LANL Activities Regarding GNDS

LA-UR-18-24032

May 14, 2018

SG43 (API) Status Report

SG43 Mandate

Particular infrastructure needs identified by SG38 as being fundamentally will necessary include:

- *An Application Programming Interface (API) for reading and writing data in the new structure; and*
- *Checking codes to help validate new evaluations and fix problems identified during validation. These include checks for proper formatting and completeness.*

SG43—Initial Subcommittees

API Design (Caleb Mattoon)

- API naming standards
- Low-level abstraction
- Fill out initial API details

Physics Checking (Jeremy Conlin)

- Compile list of physics checks
- Standardize severity levels
- Standardized output reporting

SG43—Initial Subcommittees

API Design (Caleb Mattoon)

- API naming standards
- Low-level abstraction
- Fill out initial API details

Physics Checking (Jeremy Conlin)

- Compile list of physics checks
- Standardize severity levels
- Standardized output reporting

https://github.com/GeneralizedNuclearData/SG43/wiki

Combined Work

- Want usage case (e.g., physics checking) to help inform how the API functions
- Implement just enough of the API to perform a simple physics check
- Adjust API based on lessons learned
- Repeat

SG43 Agenda

- Introduction
- Progress towards meeting mandate
  - API
  - Physics checking
- Thoughts on API design
- Existing GNDS implementations
  - GIDI at LLNL
  - AMPX at ORNL
- Data testing
- Final discussion

Physics Testing

SG43 Mandate

Particular infrastructure needs identified by SG38 as being fundamentally will necessary include:

- *An Application Programming Interface (API) for reading and writing data in the new structure; and*
- *Checking codes to help validate new evaluations and fix problems identified during validation. These include checks for proper formatting and completeness.*

SG43 Mandate

Particular infrastructure needs identified by SG38 as being fundamentally will necessary include:

- *An Application Programming Interface (API) for reading and writing data in the new structure; and*
- *Checking codes to help validate new evaluations and fix problems identified during validation. These include checks for proper formatting and completeness.*

Severity Levels

Severity Levels

`INFO`	Informational messages; typically given after earlier problem.
`NOTICE`	Not an error condition, but may require more investigation or handling.
`WARNING`	The data is not perfect, but we can work around it.
`ERROR`	The data is clearly wrong and there is no unambiguous way to fix it.
`FATAL`	Something really bad happened—can't continue.

Severity Levels

`INFO`	Informational messages; typically given after earlier problem.
`NOTICE`	Not an error condition, but may require more investigation or handling.
`WARNING`	The data is not perfect, but we can work around it.
`ERROR`	The data is clearly wrong and there is no unambiguous way to fix it.
`FATAL`	Something really bad happened—can't continue.

Physics Requirements

- Cross Section Requirements
- Secondary Distribution Requirements
  - Multiplicity Requirements
  - Fission ν Requirements
- Resonance Requirements
  - Resolved Resonance Region Requirements
  - Unresolved Resonance Region Requirements
- Derived Data Requirements

Cross Section Requirements

- Monotonically increasing energy grid
  - Discontinuity with > 2 equivalent energy points is an error
- Strictly positive energy values.
- No negative* cross section values.
- Appropriate kinematic threshold

$$\text{threshold} = -Q\frac{A + A_i}{A} \tag{1}$$

- Partial cross sections sum to total.
- No abnormally large or abnormally small cross section values.

Secondary Distribution Requirements

- Monotonically increasing energy grid.
- Monotonically increasing cosine grid.
- Secondary energy $\leq$ incident energy*.
- No negative angular probability values.
- No abnormally small angular probability values.

Multiplicity Requirements

- Monotonically increasing energy grid.
- No negative multiplicity values.
- No abnormally large or small multiplicity values.

Fission ν Requirements.

- Delayed neutron fraction must sum to 1.0.

Resolved Resonance Region Requirements

Unresolved Resonance Region Requirements

- Partial reactions must sum to total.
- No negative probability table values.

Derived Data Requirements

- Strictly positive heating values.
- No abnormally large heating values.

Reporting Results

Output needs to be:

- Computer readable—not arbitrary text
- Human readable—arbitrary formatting

Reporting Results

Output needs to be:

- Computer readable—not arbitrary text
- Human readable—arbitrary formatting

XML provides both

- Computer readable—tags & attributes/values
- Human readable—tag body

Reporting Results

Output needs to be:

- Computer readable—not arbitrary text
- Human readable—arbitrary formatting

XML provides both

- Computer readable—tags & attributes/values
- Human readable—tag body

JUnit output has become somewhat of a standard testing output framework.

- Java-based testing framework
- Python (unittest), C++ (Catch2) (others?) libraries to produce JUnit compatible output

JUnit Example 1

```
<?xml version="1.0" encoding="UTF-8" ?>
  <testsuites id="20180501_170519" name="Physics_check (2018-05-01 17:05:19)"
              tests="225" failures="1262" time="0.001">
    <testsuite id="cross.section.errors"
               name="Cross Section Errors" tests="6" failures="1" time="0.001">
      <testcase id="partials.add.to.total"
                name="Partial cross sections must sum to total" time="0.001">
        <failure message="Partial cross sections do not sum to total"
                 type="WARNING">
          WARNING: Partial cross sections do not sum to total
            Total reaction: 18
            Partial reactions: 19--21, 38
            File: /ENDF/neutrons/n-092_U_235.xml
        </failure>
      </testcase> </testsuite> </testsuites>
```

JUnit Example 2

```
<?xml version="1.0" encoding="UTF-8" ?>
  <testsuites id="20180501_171519" name="Physics_check (2018-05-01 17:15:19)"
              tests="225" failures="1262" time="0.001">
    <testsuite id="multiplicity.errors"
               name="Multiplicity Errors" tests="6" failures="1" time="0.001">
     <testcase id="no.negative.multiplicities"
                name="Multiplicities must be positive" time="0.001">
        <failure message="Found negative multiplicity"
                 type="ERROR">
          ERROR: Negative multiplicity value found
            multiplicity: -2.4
            energy: 1.0 MeV
            File: /ENDF/neutrons/n-092_U_235.xml
        </failure>
      </testcase> </testsuite> </testsuites>
```