area4.x4 -wide
  5) merge EXFOR files, filter Entries: include 41* and exclude 414*, output Entries to
out1/
     $ python -B x4update.py trans.4213 mm03.x4 trans.e150 -d:out1 -i:41 -x:414
  6) print help message and exit
     $ python3 -B x4update.py
```

x4update.py – update, split and merge EXFOR files, maintain Master file without intermediate files; using "x4py" package

Concluding remarks

Technical:

Step-1. Low-level EXFOR-Driver. Done:

1. small library of classes written in Python and JavaScript
2. needs only valid EXFOR file/text (Dictionaries are not required)
3. implements sequential and direct access using DOM
4. works with small (one Entry) and large files: retrieved/assembled/Trans/Prelim/Master
5. can translate EXFOR to JSON and backward from JSON to EXFOR and CSV
6. having command-line and Web interface for translation with various options and operations
7. integrated with Web-App "JSON-Tree editor"

Possible plans:

1. Step-2: classes for specific coding of essential BIB-keywords
2. Step-3: connection to EXFOR-Dictionaries
3. Step-4: datasets, variables and computational values
4. etc. … finally reproducing functionality of x4java/x4pro/c5/x5

Conclusions/Proposal/Question:

1. Several EXFOR parsers and converters to different versions of JSON exist
2. There is no EXFOR-JSON format agreed in NRDC and recommended for end-users (X5json?)
3. Recommended/stable/tested EXFOR-Driver is missing as starting tool for users who want to use original EXFOR instead of Web/X4Pro/X4Pro/C5/X4+ (e.g. WPEC-SG50/SG54).
4. If supported by NRDC, a common/generic low-level EXFOR-Driver on Python/JavaScript can be discussed in detailed, published as open-source product, further developed and possibly recommended for SG54 and other users' communities
5. Are NRDC, NNDC, SG54 and other nuclear data communities interested in ND-JSON-Tree Editor? If yes, conclusion and support are needed.

Thank you.