# ENSDF JSON format

Chris Morse

Not export controlled

Brookhaven National Laboratory
U.S. DEPARTMENT OF ENERGY
@BrookhavenLab

# Outline

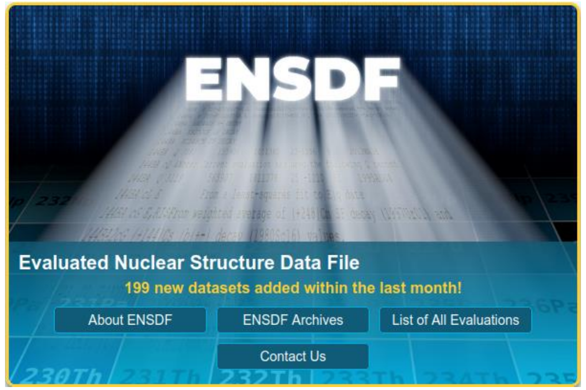
- Introductory remarks
- Overview of the JSON file structure
- Comments on the new format

# How it started

The design of ENSDF effectively envisions two kinds of users:

1. Evaluators
2. Journal readers

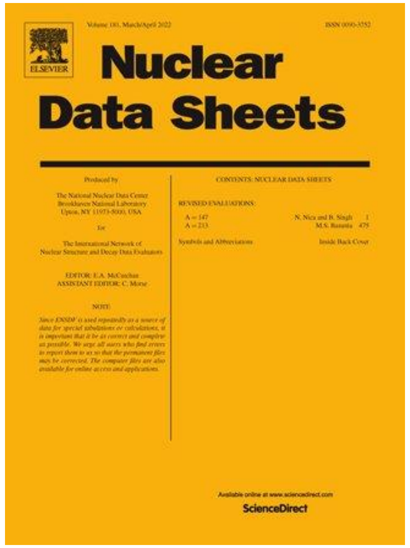
Here be evaluators





```
143BA  L 0.0        5/2-              14.5 S    3                          A
143BAX L XREF=ABC
143BA2 L %B-=100
143BA cL J$hfs (1988We07,1983Mu12,1981Ne06), |p from analysis of |m and
143BA2cL large negative Q. They suggest decoupled configuration with
143BA3cL main components |n(1/2-[530]) and |n(3/2-[532])
143BA cL T$from wt av (same as LWM) based on T{-1/2}=14.33 s {I8} (1986Ok03),
143BA2cL 15.2 s {I2} (1979En02), 15.17 s {I38} (1976AmZW), 14.5 s {I5}
143BAxcL (1978Pa01), 13.2 s {I3} (1969Ru14), 12 s {I3} (1962Wa36). See also
143BAxcL 1973Ta13
143BA2 L MOMM1=+0.443 11 (1988We07,2011StZZ)
143BA cL MOMM1     Other: +0.454 {I20} hfs (1983Mu12)
143BA3 L MOME2=-0.88 2 (1988We07,2011StZZ)
143BA cL MOME2     Other: -0.81 {I7} hfs (1983Mu12)
143BA  L 33.29      3(1/2)-
143BAX L XREF=AC
143BA cL J         |g to 5/2- is E2, log| {Ift}=6.0 via 3/2+ parent
143BA  G 33.46       100       E2                      125.7
143BA cG M$From |a(L)(exp)=107 {I21} and ce(L)/ce(M)|?5 in {+143}Cs |b{+-}
143BA2cG Decay, after normalizing electron and |g-ray intensities using
143BA3cG |a(K)(117|g, E2, theory)=0.747.
143BAS G LC=98.9 14$MC=21.8 3$NC+=5.05 7
143BAS G NC=4.48 7$OC=0.569 8$PC=0.000631 9
143BA  L 117.368   24 9/2-              3.5 NS    8                          A
143BAX L XREF=ABC
143BA2 L MOMM1=+0.5 3 (1999Sm05,2011StZZ)
143BA cL T$from LWM based on T{-1/2}=2.6 ns {I8} (1999Sm05), 3.8 ns {I12} from
143BAxcL |b{+-} decay (1979Sc11), 6 ns {I2} from {+252}Cf SF (1974ClZX).
143BA2cL Other: 2005Fo17.
143BA cL           |m from g-factor (1999Sm05)
143BA  G 117.32    5 100       E2                      1.094
143BAB G BE2W=1.0E+2 4
143BAS G KC=0.741 11$LC=0.278 4$MC=0.0605 9$NC+=0.01432 21
143BAS G NC=0.01260 18$OC=0.001686 24$PC=3.52E-5 5
143BA cG M         |a(K)exp=0.846 {I25}, |a(L)exp=0.378 {I113}, |a(M)exp=0.102
143BAxcG {I35}
```


Everyone else



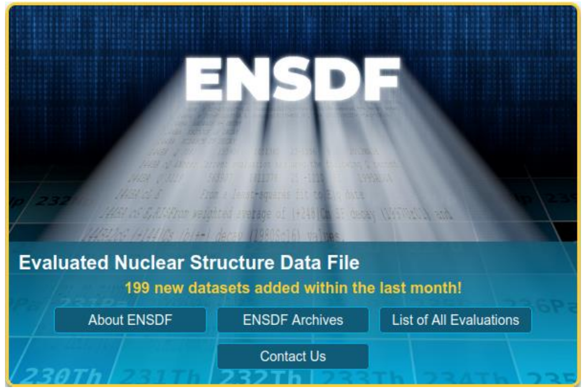

# How it's going

The design of ENSDF effectively envisions two kinds of users:
1. Evaluators
2. Journal readers

But there is a third class of users which is effectively unsupported: computational users

Computers...?

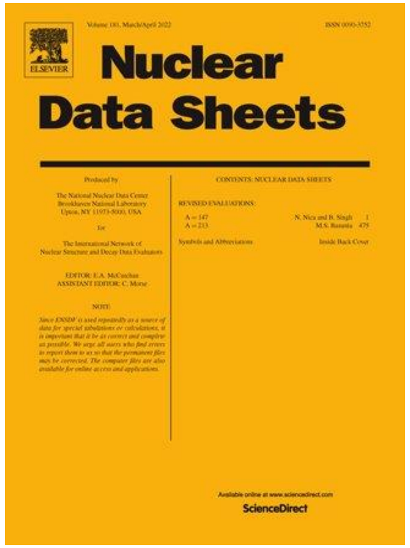
Here be evaluators





```
143BA  L 0.0         5/2-              14.5 S    3                          A
143BAX L XREF=ABC
143BA2 L %B-=100
143BA cL J$hfs (1988We07,1983Mu12,1981Ne06), |p from analysis of |m and
143BA2cL large negative Q. They suggest decoupled configuration with
143BA3cL main components |n(1/2-[530]) and |n(3/2-[532])
143BA cL T$from wt av (same as LWM) based on T{-1/2}=14.33 s {I8} (1986Ok03),
143BA2cL 15.2 s {I2} (1979En02), 15.17 s {I38} (1976AmZW), 14.5 s {I5}
143BAxcL (1978Pa01), 13.2 s {I3} (1969Ru14), 12 s {I3} (1962Wa36). See also
143BAxcL 1973Ta13
143BA2 L MOMM1=+0.443 11 (1988We07,2011StZZ)
143BA cL MOMM1     Other: +0.454 {I20} hfs (1983Mu12)
143BA3 L MOME2=-0.88 2 (1988We07,2011StZZ)
143BA cL MOME2     Other: -0.81 {I7} hfs (1983Mu12)
143BA  L 33.29      3(1/2)-
143BAX L XREF=AC
143BA cL J         |g to 5/2- is E2, log| {Ift}=6.0 via 3/2+ parent
143BA  G 33.46       100       E2                    125.7
143BA cG M$From |a(L)(exp)=107 {I21} and ce(L)/ce(M)|?5 in {+143}Cs |b{+-}
143BA2cG Decay, after normalizing electron and |g-ray intensities using
143BA3cG |a(K)(117|g, E2, theory)=0.747.
143BAS G LC=98.9 14$MC=21.8 3$NC+=5.05 7
143BAS G NC=4.48 7$OC=0.569 8$PC=0.000631 9
143BA  L 117.368   24 9/2-             3.5 NS    8                          A
143BAX L XREF=ABC
143BA2 L MOMM1=+0.5 3 (1999Sm05,2011StZZ)
143BA cL T$from LWM based on T{-1/2}=2.6 ns {I8} (1999Sm05), 3.8 ns {I12} from
143BAxcL |b{+-} decay (1979Sc11), 6 ns {I2} from {+252}Cf SF (1974ClZX).
143BA2cL Other: 2005Fo17.
143BA cL           |m from g-factor (1999Sm05)
143BA  G 117.32    5 100       E2                    1.094
143BAB G BE2W=1.0E+2 4
143BAS G KC=0.741 11$LC=0.278 4$MC=0.0605 9$NC+=0.01432 21
143BAS G NC=0.01260 18$OC=0.001686 24$PC=3.52E-5 5
143BA cG M         |a(K)exp=0.846 {I25}, |a(L)exp=0.378 {I113}, |a(M)exp=0.102
143BAxcG {I35}
```


Humans




# Problems for non-experts

The 80-column ENSDF format is hard to use

- For the standard one-card records:
  - No delimiters - must remember field widths
  - No labels - must remember field locations
  - Inconsistent units - must remember what/where
  - Asymmetric errors almost never supported
- For the continuation items:
  - Labels can be confusing, often used inconsistently
  - Units are not allowed
  - Multiple ways to indicate limits or approximations

```
143BA  L 0.0        5/2-              14.5 S    3                          A
143BAX L XREF=ABC
143BA2 L %B-=100
143BA cL J$hfs (1988We07,1983Mu12,1981Ne06), |p from analysis of |m and
143BA2cL large negative Q. They suggest decoupled configuration with
143BA3cL main components |n(1/2-[530]) and |n(3/2-[532])
143BA cL T$from wt av (same as LWM) based on T{-1/2}=14.33 s {I8} (1986Ok03),
143BA2cL 15.2 s {I2} (1979En02), 15.17 s {I38} (1976AmZW), 14.5 s {I5}
143BAxcL (1978Pa01), 13.2 s {I3} (1969Ru14), 12 s {I3} (1962Wa36). See also
143BAxcL 1973Ta13
143BA2 L MOMM1=+0.443 11 (1988We07,2011StZZ)
143BA cL MOMM1     Other: +0.454 {I20} hfs (1983Mu12)
143BA3 L MOME2=-0.88 2 (1988We07,2011StZZ)
143BA cL MOME2     Other: -0.81 {I7} hfs (1983Mu12)
143BA  L 33.29      3(1/2)-
143BAX L XREF=AC
143BA cL J         |g to 5/2- is E2, log| {Ift}=6.0 via 3/2+ parent
143BA  G 33.46      100       E2                       125.7
143BA cG M$From |a(L)(exp)=107 {I21} and ce(L)/ce(M)|?5 in {+143}Cs |b{+-}
143BA2cG Decay, after normalizing electron and |g-ray intensities using
143BA3cG |a(K)(117|g, E2, theory)=0.747.
143BAS G LC=98.9 14$MC=21.8 3$NC+=5.05 7
143BAS G NC=4.48 7$OC=0.569 8$PC=0.000631 9
143BA  L 117.368   24 9/2-            3.5 NS    8                          A
143BAX L XREF=ABC
143BA2 L MOMM1=+0.5 3 (1999Sm05,2011StZZ)
143BA cL T$from LWM based on T{-1/2}=2.6 ns {I8} (1999Sm05), 3.8 ns {I12} from
143BAxcL |b{+-} decay (1979Sc11), 6 ns {I2} from {+252}Cf SF (1974ClZX).
143BA2cL Other: 2005Fo17.
143BA cL           |m from g-factor (1999Sm05)
143BA  G 117.32    5 100      E2                       1.094
143BAB G BE2W=1.0E+2 4
143BAS G KC=0.741 11$LC=0.278 4$MC=0.0605 9$NC+=0.01432 21
143BAS G NC=0.01260 18$OC=0.001686 24$PC=3.52E-5 5
143BA cG M         |a(K)exp=0.846 {I25}, |a(L)exp=0.378 {I113}, |a(M)exp=0.102
143BAxcG {I35}
```

Brookhaven National Laboratory

# New paradigm

1. Evaluators interact with ENSDF via an editor (c.f. upcoming talk by D. Mason)
2. Human readers interact with ENSDF via PDFs
3. Computational users interact with ENSDF via new JSON format

Benefits:

- The representation of the data is decoupled from the data itself
  - E.g. evaluators do not have to worry about format changes, the editor handles those details
- JSON enjoys widespread adoption in computing
  - Much of the tool-development work is done for us

# What is JSON?

- A highly structured data interchange format
- Governed by a simple set of rules:
  - Data entries are key-value pairs
  - Keys are (unique) strings
  - Values can have three types:
    - Basic: string, integer, number, boolean, NULL
    - Object: A collection of key-value pairs enclosed in { }
    - Array: An ordered list of values enclosed in [ ]
- Trivially easy to deserialize

affiliations.json


```json
[
    {
        "institution": "University of Nowhere",
        "address": {
            "street": "University Ave",
            "number": 1,
            "zip": 12345
        },
        "presentAddress": true
    }
]
```


Brookhaven National Laboratory

# JSON with Python

deserialize.py

```
import json

with open("affiliations.json") as jsonfile
    jsondata = json.load(jsonfile)
    for item in jsondata:
        print(item["institution"])
```

```
ensdf@nndc:~$ python deserialize.py
University of Nowhere
```

affiliations.json

```
[
    {
        "institution": "University of Nowhere",
        "address": {
            "street": "University Ave",
            "number": 1,
            "zip": 12345
        },
        "presentAddress": true
    }
]
```

Brookhaven National Laboratory

# Overview of the new format

The new files are available at https://www.nndc.bnl.gov/ensdf-json

NB: These are still considered a beta release

# Organization

- Datasets
  - Header (Z, A, ...)
  - Comments
  - Various info (e.g. Q-values)
  - Levels
    - Level properties (energy, spin-parity, ...)
    - Radiations (alpha, beta, gamma...)
      - Radiation properties (energy, ...)

# Datasets

There are currently 14 defined types of datasets in the JSON format

The number of datasets in each category is given in parentheses

| | |
|---|---|
| adopted (3411) | general reaction (6976) |
| alpha decay (829) | isomer decay (589) |
| beta decay (2370) | neutron capture (607) |
| charge exchange (142) | prompt-particle decay (48) |
| coulomb excitation (389) | spontaneous fission (249) |
| delayed-particle decay (277) | transfer (2563) |
| fluorescence (198) | comments (278) |

# Header

Contains information to identify a file

- Nuclide properties like mass number (A), proton number (Z), element symbol
- Dataset name
  - Like DSID, but only used for human readers
- Dataset type
  - One of the 14 categories on the previous slide
- Database
  - E.g. ENSDF or XUNDL
- History
  - Full evaluations, or Updates

# Levels

- Organized as an array of objects
- Easy to iterate in code
- Contains the usual properties


```
{
    "energy": {
        "value": 0,
        "unit": "keV",
        "uncertainty": {
            "value": 0,
            "type": "symmetric"
        }
    },
    "halflife": {
        "value": 13,
        "unit": "ms",
        "uncertainty": {
            "type": "asymmetric",
            "upperLimit": 8,
            "lowerLimit": 5
        }
    }
}
```


Brookhaven National Laboratory

# Gammas

- Similar structure to levels
- Main difference is strict indexing
  - Exception is unplaced gammas


```json
{
    "energy": {
        "value": 121.03,
        "unit": "keV",
        "uncertainty": {
            "value": 0.1,
            "type": "symmetric"
        }
    },
    "initialStateIndex": 1,
    "finalStateIndex": 0
}
```


Brookhaven National Laboratory

# Individual quantities

Almost everything is based on a common template which expresses physical values, containing:

- A value
- An uncertainty
  - Symmetric
  - Asymmetric
  - Limit
  - Approximation
  - Unreported
- Extensible as needed

```
{
    "energy": {
        "value": 121.03,          ← Always
        "unit": "keV",            ← If needed
        "uncertainty": {          ← Always
            "value": 0.1,
            "type": "symmetric"
        }
    }
}
```

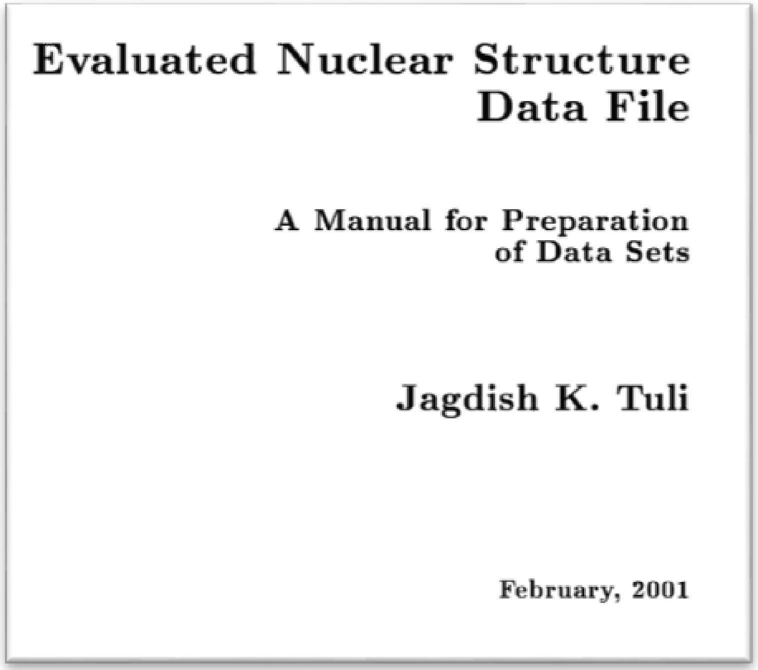
Brookhaven National Laboratory

# Remarks on the new format

Brookhaven National Laboratory

# What isn't JSON?

- Code: JSON does nothing on its own
  - ...but there are tools to make it do things
- A replacement for documentation
  - ...but it does make the data more expressive
- A quick fix for the challenges of evaluations
  - ...but it creates some new possibilities


Evaluated Nuclear Structure
Data File

A Manual for Preparation
of Data Sets

Jagdish K. Tuli

February, 2001

# How can we leverage JSON in software?

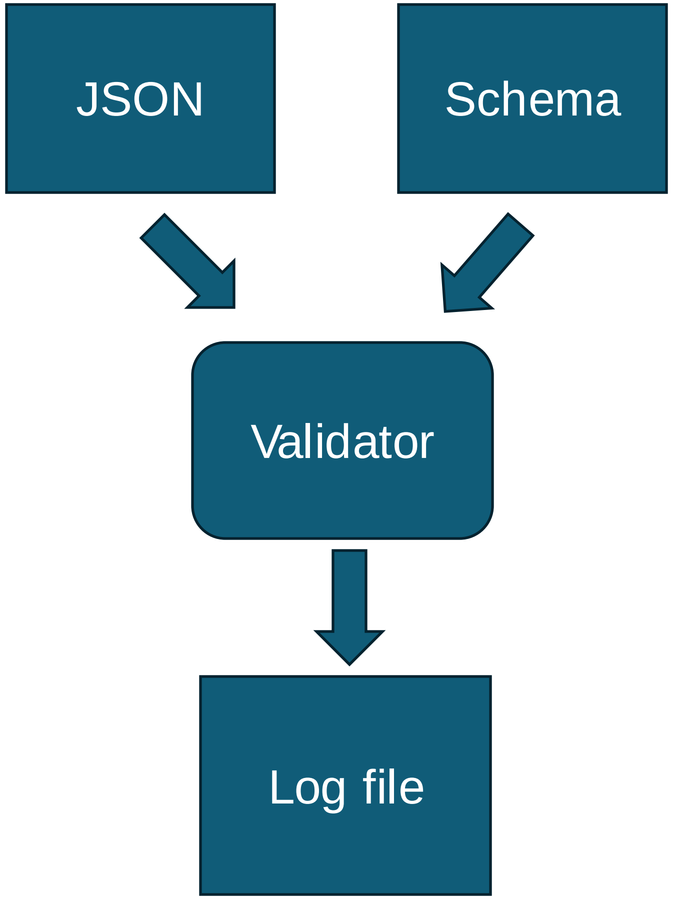
JSON Schema is a means of enforcing data consistency

- More specifically to our use case:
  - A set of rules for JSON documents
  - Defines what data are allowed and in what form
  - Extensible

```
{
    "$id": "my.schema/person.json,
    "type": "object",
    "properties": {
        "name": {
            "type": "string"
        },
        "age": {
            "type": "integer"
        },
        "passport": {
            "type": "integer"
        }
    }
}
```

# Validation

- The validator **is** code (we're using Python)
- The validator compares a document to a specified schema
- If either input is malformed, it errors out
  - We develop the schema, and warrant that it is correct
  - Use of the new editor ensures correct documents
    - D. Mason will cover the editor later
- Finally, a list of validation failures, if any, is printed
- NB: validation is like running FMTCHK




Brookhaven National Laboratory

# Can the data speak for itself?

Not entirely, but we can do a lot

- Several features of JSON allow control of the data
  - Restrict possible values in a field
  - Provide inline documentation/annotations
  - Require or forbid certain data
    - Can be based on conditional statements
- Like a highly formalized format manual
  - Can encode (some) policies
  - Closer to FMTCHK than Consistency Check
- Can even auto-generate code from schema

```
{
    "$id": "my.schema/person.json,
    "type": "object",
    "description": "A person",
    "properties": {
        "name": {
            "type": "string"
        },
        "age": {
            "type": "integer",
            "minimum": 0
        },
        "passport": {
            "type": "integer"
        }
    },
    "required": [ "name", "age" ],
    "additionalProperties": false
}
```

Brookhaven National Laboratory

# Will it make my life easier?

Not on its own - but it makes tool development much easier

- Example: measurements
  - Dedicated spot for individual measurements
  - New evaluation? Just add new measurements
- Hypothetical: web-aware evaluations
  - Automatic referencing to XUNDL?

```
"measurements": {
    "method": "bestValue",
    "summary": "From 2017Ar10",
    "measuredValues": [
        {
            "value": 10.1,
            "unit": "s",
            "uncertainty": {
                "value": 0.3,
                "type": "symmetric"
            },
            "isIncluded": true,
            "reference": "2017Ar10"
        }
    ]
}
```

Brookhaven National Laboratory

# Questions?

AMA