

LA-UR-24-29265

Approved for public release; distribution is unlimited.

Title: SG50 Data-format Specifications Document for the Automatically Readable, Comprehensive, and Curated Experimental Reaction Database MEDUSAL

Author(s): Barry, Devin; Brown, Jesse; Danon, Yaron; Fleming, Michael; Foligno, Daniela; Holcomb, Andrew; Koning, Arjan; Lewis, Amanda; Neudecker, Denise; Okumura, Shin; Otsuka, Naohiko; Pritychenko, Boris; Schnabel, Georg; Sjostrand, Henrik; Smith, Donald; Wang, Jimin; Zerkin, Viktor

Intended for: Report
Web

Issued: 2025-01-23 (Rev.1) (Draft)



Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

SG50 Data-format Specifications Document for the
Automatically Readable, Comprehensive, and
Curated Experimental Reaction Database MEDUSAL

WPEC SG-50

December 13, 2024

Foreword

This report constitutes an initial version of the specifications for the experimental reaction database, MEDUSAL (Machine-readable Experimental Data User App & Library), created by OECD/NEA/WPEC SubGroup 50. This initial draft of the specifications is not complete, but will serve as the starting point for future work on the database and schema.

The NEA Working Party on International Nuclear Data Evaluation Co-operation (WPEC) was created in 1989, under the aegis of the Nuclear Science Committee (NSC), to improve the quality and completeness of nuclear data by bringing together representatives of the major nuclear data evaluation projects of NEA member countries and of selected invitees.

Acknowledgements

The members of the Nuclear Energy Agency (NEA), the Nuclear Science Committee and the Working Party on International Nuclear Data Evaluation Co-operation (WPEC), wish to acknowledge the authors of the working paper, including the co-Chairs of SG50 (*), the monitor of SG50 (**), and the NEA Secretariat (†).

Name	Country	Establishment
Devin Barry BARRY	USA	Naval Nuclear Laboratory
Jesse BROWN	USA	Oak Ridge National Laboratory
Yaron DANON	USA	Rensselaer Polytechnic Institute
Michael FLEMING		OECD Nuclear Energy Agency
Daniela FOLIGNO		OECD Nuclear Energy Agency
Andrew HOLCOMB†		OECD Nuclear Energy Agency
Arjan KONING**	Austria	International Atomic Energy Agency
Amanda LEWIS*	USA	Naval Nuclear Laboratory
Denise NEUDECKER*	USA	Los Alamos National Laboratory
Shin OKUMURA	Austria	International Atomic Energy Agency
Naohiko OTSUKA	Austria	International Atomic Energy Agency
Boris PRITYCHENKO	USA	Brookhaven National Laboratory
Georg SCHNABEL	Austria	International Atomic Energy Agency
Henrik SJÖSTRAND	Sweden	Uppsala University
Donald SMITH	USA	Argonne National Laboratory
Jimin WANG	China	China Nuclear Data Center
Viktor ZERKIN	Austria	International Atomic Energy Agency

Abbreviations and Acronyms

AI	Artificial Intelligence
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
DOI	Digital Object Identifier
EXFOR	EXchange FORmat (Nuclear Reaction Physics Database)
FAIR	Findable, Accessible, Interoperable, and Reproducible
GUI	Graphical User Interface
IAEA	International Atomic Energy Agency
JSON	JavaScript Object Notation
MEDUSAL	Machine-readable Experimental Data User App & Library
ML	Machine Learning
NDS	Nuclear Data Section (IAEA)
NEA	Nuclear Energy Agency
NRDC	Nuclear Reaction Data Centers
OECD	Organisation for Economic Co-operation and Development
RIPL	Reference Input Parameter Library
SQL	Structured Query Language
WPEC	Working Party on International Nuclear Data Evaluation Co-operation (NEA)
XML	eXtensible Markup Language

Contents

Acknowledgements	iii
Abbreviations and Acronyms	v
1 Introduction	1
1.1 Data Types	1
1.2 Database Layers	2
2 Data Types	7
2.1 Abstract	7
2.2 General Datatypes	15
2.3 Observable	23
2.4 Bibliography	51
2.5 Incident	59
2.6 Detector	92
2.7 Sample	145
2.8 Response Functions	185
2.9 Reference Data	190
2.10 Correlation	199
A Allowed Values	205
A.1 General Datatypes	206
A.2 Observable	210
A.3 Bibliography	216
A.4 Incident	217
A.5 Detector	221
A.6 Sample	224
A.7 Response Functions	227
A.8 Reference Data	228
A.9 Correlation	229
References	231
Index	233

1. Introduction

The aim of this document is to lay out a first draft of the specifications for the MEDUSAL database (Machine-readable Experimental Data User App & Library) that is being described by OECD/NEA/WPEC SG-50. The EXFOR database (Otuka et al., 2014) has a format that is based on code-value pairs, and a significant portion of the information in the EXFOR entry is contained in free text sections. Several high-level requirements for the MEDUSAL database, as laid out in the Use Cases and Requirements Working Paper ([citation](#)), relate to the definition of the specifications:

Creating a format that is easy to read automatically for a large amount of data All users would need to be able to download the data in a format that is easy to read. Especially, large-scale users (for model development or applying ~~ML/AI~~ ML/AI) need a format that is easy to read automatically.

Using unique identifiers for metadata to easily find common features among experiments It is key for users wanting to apply ~~ML/AI~~ ML/AI to metadata to have them in a common format and describe them with unique identifiers. That means that one and the same physics process, uncertainty source, equipment, etc., should be described with only one unique word. For instance, one might find several terms to describe detector-efficiency uncertainties in EXFOR (“extrapolation to zero”, “efficiency”, “detector efficiency”, “detector calibration”). While a human might be able to easily identify those as belonging to the all-encompassing term of “detector efficiency uncertainty”, a computer cannot easily make that connection. Having one common identifier to describe the uncertainty category, metadata feature, etc., increases interpretability. It also helps users such as evaluators to quickly compare two data sets.

The specifications for MEDSUAL attempt to meet this requirement by creating a set of data structures that completely cover the information stored in EXFOR, provide unique identifiers for the metadata, and remain as simple as is possible to allow for straightforward processing of the MEDUSAL entries. In this work, an initial draft [form](#) of the specifications for the database is proposed.

1.1 Data Types

There are three classes of data types described in this version of the MEDUSAL specifications — abstract data types, general data types, and the actual data containers

that will compose the experimental data set entries. The abstract classes form a basis for all data containers, and the data types are like templates that show up frequently across the data containers. They are described in brief in this introductory section, and in more detail in the following sections.

1.1.1 Abstract Data Types

The abstract data types that underlay all data containers are described in Section 2.1. These data types are based directly on the JSON schema data types (“JSON 2020-12 Release”, 2020), which allows for straightforward creation of a JSON schema for the data files. The full list of JSON data types are shown in Table 1.1, and are all included in the list of abstract data types. This initial draft of the specifications does not make use of all of the abstract data types, but they are defined for future expansion of the specifications.

Table 1.1: Abstract data types, which are directly based on the JSON data types (and share the same names). The analogous python data types are also listed.

Abstract data type	Python data type
null	None
boolean	bool
string	unicode
integer	int
number	float
array	list, tuple
object	dict

1.1.2 General Data Types

Throughout the specifications, there are some data type “templates” that are repeated many times over the specifications. These have been grouped into the general data types, which are all abstract type object. The general data types are listed in Table 1.2 and are described in Section 2.2.

1.1.3 Data Containers

The numerical data values and metadata for an experiment have been split up into eight categories, shown in Table 1.3. Within each category, there are individual data containers that hold one type of metadata for the experiment. The data containers include, and expand on, the EXFOR keywords and codes (Otuka et al., 2014). These containers should cover the entire space of information stored in EXFOR entries, although in this initial draft there is information in some EXFOR entries that would not fall into any of these containers.

1.2 Database Layers

Several of the high-level requirements defined for the MEDUSAL database in the Use Cases and Requirements Working Paper ([citation](#)) address the need for storing some

Table 1.2: General data types in the current version of the MEDUSAL specifications. A short description of the use of each general data type is given as well.

General data type	Use
text	single string
text_array	list of strings
isotope	identifier for an isotope
number	single number
numerical_array	list of numerical values
link	internal or external link
reference	a single reference
commerical_product	product details
correction	details of a correction
uncertainty	value, type, and unit of an uncertainty
measured_quantity	value(s) with a unit and uncertainty

Table 1.3: The data categories in the MEDUSAL database, as well as the sections in the document.

Category	Section
Observable	2.3
Bibliography	2.4
Incident	2.5
Detector	2.6
Sample	2.7
Response Functions	2.8
Reference Data	2.9
Correlation	2.10

metadata about where the information itself came from:

Storing past judgments on the data Evaluators and experimentalists gain knowledge on past data (e.g., on missing corrections, uncertainties, etc.) through detailed studies of their own. This information is often lost as there is no central place to store this information. However, it could be very helpful to future users of the data to know what earlier users of data uncovered about the data. MEDUSAL strives to store these past judgments in layer 3. A format needs to be developed to store this information and an API to save and upload the information.

Identifying if a data set was used for an evaluation It might be helpful for some users (validators, evaluators and experimentalists) to know if a specific data set was used for an evaluation as this lends credibility to a data set. It would also render evaluations more reproducible if the data could

be stored as used for the evaluation. The MEDUSAL database will store in layer 3 data as used for the evaluation and introduce a flag that shows if data were used for particular evaluations.

The way that these requirements are met in the current version of the MEDUSAL database specifications is by directly incorporating the metadata keywords `source` and `description` into the attributes of the data containers. The `source` keyword holds where the information in that container came from. Values such as `EXFOR`, `publication`, and `author` can be used when the information in the container was taken from the EXFOR compilation, a published reference, or from the authors. As the majority of the information in a data set entry will likely come from one source, each data set should have a default value for `source` keyword, and the keyword only explicitly included in containers that deviate from that default.

1.2.1 Layer 0

As defined in the MEDUSAL Requirements Chapter, the database Layer 0 is a translation of EXFOR entries into the MEDUSAL format. The only information in entries in this layer comes directly from the EXFOR entry, so the only `source` value in this layer is `EXFOR`.

JSON example of `source`

```
"attributes": {  
  "source": ["EXFOR"]  
}
```

1.2.2 Layer 1

Layer 1 of the database can also contain information in published reports [or from the authors](#) that was not compiled into EXFOR, [or information directly from the authors](#). Therefore, the only `source` values that will be in this layer are `EXFOR`, `publication`, and `authors`.

JSON example of `source`

```
"attributes": {  
  "source": ["publication"]  
}
```

The type of the `source` keyword is a `text_array`, rather than `text`, because information in one container may come from multiple sources. For example, information compiled in the EXFOR entry could be supplemented by descriptive text from the publication. In this case, the value of `source` would be `["EXFOR", "publication"]`.

JSON example of source

```
"attributes": {
  "source": ["EXFOR", "publication"]
}
```

1.2.3 Layer 2

Layer 2 of the database contains “objective” corrections, such as updates of standard values and flags indicating missing uncertainties (as described in the [SG50 Working Paper](#)). There can be multiple versions of the same container in this layer. If the information was updated by the standards, the version should be included. In this layer, there should always be a description of the updates.

JSON example of source

```
"attributes": {
  "source": ["Standards-2018"],
  "description": "Monitor value updated to 2018 standards value"
}
```

JSON example of source

```
"attributes": {
  "source": ["EXFOR", "templates"],
  "description": "Uncertainty is missing"
}
```

1.2.4 Layer 3

Layer 3 of the database contains “subjective” corrections, including uncertainty values updated based on the template values, and the exact version of a data set that an evaluator used in an evaluation (as described in the [SG50 Working Paper](#)). For versions of the data set used in specific evaluations, the library and version number should be used as the source. In this layer as well, all modifications should come with a description.

For example, if information in a container came from EXFOR and a publication, and then was modified slightly for an evaluation in ENDF-8.0, all three of those values should be listed.

JSON example of source

```
"attributes": {  
  "source": ["EXFOR", "publication", "ENDF-8.0"],  
  "description": "Uncertainty values increased by 3% for ENDF-8.0  
evaluation."  
}
```

If a modified version of the container is submitted by a user of the data, but was not used in a specific library evaluation, the year and last name of the submitter could be included.

JSON example of source

```
"attributes": {  
  "source": ["EXFOR", "year-name"],  
  "description": "description of the change"  
}
```

There can be multiple versions of the same data set in this layer, allowing for improved reproducibility and knowledge management. When searching the database, a user should be presented with all layer 1, 2, and 3 versions of the data set without recommendation.

2. Data Types

2.1 Abstract

The abstract datatypes are the base for all datatypes used in the schema. They are directly based on the JSON schema datatypes, so all higher-level schemas can be translated into JSON schema in a straightforward manner. Not all of the abstract datatypes are in use in this initial draft of the schema, but can be used in a future extension.

Section Contents:

2.1.1	Abstract type: array	8
2.1.2	Abstract type: boolean	9
2.1.3	Abstract type: integer	10
2.1.4	Abstract type: null	11
2.1.5	Abstract type: number	12
2.1.6	Abstract type: object	13
2.1.7	Abstract type: string	14

2.1.1 Abstract type: array

The basic array data type in the JSON schema.

Specifications for: array

Type: abstract

Allowed Keywords:

title: [str] the name of the array

description: [str] a description of the array

default: [dict] default instance of the array

examples: [list] a list of example array instances that validate correctly

readOnly: [bool] indicates to an API that the value should not be modified

writeOnly: [bool] indicates to an API that the value should remain hidden

deprecated: [bool] indicates whether the instance value is deprecated

\$comment: [str] comments to editors of the schema, not to users

const: [dict] used to restrict a value to a single value in all instances

enum: [dict] used to restrict a value to a certain set of values

items: [dict] used to define the allowed type(s) of the elements

prefixItems: [list] used for tuple validation by defining the expected type of each element in order

unevaluatedItems: [bool] whether or not to allow items that do not match the items or prefixItems values

contains: [dict] pattern or characteristic that must be matched by at least one element in the array

minContains: [int] minimum number of elements in the array that match the characteristics in contains

maxContains: [int] maximum number of elements in the array that match the characteristics in contains

minItems: [int] minimum number of elements in the array

maxItems: [int] maximum number of elements in the array

uniqueItems: [bool] whether or not all elements in the array need to be unique

2.1.2 Abstract type: boolean

The basic boolean data type in the JSON schema.

Specifications for: boolean

Type: abstract

Allowed Keywords:

title: [str] the name of the boolean

description: [str] a description of the boolean

default: [dict] default instance of the boolean

examples: [list] a list of example boolean instances that validate correctly

readOnly: [bool] indicates to an API that the value should not be modified

writeOnly: [bool] indicates to an API that the value should remain hidden

deprecated: [bool] indicates whether the instance value is deprecated

\$comment: [str] comments to editors of the schema, not to users

const: [dict] used to restrict a value to a single value in all instances

2.1.3 Abstract type: integer

The basic integer data type in the JSON schema.

Specifications for: integer

Type: abstract

Allowed Keywords:

title: [str] the name of the integer

description: [str] a description of the integer

default: [dict] default instance of the integer

examples: [list] a list of example integer instances that validate correctly

readOnly: [bool] indicates to an API that the value should not be modified

writeOnly: [bool] indicates to an API that the value should remain hidden

deprecated: [bool] indicates whether the instance value is deprecated

\$comment: [str] comments to editors of the schema, not to users

const: [dict] used to restrict a value to a single value in all instances

enum: [dict] used to restrict a value to a certain set of values

multipleOf: [int] used to restrict the value to being a multiple of a specific number

minimum: [int] used to restrict the value to being greater than or equal to a specific value

exclusiveMinimum: [int] used to restrict the value to being greater than a specific value

maximum: [int] used to restrict the value to being less than than or equal to a specific value

exclusiveMaximum: [int] used to restrict the value to being less than than a specific value

2.1.4 Abstract type: null

The basic null data type in the JSON schema.

Specifications for: null

Type: abstract

Allowed Keywords:

title: [str] the name of the null

description: [str] a description of the null

default: [dict] default instance of the null

examples: [list] a list of example null instances that validate correctly

readOnly: [bool] indicates to an API that the value should not be modified

writeOnly: [bool] indicates to an API that the value should remain hidden

deprecated: [bool] indicates whether the instance value is deprecated

\$comment: [str] comments to editors of the schema, not to users

const: [dict] used to restrict a value to a single value in all instances

enum: [dict] used to restrict a value to a certain set of values

2.1.5 Abstract type: number

The basic number data type in the JSON schema.

Specifications for: number

Type: abstract

Allowed Keywords:

title: [str] the name of the float

description: [str] a description of the float

default: [dict] default instance of the float

examples: [list] a list of example float instances that validate correctly

readOnly: [bool] indicates to an API that the value should not be modified

writeOnly: [bool] indicates to an API that the value should remain hidden

deprecated: [bool] indicates whether the instance value is deprecated

\$comment: [str] comments to editors of the schema, not to users

const: [dict] used to restrict a value to a single value in all instances

enum: [dict] used to restrict a value to a certain set of values

multipleOf: [int] used to restrict the value to being a multiple of a specific number

minimum: [int] used to restrict the value to being greater than or equal to a specific value

exclusiveMinimum: [int] used to restrict the value to being greater than a specific value

maximum: [int] used to restrict the value to being less than than or equal to a specific value

exclusiveMaximum: [int] used to restrict the value to being less than than a specific value

2.1.6 Abstract type: object

The basic object data type in the JSON schema.

Specifications for: object

Type: abstract

Allowed Keywords:

title: [str] the name of the object

description: [str] a description of the object

default: [dict] default instance of the object

examples: [list] a list of example object instances that validate correctly

readOnly: [bool] indicates to an API that the value should not be modified

writeOnly: [bool] indicates to an API that the value should remain hidden

deprecated: [bool] indicates whether the instance value is deprecated

\$comment: [str] comments to editors of the schema, not to users

const: [dict] used to restrict a value to a single value in all instances

properties: [dict] the key-value pairs expected in the object

patternProperties: [dict] expected value characteristics if the key matches a defined pattern

additionalProperties: [bool] whether or not to allow keys that are not defined in properties and do not match any patterns in patternProperties

required: [list] list of keywords required in the object

allOf: [dict] the object must be valid against all given subschemas

unevaluatedProperties: [bool] whether or not to allowed keys in subschemas that do not match properties or patternProperties

propertyNames: [dict] patterns to validate the keys against

minProperties: [int] minimum number of properties on the object

maxProperties: [int] maximum number of properties on the object

2.1.7 Abstract type: string

The basic string data type in the JSON schema.

Specifications for: string

Type: abstract

Allowed Keywords:

title: [str] the name of the string

description: [str] a description of the string

default: [dict] default instance of the string

examples: [list] a list of example string instances that validate correctly

readOnly: [bool] indicates to an API that the value should not be modified

writeOnly: [bool] indicates to an API that the value should remain hidden

deprecated: [bool] indicates whether the instance value is deprecated

\$comment: [str] comments to editors of the schema, not to users

const: [dict] used to restrict a value to a single value in all instances

enum: [dict] used to restrict a value to a certain set of values

minLength: [int] minimum length of the string

maxLength: [int] maximum length of the string

pattern: [str] used to restrict the string to a specific regex pattern

format: [str] used to restrict the string to a specific commonly-used pattern

2.2 General Datatypes

While the majority of the information specified in this document represents meta-data that is specialized to the category (e.g. sample material), there are some types of information that show up all over the file. This includes numerical values, energies, links and references. For these, general containers should be defined that can act as a basic unit and can be used anywhere in the schema. The following general containers should be supported in the database.

Section Contents:

2.2.1	General container: text	16
2.2.2	General container: text array	16
2.2.3	General container: number	16
2.2.4	General container: numerical array	17
2.2.5	General container: isotope	17
2.2.6	General container: link	18
2.2.7	General container: uncertainty	18
2.2.8	General container: measured quantity	19
2.2.9	General container: reference	20
2.2.10	General container: correction	20
2.2.11	General container: commercial product	21

2.2.1 General container: text

A data type to hold free text, in a single string. Values can be enumerated.

Specifications for: text
<p>abstract: object</p> <p>properties:</p> <ul style="list-style-type: none">attributes: [object] Attributes of text container. Properties:<ul style="list-style-type: none">source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.description: [string] Any free-text description available for the information in the container.body: [string, required] The free text.

2.2.2 General container: text array

Data type for a list of text values. Values can be enumerated.

Specifications for: text array
<p>abstract: object</p> <p>properties:</p> <ul style="list-style-type: none">attributes: [object] Attributes of the text array container. Properties:<ul style="list-style-type: none">source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.description: [string] Any free-text description available for the information in the container.body: [array, required] Body of the text array container.

2.2.3 General container: number

A data type to hold a single number.

Specifications for: number

abstract: object**properties:****attributes:** [object] Attributes of text container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [number, required] The number.

2.2.4 General container: numerical array

A list of numerical values, any combination of integers or floats.

Specifications for: numerical array

abstract: object**properties:****attributes:** [object] Attributes of array container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [array, required] The numerical array.

2.2.5 General container: isotope

A data type to hold the identifier of an isotope and excited level, if applicable. For an isotope and specific excited level, the format is ZZZ-XX-AAA-LL, where ZZZ is the three digit atomic number, AAA is the three digit mass number, XX is the chemical symbol, and LL is the three digit excited level number, with the ground state defined as level zero. When there is not a specific excited level, the form ZZZ-XX-AAA should be used. For a natural abundance element, the form ZZZ-AA-000 should be used. Particles should be written in the same format for consistency. Neutrons can be written as 000-n-0001 and gammas as 000-g-000.

Specifications for: isotope

abstract: object**properties:****attributes:** [object] Attributes of isotope container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required] The isotope in ZZZ-XX-AAA-LL format.

2.2.6 General container: link

In many cases the file will refer to information outside of the entry, such as DOI's for papers or references to data in other entries. For these, a link container should be created. The information in this container should allow the user to find the extra information easily. The link can be a URL, a library, or a Nuclear Science References (NSR) link.

Specifications for: link

abstract: object**properties:****attributes:** [object, required] Attributes of the link container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**link type:** [string, required, enumerated] The type of link.**body:** [string, required] The link.

2.2.7 General container: uncertainty

This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

Specifications for: uncertainty

abstract: object

properties:

attributes: [object, required] Attributes of uncertainty container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

unit: [string, required, enumerated] The unit.

unc type: [string, required, enumerated] The type of uncertainty.

body: [array, required] The value of the uncertainty.

2.2.8 General container: measured quantity

A data type to hold a measured quantity, which is a measured value and its uncertainty. When an uncertainty is included, it is assumed to be the total uncertainty on the value.

Specifications for: measured quantity

abstract: object

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

2.2.9 General container: reference

A data type to hold a single reference.

Specifications for: reference

abstract: object

properties:

attributes: [object, required] Attributes of reference container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

reference type: [string, enumerated] The type of reference.

content: [object, required] The contents of the reference container. Properties:

formatted: [string] The reference formatted in any bibliographic style.

link: [object] In many cases the file will refer to information outside of the entry, such as DOI's for papers or references to data in other entries. For these, a link container should be created. The information in this container should allow the user to find the extra information easily. The link can be a URL, a library, or a Nuclear Science References (NSR) link.

description: [string] Any free-text description available for the information in the container.

2.2.10 General container: correction

Corrections can be applied during data analysis for any of the data categories described above. All of the corrections can be described using a general correction data container.

Specifications for: correction

abstract: object

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

2.2.11 General container: commercial product

A data type to hold information about a commercial product, such as the manufacturer and model number.

Specifications for: commercial product**abstract:** object**properties:****attributes:** [object] Attributes of commercial product container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**content:** [object, required] Content of the commercial product container. Properties:**manufacturer:** [string, required] The name of the manufacturer, or in house.**model number:** [string] The model number of the product**use:** [string] What the product is used for**description:** [string] Any free-text description available for the information in the container.

2.3 Observable

This section will describe the observable of the data set, which covers the reaction being measured, the type of measurement, and type of data presented. Several of the keys are directly based on EXFOR codes and their values, which can be found in the EXFOR dictionaries [and are described in the LEXFOR compilation manual \(Schwerer, 2015\)](#).

Section Contents:

2.3.1	Observable container: type	24
2.3.2	Observable container: incident particle	24
2.3.3	Observable container: target	25
2.3.4	Observable container: measured particle	25
2.3.5	Observable container: reaction	26
2.3.6	Observable container: residual nucleus	27
2.3.7	Observable container: mf	27
2.3.8	Observable container: mt	28
2.3.9	Observable container: exfor process	29
2.3.10	Observable container: exfor branch	30
2.3.11	Observable container: exfor parameter	31
2.3.12	Observable container: exfor particle considered	32
2.3.13	Observable container: exfor modifier	32
2.3.14	Observable container: exfor data code	33
2.3.15	Observable container: absolute	34
2.3.16	Observable container: polarized	34
2.3.17	Observable container: observable method	35
2.3.18	Observable container: analysis method	36
2.3.19	Observable container: reference frame	36
2.3.20	Observable container: measured dependency	37
2.3.21	Observable container: incident energy	38
2.3.22	Observable container: incident angle	39
2.3.23	Observable container: exit dependency	40
2.3.24	Observable container: exit energy	41
2.3.25	Observable container: exit angle	41
2.3.26	Observable container: exit mass	42
2.3.27	Observable container: normalization	43
2.3.28	Observable container: background correction	44
2.3.29	Observable container: data	46
2.3.30	Observable container: statistical uncertainty	47
2.3.31	Observable container: systematic uncertainty	48
2.3.32	Observable container: unknown uncertainty	49

2.3.1 Observable container: type

A data type to hold the type of observable measured. The values are currently based on the EXFOR quantity field, the values of which are given in Dictionary 236.

Specifications for: type

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required, enumerated] The type of observable.

JSON example of type

```
"type": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "CS"
}
```

2.3.2 Observable container: incident particle

A data type to hold the incident particle of the reaction.

Specifications for: incident particle

type: isotope

properties:

attributes: [object] Attributes of isotope container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required] The isotope in ZZZ-XX-AAA-LL format.

JSON example of incident particle

```

"incident_particle": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "000-n-001"
}

```

2.3.3 Observable container: target

A data type to hold the target of the reaction.

Specifications for: target

type: isotope

properties:

attributes: [object] Attributes of isotope container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required] The isotope in ZZZ-XX-AAA-LL format.

JSON example of target

```

"target": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "006-C-012"
}

```

2.3.4 Observable container: measured particle

A data type to hold the type of particle measured in the experiment.

Specifications for: measured particle

type: isotope**properties:****attributes:** [object] Attributes of isotope container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required] The isotope in ZZZ-XX-AAA-LL format.

JSON example of measured particle

```

"measured_particle": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "000-g-000"
}

```

2.3.5 Observable container: reaction

A data type to hold the reaction measured, in reaction notation.

Specifications for: reaction

type: text**properties:****attributes:** [object] Attributes of text container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required] The reaction.

JSON example of reaction

```

"reaction": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "06-C-012(n,inl)006-C-012"
}

```

2.3.6 Observable container: residual nucleus

A data type to hold the residual nucleus left after the reaction, if applicable. This can be used to specify the excitation of the product nucleus for reactions such as inelastic scattering to specific levels or reactions leaving the product nucleus in an isomeric excited level. For reactions without a single product nucleus (such as fission), this key should not be used.

Specifications for: residual nucleus

type: isotope

properties:

attributes: [object] Attributes of isotope container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required] The isotope in ZZZ-XX-AAA-LL format.

JSON example of residual nucleus

```

"residual_nucleus": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "028-Ni-058-L003"
}

```

2.3.7 Observable container: mf

A data type to hold the analogous MF number, if applicable. [The MF number is a label used in the ENDF-6 format to indicate the type of data \(Herman & Trkov, 2010\)](#). If the measured [reaction observable](#) does not have a single analogous MF number, this key should not be used.

Specifications for: mf

type: text**properties:****attributes:** [object] Attributes of text container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required, enumerated] The MF value.

JSON example of mf

```

"mf": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "3"
}

```

2.3.8 Observable container: mt

A data type to hold the analogous MT number, if applicable, which can be found at the NNDC website. [The MT number is a label used in the ENDF-6 format to indicate the reaction \(Herman & Trkov, 2010\)](#). If the measured reaction does not have a single analogous MT number, this key should not be used.

Specifications for: mt

type: text**properties:****attributes:** [object] Attributes of text container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required, enumerated] The MT value.

JSON example of mt

```

"mt": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "102"
}

```

2.3.9 Observable container: exfor process

A data type to hold the process, which is the 3rd [quantity sub-field](#) in the [EXFOR](#) reaction string. This quantity describes the type of reaction or outgoing particles. The value can be a particle code in Dictionary 033, a nuclide code, or a process code in Dictionary 030, or any combination (with + as a separator) so value of this data type is not enumerated. Some examples are taken from the EXFOR Formats Manual(Otsuka, 2011).

Specifications for: exfor process

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required] The EXFOR process value.

JSON example of exfor process

```

"exfor_process": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "ABS"
}

```

JSON example of exfor process

```
"exfor_process": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "Emission of 3He and two 6Li"
  },
  "body": "HE3+3-LI-6"
}
```

JSON example of exfor process

```
"exfor_process": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "Emission of proton and anything"
  },
  "body": "P+X"
}
```

JSON example of exfor process

```
"exfor_process": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "Inelastic scattering followed by fission"
  },
  "body": "N+F"
}
```

2.3.10 Observable container: exfor branch

A data type to hold the branch, which is the 5th [quantity sub-field](#) in the [EXFOR](#) reaction string. This key can be used to indicate that a partial channel or cumulative reaction is measured. These values are given in [Dictionary 031](#).

Specifications for: exfor branch

type: text**properties:****attributes:** [object] Attributes of text container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required, enumerated] The EXFOR branch value.

JSON example of exfor branch

```

"exfor_branch": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "PAR"
}

```

2.3.11 Observable container: exfor parameter

A data type to hold the parameter, which is the 6th [quantity sub-field](#) in the EXFOR reaction string. This quantity describes the type of data presented, [such as cross section](#). These values are given in Dictionary 032.

Specifications for: exfor parameter

type: text**properties:****attributes:** [object] Attributes of text container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required, enumerated] The EXFOR parameter value.

JSON example of exfor parameter

```

"exfor_parameter": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "SIG"
}

```

2.3.12 Observable container: exfor particle considered

A data type to hold the particle considered, which is the 7th [quantity sub-field](#) in the EXFOR reaction string. The values can be particle codes from Dictionary 033 or nuclide codes.

Specifications for: exfor particle considered

type: isotope

properties:

attributes: [object] Attributes of isotope container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required] The EXFOR particle considered value.

JSON example of exfor particle considered

```

"exfor_particle_considered": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "000-g-000"
}

```

2.3.13 Observable container: exfor modifier

A data type to hold the modifier, which is the 8th [quantity sub-field](#) in the EXFOR reaction string. This quantity holds information about how the data is represented. These values are given in Dictionary 034.

Specifications for: exfor modifier

type: text**properties:****attributes:** [object] Attributes of text container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required, enumerated] The EXFOR modifier value

JSON example of exfor modifier

```

"exfor_modifier": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "REL"
}

```

2.3.14 Observable container: exfor data code

A data type to hold the data code, which is the 9th [quantity sub-field](#) in the EXFOR reaction string. This quantity holds information about whether the data are experimental or not. These values are given in Dictionary 35.

Specifications for: exfor data code

type: text array**properties:****attributes:** [object] Attributes of the text array container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [array, required, enumerated] The EXFOR data code value

JSON example of exfor data code

```

"exfor_data_code": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": ["EXP"]
}

```

2.3.15 Observable container: absolute

A data type to hold whether or not the measurement is presented in the units of the observable (such as presenting cross sections in barns). If false, the data are shape data and can float with respect to the normalization. For instance, all PFNS measurement provide shape values and, hence, “no” applies automatically.

Specifications for: absolute

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required, enumerated] Whether the data is absolute.

JSON example of absolute

```

"absolute": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "no"
}

```

2.3.16 Observable container: polarized

A data type to hold whether or not the incident beam is polarized.

Specifications for: polarized

type: text**properties:****attributes:** [object] Attributes of text container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required, enumerated] Whether the incident source is polarized.

JSON example of polarized

```

"polarized": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "no"
}

```

2.3.17 Observable container: observable method

A data type to hold the EXFOR method codes keywords, which are given in Dictionary 021. Free-text descriptions that are given with the code can be put into the description attribute. The example is taken from EXFOR entry 14463.

Specifications for: observable method

type: text array**properties:****attributes:** [object] Attributes of the text array container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [array, required, enumerated] The observable method value.

JSON example of observable method

```

"observable_method": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "The timing structure of proton and neutron
pulses is a 625 mu-s long macropulse consisting of micropulses with
1.8 mu-s separation and widths of about 250 ps. The macropulse repetition
rate was 40 Hz. The available neutron energy range is typically hundreds
of keV to hundreds of MeV. The flight-path length was determined to be
L = 9.596 +- 0.015 m."
  },
  "body": ["TOF"]
}

```

2.3.18 Observable container: analysis method

A data type to hold the EXFOR analysis codes, which are given in Dictionary 023. Free-text descriptions that are given with the code can be put into the description attribute.

Specifications for: analysis method

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required, enumerated] The analysis method value.

JSON example of analysis method

```

"analysis_method": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": ["SURGT"]
}

```

2.3.19 Observable container: reference frame

A data type to hold the reference frame the measurement results are presented in.

Specifications for: reference frame

type: text**properties:****attributes:** [object] Attributes of text container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required, enumerated] The reference frame of the data.

JSON example of reference frame

```

"reference_frame": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "laboratory"
}

```

2.3.20 Observable container: measured dependency

A data type to hold what the incident particle is a function of.

Specifications for: measured dependency

type: text**properties:****attributes:** [object] Attributes of text container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required, enumerated] The measured dependency.

JSON example of measured dependency

```

"measured_dependency": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "energy"
}

```

2.3.21 Observable container: incident energy

A data type to hold the value or list of values for the incident particle energy. The example is taken from EXFOR entry 23605.

Specifications for: incident energy

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of incident energy

```

"incident_energy": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "eV",
    "value_type": "mean"
  },
  "value": [449.196, 448.278, 447.364, 446.452, 445.543, 444.637]
}

```

2.3.22 Observable container: incident angle

A data type to hold the value of list of values for the incident particle angle.

Specifications for: incident angle

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of incident angle

```

"incident_angle": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "degree",
    "type": "mean"
  },
  "value": [90],
  "uncertainty": {
    "attributes": {
      "source": ["EXFOR"],
      "unit": "degree",
      "type": "absolute"
    },
    "body": [2]
  }
}

```

2.3.23 Observable container: exit dependency

A data type to hold what the exit particle is a function of.

Specifications for: exit dependency

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required, enumerated] The exit dependency.

JSON example of exit dependency

```

"exit_dependency": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "mass"
}

```

2.3.24 Observable container: exit energy

A data type to hold the value or list of values for the exit particle energy.

Specifications for: exit energy

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of exit energy

```
"exit_energy": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "eV",
    "type": "mean"
  },
  "value": [846.0]
}
```

2.3.25 Observable container: exit angle

A data type to hold the value or list of values for the exit particle angle.

Specifications for: exit angle

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of exit angle

```
"exit_angle": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "degree",
    "type": "mean"
  },
  "value": [90],
  "uncertainty": {
    "attributes": {
      "source": ["EXFOR"],
      "unit": "degree",
      "type": "resolution"
    },
    "body": [5]
  }
}
```

2.3.26 Observable container: exit mass

A data type to hold the value or list of values for the exit particle mass.

Specifications for: exit mass

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, Or year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of exit mass

```
"exit_mass": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "u",
    "type": "range"
  },
  "value": [238, 240]
}
```

2.3.27 Observable container: normalization

A data type to hold information about an overall normalization on the data set. In many cases, an uncertainty due to a normalization is provided without providing the normalization values themselves. The example is taken from EXFOR entry 30423.

Specifications for: normalization

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, Or year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of normalization

```
"normalization": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "uncertainty": {
    "attributes": {
      "source": ["EXFOR"],
      "unit": "unitless",
      "unc_type": "relative"
    },
    "body": [5.0]
  },
  "description": "The absolute error in normalization was
estimated to be about 5%."
}
```

2.3.28 Observable container: background correction

A data type to hold information about a correction for an overall background that does not belong in any other category. The example is taken from EXFOR entry 30340.

Specifications for: background correction**type:** correction**properties:****attributes:** [object, required] Attributes of correction container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**applied:** [string, required, enumerated] Whether or not the correction was applied.**energy dependent:** [string, required, enumerated] Whether or not the correction is energy-dependent.**content:** [object, required] The contents of the correction container. Properties:**methods:** [array, enumerated] The methods used to perform the correction.**components:** [array, enumerated] The components of the experimental setup that were included in the correction.**code:** [array, enumerated] The primary code(s) used to perform or calculate the correction.**reference:** [object] A data type to hold a single reference.**magnitude:** [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.**description:** [string] Any free-text description available for the information in the container.

JSON example of background correction

```
"background_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "magnitude": {
    "attributes": {
      "value_type": "maximum",
      "unit": "percent"
    },
    "value": [5]
  },
  "description": "For determination of background (15-30%) the
transmission of a Cu sample was measured. For the measurement with t(p,n)
neutrons background could be reduced (less than 5%) using another sample
container setup. Multiple-scattering and in-scattering corrections were
not necessary since no absolute cross section determination was done."
}
```

2.3.29 Observable container: data

A data type to hold the value(s) presented of the observable. The uncertainty in this container should only be the total uncertainty on the data. The example is taken from EXFOR entry 23605.

Specifications for: data

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of data

```
"data": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "b",
    "value_type": "mean"
  },
  "value": [4.977, 5.798, 4.083, 5.672, 4.751, 5.969, 4.645],
  "uncertainty": {
    "attributes": {
      "source": ["EXFOR"],
      "unit": "b",
      "unc_type": "absolute"
    },
    "body": [0.915, 0.916, 0.917, 0.908, 0.907, 0.909, 0.922]
  }
}
```

2.3.30 Observable container: statistical uncertainty

A data type to hold the total statistical uncertainty on the data values, if it is reported. The example is taken from EXFOR entry 23605.

Specifications for: statistical uncertainty

type: uncertainty

properties:

attributes: [object, required] Attributes of uncertainty container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

unit: [string, required, enumerated] The unit.

unc_type: [string, required, enumerated] The type of uncertainty.

body: [array, required] The value of the uncertainty.

JSON example of statistical uncertainty

```
"statistical_uncertainty": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "uncertainty": {
    "attributes": {
      "source": ["EXFOR"],
      "unit": "b",
      "unc_type": "absolute"
    },
    "body": [0.817, 0.817, 0.821, 0.812, 0.813, 0.813, 0.823]
  }
}
```

2.3.31 Observable container: systematic uncertainty

A data type to hold the total systematic uncertainty on the data values, if it is reported. The example is taken from EXFOR entry 23605.

Specifications for: systematic uncertainty

type: uncertainty

properties:

attributes: [object, required] Attributes of uncertainty container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

unit: [string, required, enumerated] The unit.

unc type: [string, required, enumerated] The type of uncertainty.

body: [array, required] The value of the uncertainty.

JSON example of systematic uncertainty

```
"systematic_uncertainty": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "uncertainty": {
    "attributes": {
      "source": ["EXFOR"],
      "unit": "b",
      "unc_type": "absolute"
    },
    "body": [0.411, 0.415, 0.41, 0.405, 0.404, 0.407, 0.414]
  }
}
```

2.3.32 Observable container: unknown uncertainty

A data type to hold unknown type of uncertainty on the data values, if reported.

Specifications for: unknown uncertainty

type: uncertainty

properties:

attributes: [object, required] Attributes of uncertainty container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

unit: [string, required, enumerated] The unit.

unc_type: [string, required, enumerated] The type of uncertainty.

body: [array, required] The value of the uncertainty.

JSON example of unknown uncertainty

```
"unknown_uncertainty": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "uncertainty": {
    "attributes": {
      "source": ["EXFOR"],
      "unit": "percent",
      "unc_type": "relative"
    },
    "body": [1.4]
  }
}
```

2.4 Bibliography

This section will describe the sources and references for the data set. This includes entry number in this database and in the EXFOR database, the main and supplemental references, the compilation history, and the institute that produced the data. Several of the keys are directly based on EXFOR codes and their values, which can be found in the EXFOR dictionaries [and are described in the LEXFOR compilation manual \(Schwerer, 2015\)](#).

Section Contents:

2.4.1	Bibliography container: entry number	52
2.4.2	Bibliography container: authors	52
2.4.3	Bibliography container: institute	53
2.4.4	Bibliography container: main reference	53
2.4.5	Bibliography container: exfor number	54
2.4.6	Bibliography container: history	55
2.4.7	Bibliography container: comments	56
2.4.8	Bibliography container: critique	56
2.4.9	Bibliography container: status	57
2.4.10	Bibliography container: alteration	58

2.4.1 Bibliography container: entry number

A data type to hold the database entry number for the dataset.

Specifications for: entry number

type: number

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [number, required] The entry number.

JSON example of entry number

```
"entry_number": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": 1000
}
```

2.4.2 Bibliography container: authors

A data type to hold the authors on the main reference, which are used to identify the data set. The example is taken from EXFOR entry 10001.

Specifications for: authors

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required] List of authors.

JSON example of authors

```

"authors": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": ["R.W.Hockenbury", "Z.M.Bartolome", "J.R.Tatarczuk", "..."]
}

```

2.4.3 Bibliography container: institute

A data type to hold the EXFOR institute codes keyword, which are given in Dictionary 003. The example is taken from EXFOR entry 13801.

Specifications for: institute

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required, enumerated] List of institutes.

JSON example of institute

```

"institute": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "WNR facility at LAMPF"
  },
  "body": ["1USALAS"]
}

```

2.4.4 Bibliography container: main reference

A data type to hold the main reference for the data set. The example is taken from EXFOR entry 12898.

Specifications for: main reference

type: reference

properties:

attributes: [object, required] Attributes of reference container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

reference type: [string, enumerated] The type of reference.

content: [object, required] The contents of the reference container. Properties:

formatted: [string] The reference formatted in any bibliographic style.

link: [object] In many cases the file will refer to information outside of the entry, such as DOI's for papers or references to data in other entries. For these, a link container should be created. The information in this container should allow the user to find the extra information easily. The link can be a URL, a library, or a Nuclear Science References (NSR) link.

description: [string] Any free-text description available for the information in the container.

JSON example of main reference

```

"main_reference": {
  "attributes": {
    "source": ["EXFOR"],
    "type": "journal"
  },
  "formatted": "D.L.Smith, J.W.Meadows, I.Kanno, Measured Activation
Cross Sections below 10 MeV for the 51V(n,p)51Ti and 51V(n,a)48Sc
Reactions, Annals of Nuclear Energy, 11 (1984)",
  "link": {
    "attributes": {
      "source": ["EXFOR"],
      "type": "DOI"
    },
    "body": "10.1016/0306-4549(84)90031-8"
  }
}

```

2.4.5 Bibliography container: exfor number

A data type to hold the EXFOR [subentry accession](#) number for the dataset.

Specifications for: exfor number

type: number**properties:****attributes:** [object] Attributes of text container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [number, required] The EXFOR dataset number.

JSON example of exfor number

```

"exfor_number": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": 12898002
}

```

2.4.6 Bibliography container: history

A data type to hold the the EXFOR HISTORY keyword. The codes for this keyword are not enumerated, so the code should be used instead of the description. The example is taken from EXFOR entry 12898.

Specifications for: history

type: text**properties:****attributes:** [object] Attributes of text container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required] HISTORY code

JSON example of history

```

"history": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "REFERENCE UPDATE"
  },
  "body": "850212A"
}

```

2.4.7 Bibliography container: comments

A data type to hold the EXFOR COMMENTS keyword, which is entirely free text. The example is taken from EXFOR entry 12898.

Specifications for: comments

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required] List of comments.

JSON example of comments

```

"comments": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": ["SEE DATA SET 12904 FOR 51-V(N,ALPHA)."]
}

```

2.4.8 Bibliography container: critique

A data type to hold the EXFOR CRITIQUE keyword, which is entirely free text. The example is taken from EXFOR entry A1402.

Specifications for: critique

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required] List of critiques.

JSON example of critique

```
"critique": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": ["The errors were not discussed at the paper."]
}
```

2.4.9 Bibliography container: status

A data type to hold the EXFOR EXFOR status codes keyword, which are given in Dictionary 016. The example is taken from EXFOR entry 12898.

Specifications for: status

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required, enumerated] List of status values.

JSON example of status

```

"status": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "D.L.Smith, 84/10/3."
  },
  "body": ["APRVD"]
}

```

2.4.10 Bibliography container: alteration

A data type to hold a type of alteration, if the data in the entry is different from the data given in the main reference.

Specifications for: alteration

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required, enumerated] List of alterations.

JSON example of alteration

```

"alteration": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": ["rebinned"]
}

```

2.5 Incident

This section will describe the incident particles used for the data set. This includes information about the particles, the facility, the flux, and the spectrum. Several of the keys are directly based on EXFOR codes and their values, which can be found in the EXFOR dictionaries [and are described in the LEXFOR compilation manual \(Schwerer, 2015\)](#).

Section Contents:

2.5.1 Incident container: facility	60
2.5.2 Incident container: particle source	60
2.5.3 Incident container: reaction	61
2.5.4 Incident container: energy calibration	62
2.5.5 Incident container: fluence method	63
2.5.6 Incident container: fluctuation correction	64
2.5.7 Incident container: uniformity correction	65
2.5.8 Incident container: secondary group correction	67
2.5.9 Incident container: beam stability monitored	69
2.5.10 Incident container: accelerator stability monitored	70
2.5.11 Incident container: flight path	70
2.5.12 Incident container: pulse width	72
2.5.13 Incident container: micro pulse width	73
2.5.14 Incident container: repetition rate	74
2.5.15 Incident container: zero time offset	75
2.5.16 Incident container: beam energy	76
2.5.17 Incident container: beam power	77
2.5.18 Incident container: beam filters	78
2.5.19 Incident container: collimators	79
2.5.20 Incident container: moderators	80
2.5.21 Incident container: associated particle detected	81
2.5.22 Incident container: target thickness	81
2.5.23 Incident container: number target atoms	82
2.5.24 Incident container: stable target	83
2.5.25 Incident container: gas cell pressure	84
2.5.26 Incident container: gas cell temperature	85
2.5.27 Incident container: gas cell temperature correction	86
2.5.28 Incident container: gas cell effects correction	87
2.5.29 Incident container: anisotropy correction	88
2.5.30 Incident container: fluence values	90
2.5.31 Incident container: current values	91

2.5.1 Incident container: facility

A data type to hold the EXFOR facility codes keywords, which are given in Dictionary 018. Free-text descriptions that are given with the code can be put into the description attribute. The example is taken from EXFOR entry 13801.

Specifications for: facility

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required, enumerated] List of the facility or facilities where the measurement was performed.

JSON example of facility

```
"facility": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "WNR facility at LAMPF"
  },
  "body": ["LINAC"]
}
```

2.5.2 Incident container: particle source

A data type to hold the EXFOR inc-source codes keywords, which are given in Dictionary 019. Free-text descriptions that are given with the code can be put into the description attribute. The examples are taken from EXFOR entries 13801 and 14576.

Specifications for: particle source

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required, enumerated] The incident source for the measurement.

JSON example of particle source

```
"particle_source": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "800-MeV protons on tungsten."
  },
  "body": "SPALL"
}
```

JSON example of particle source

```
"particle_source": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "Neutrons were produced through a (gamma,n)
reaction when electrons from the RPI 60-MeV electron linac interact with
a tantalum target."
  },
  "body": "PHOTO"
}
```

2.5.3 Incident container: reaction

A data type to hold the reaction used to produce the incident particles, if the `incident_source` keyword does not specify it. This can be used in cases such as photo-neutrons, where the incident source code is “photoneutron”, where this key can be used to document the reaction producing the neutrons, e.g. $\text{Ta}(\gamma, n)$. The example is taken from EXFOR entry 14576.

Specifications for: reaction

type: text**properties:****attributes:** [object, required] Attributes of text container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required] The incident reaction in reaction notation of target(incident,outgoing)product

JSON example of reaction

```

"reaction": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "Neutrons were produced through a
(gamma,n) reaction when electrons from the RPI 60-MeV electron linac
interact with a tantalum target"
  },
  "body": "073-Ta-000(g,n)"
}

```

2.5.4 Incident container: energy calibration

A data type to hold the method or methods used to calibrate the energy of the incident particles. The example is taken from EXFOR entry A0585.

Specifications for: energy calibration

type: text array**properties:****attributes:** [object] Attributes of the text array container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [array, required, enumerated] List of energy calibration methods.

JSON example of energy calibration

```

"energy_calibration": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "The terminal voltage of the accelerator
was calibrated with the well-known resonances of the 27Al(p,g)28Si
reaction at 992 and 1380 KeV. The energy spread of the beam is below 1
keV for protons at 1000 KeV and is assumed to be of the same order for
Li ions, which are produced with the same source."
  },
  "body": ["known_resonance"]
}

```

2.5.5 Incident container: fluence method

A data type to hold the method used to determine the magnitude of the fluence, if applicable. For experiments where the magnitude of the fluence was not explicitly measured, this key should not be used. This key should not be used for experiments where the fluctuations of the incident particle fluence were measured as a way to normalize different runs, but the magnitude itself was not determined. The examples are taken from EXFOR entries 20786 and 22304.

Specifications for: fluence method

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required, enumerated] List of methods used.

JSON example of fluence method

```

"fluence_method": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "Fission counter at the 11-m flight path
and the flux detectors at the 57-m station in the same neutron beam."
  },
  "body": ["direct_measurement"]
}

```

JSON example of fluence method

```

"fluence_method": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "(TELES,SIBAR,SIBAR) Associated particle
detection"
  },
  "body": ["associated_particle"]
}

```

2.5.6 Incident container: fluctuation correction

A data type to hold information about a correction applied for fluctuations in the incident particle flux between runs or within a single run. The example is taken from EXFOR entry 41335.

Specifications for: fluctuation correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of fluctuation correction

```
"fluctuation_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "magnitude": {
    "attributes": {
      "value_type": "maximum",
      "unit": "percent"
    },
    "body": [0.5]
  },
  "description": "For neutron beam fluctuations < 0.5%."
}
```

2.5.7 Incident container: uniformity correction

A data type to hold information about a correction for the uniformity of the beam across the face of the sample. The example is taken from EXFOR entry 10950.

Specifications for: uniformity correction**type:** correction**properties:****attributes:** [object, required] Attributes of correction container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**applied:** [string, required, enumerated] Whether or not the correction was applied.**energy dependent:** [string, required, enumerated] Whether or not the correction is energy-dependent.**content:** [object, required] The contents of the correction container. Properties:**methods:** [array, enumerated] The methods used to perform the correction.**components:** [array, enumerated] The components of the experimental setup that were included in the correction.**code:** [array, enumerated] The primary code(s) used to perform or calculate the correction.**reference:** [object] A data type to hold a single reference.**magnitude:** [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.**description:** [string] Any free-text description available for the information in the container.

JSON example of uniformity correction

```
"uniformity_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "no"
  },
  "magnitude": {
    "attributes": {
    },
    "uncertainty": {
      "attributes": {
        "unit": "percent",
        "unc_type": "relative"
      },
      "body": [0.3]
    }
  },
  "description": "Neutron beam non-uniformity (0.3%)"
}
```

2.5.8 Incident container: secondary group correction

A data type to hold information about a correction for secondary particle energies in the flux. The example is taken from EXFOR entry 14721.

Specifications for: secondary group correction**type:** correction**properties:****attributes:** [object, required] Attributes of correction container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**applied:** [string, required, enumerated] Whether or not the correction was applied.**energy dependent:** [string, required, enumerated] Whether or not the correction is energy-dependent.**content:** [object, required] The contents of the correction container. Properties:**methods:** [array, enumerated] The methods used to perform the correction.**components:** [array, enumerated] The components of the experimental setup that were included in the correction.**code:** [array, enumerated] The primary code(s) used to perform or calculate the correction.**reference:** [object] A data type to hold a single reference.**magnitude:** [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.**description:** [string] Any free-text description available for the information in the container.

JSON example of secondary group correction

```

"secondary_group_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "yes"
  },
  "magnitude": {
    "attributes": {
      "source": ["EXFOR"],
      "value_type": "mean",
      "unit": "dimensionless"
    },
    "value": [0.38, 0.28, 0.24]
  },
  "description": "Following corrections were applied
- wraparound correction
- contamination correction
- overlap correction
- beam attenuation and scattering correction
(ERR-2) Uncertainty of the wraparound correction"
}

```

2.5.9 Incident container: beam stability monitored

A data type to hold whether or not the beam stability was monitored during the experiment.

Specifications for: beam stability monitored

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required, enumerated] Whether the beam stability was monitored.

JSON example of beam stability monitored

```

"beam_stability_monitored": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "unknown"
}

```

2.5.10 Incident container: accelerator stability monitored

A data type to hold whether or not the accelerator stability was monitored during the experiment.

Specifications for: accelerator stability monitored

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required, enumerated] Whether the accelerator stability was monitored.

JSON example of accelerator stability monitored

```

"accelerator_stability_monitored": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "unknown"
}

```

2.5.11 Incident container: flight path

A data type to hold information about the length of the flight path from the incident particle production to the sample. The example is taken from EXFOR entry 14463.

Specifications for: flight path

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, Or year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of flight path

```
"flight_path": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "m",
    "value_type": "mean"
  },
  "value": [9.596],
  "uncertainty": {
    "attributes": {
      "source": ["EXFOR"],
      "unit": "m",
      "unc_type": "absolute"
    },
    "body": [0.015]
  },
  "description": "The timing structure of proton and neutron
pulses is a 625 mu-s long macropulse consisting of micropulses with
1.8 mu-s separation and widths of about 250 ps. The macropulse repetition
rate was 40 Hz. The available neutron energy range is typically hundreds
of keV to hundreds of MeV. The flight-path length was determined to be
L = 9.596 +- 0.015 m."
}
```

2.5.12 Incident container: pulse width

A data type to hold the value of the pulse width of the accelerator. The example is taken from EXFOR entry 14463.

Specifications for: pulse width

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of pulse width

```
"pulse_width": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "s",
    "value_type": "mean"
  },
  "value": [0.000625],
  "description": "The timing structure of proton and neutron
pulses is a 625 mu-s long macropulse consisting of micropulses with
1.8 mu-s separation and widths of about 250 ps. The macropulse repetition
rate was 40 Hz. The available neutron energy range is typically hundreds
of keV to hundreds of MeV. The flight-path length was determined to be
L = 9.596 +- 0.015 m."
}
```

2.5.13 Incident container: micro pulse width

A data type to hold the micro-pulse width of the accelerator. The example is taken from EXFOR entry 14463.

Specifications for: micro pulse width

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, Or year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of micro pulse width

```
"micro_pulse_width": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "s",
    "value_type": "mean"
  },
  "value": [2.5e-10],
  "description": "The timing structure of proton and neutron
pulses is a 625 mu-s long macropulse consisting of micropulses with
1.8 mu-s separation and widths of about 250 ps. The macropulse repetition
rate was 40 Hz. The available neutron energy range is typically hundreds
of keV to hundreds of MeV. The flight-path length was determined to be
L = 9.596 +- 0.015 m."
}
```

2.5.14 Incident container: repetition rate

A data type to hold information about the repetition rate of the accelerator. The example is taken from EXFOR entry 14463.

Specifications for: repetition rate

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, Or year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of repetition rate

```
"repetition_rate": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "Hz",
    "value_type": "mean"
  },
  "value": [40],
  "description": "The timing structure of proton and neutron
pulses is a 625 mu-s long macropulse consisting of micropulses with
1.8 mu-s separation and widths of about 250 ps. The macropulse repetition
rate was 40 Hz. The available neutron energy range is typically hundreds
of keV to hundreds of MeV. The flight-path length was determined to be
L = 9.596 +- 0.015 m."
}
```

2.5.15 Incident container: zero time offset

A data type to hold information about the zero time offset for time-of-flight measurements.

Specifications for: zero time offset

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of zero time offset

```
"zero_time_offset": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "s",
    "value_type": "mean"
  },
  "value": [5e-09]
}
```

2.5.16 Incident container: beam energy

A data type to hold information about the energy of the charged particles in the accelerator. The example is taken from EXFOR entry 14576.

Specifications for: beam energy

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, Or year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of beam energy

```
"beam_energy": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "eV",
    "value_type": "mean"
  },
  "value": [53000000.0],
  "description": "Neutrons were produced through a (gamma,n)
reaction when electrons from the RPI 60-MeV electron linac interact with
a tantalum target
Nominal beam energy      53 MeV
Nominal beam power      330 W
Neutron target          bare bounce target,no moderator"
}
```

2.5.17 Incident container: beam power

A data type to hold information about the power of the charged particles in the accelerator. The example is taken from EXFOR entry 14576.

Specifications for: beam power

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of beam power

```
"beam_power": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "W",
    "value_type": "mean"
  },
  "value": [330],
  "description": "Neutrons were produced through a (gamma,n)
reaction when electrons from the RPI 60-MeV electron linac interact with
a tantalum target
Nominal beam energy      53 MeV
Nominal beam power      330 W
Neutron target          bare bounce target,no moderator"
}
```

2.5.18 Incident container: beam filters

A data type to hold information about filters used in the beam to mitigate effects. This can include filters to reduce the gamma flash or secondary particle groups, and filters to modify the spectrum. The example is taken from EXFOR entry 14576.

Specifications for: beam filters

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required] List of beam filters.

JSON example of beam filters

```
"beam_filters": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "Electron linac was operated in pulse
mode, detectors were placed at a distance of 100m and 250m. Neutrons
migrated through evacuated neutron flight tubes and gamma rays were
filtered using depleted uranium. Neutrons were collimated using series of
lead, boron carbide, polyethylene, and iron collimators."
  },
  "body": ["depleted_uranium"]
}
```

2.5.19 Incident container: collimators

A data type to hold the collimators used in the beam. The example is taken from EXFOR entry 14576.

Specifications for: collimators

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required] List of collimators.

JSON example of collimators

```

"collimators": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "Electron linac was operated in pulse
mode, detectors were placed at a distance of 100m and 250m. Neutrons
migrated through evacuated neutron flight tubes and gamma rays were
filtered using depleted uranium. Neutrons were collimated using series of
lead, boron carbide, polyethylene, and iron collimators."
  },
  "body": ["lead", "boron carbide", "polyethylene", "iron"]
}

```

2.5.20 Incident container: moderators

A data type to hold the moderator materials, if used, for a thermal or epi-thermal measurement. The example is taken from EXFOR entry 41388.

Specifications for: moderators

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required] List of moderators.

JSON example of moderators

```

"moderators": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "Fast neutrons are moderated in a water
moderator and then transmitted through a vacuum tube to the experimental
installation."
  },
  "body": ["water"]
}

```

2.5.21 Incident container: associated particle detected

A data type to hold the type of particle that was detected to determine the incident neutron flux for associated particle measurements.

Specifications for: associated particle detected

type: isotope

properties:

attributes: [object] Attributes of isotope container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required] The isotope in ZZZ-XX-AAA-LL format.

JSON example of associated particle detected

```
"associated_particle_detected": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "001-H-001"
}
```

2.5.22 Incident container: target thickness

A data type to hold the thickness of the target used to produce the incident particles.

Specifications for: target thickness

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of target thickness

```
"target_thickness": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "m",
    "value_type": "mean"
  },
  "value": [0.005]
}
```

2.5.23 Incident container: number target atoms

A data type to hold information about the number of atoms in the target used to produce the incident particles.

Specifications for: number target atoms

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of number target atoms

```

"number_target_atoms": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "unitless",
    "value_type": "mean"
  },
  "value": [5.5e+23],
  "uncertainty": {
    "attributes": {
      "source": ["EXFOR"],
      "unit": "unitless",
      "unc_type": "absolute"
    },
    "body": [5.5e+21]
  }
}

```

2.5.24 Incident container: stable target

A data type to hold information about whether not the target used to produce the incident particles is stable or radioactive. For effectively stable targets, use yes.

Specifications for: stable target

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required, enumerated] Whether or not the target is stable.

JSON example of stable target

```
"stable_target": {  
  "attributes": {  
    "source": ["EXFOR"]  
  },  
  "body": "yes"  
}
```

2.5.25 Incident container: gas cell pressure

A data type to hold information about the pressure of the gas cell used in some incident particle production methods. The example is taken from EXFOR entry 23032.

Specifications for: gas cell pressure

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, Or year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of gas cell pressure

```
"gas_cell_pressure": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "Pa",
    "value_type": "mean"
  },
  "value": [160000.0],
  "description": "The gas cell is filled with D2 at 1.6E+5 Pa and
is separated from the vacuum of the accelerator by a 2.2-micrometer thick
Havar foil, which acts as a window."
}
```

2.5.26 Incident container: gas cell temperature

A data type to hold information about the temperature of the gas cell used in some incident particle production methods.

Specifications for: gas cell temperature

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, Or year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of gas cell temperature

```
"gas_cell_temperature": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "C",
    "value_type": "mean"
  },
  "value": [73]
}
```

2.5.27 Incident container: gas cell temperature correction

A data type to hold information about a correction for the temperature of the gas cell.

Specifications for: gas cell temperature correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of gas cell temperature correction

```
"gas_cell_temperature_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "negligible",
    "energy_dependent": "no"
  }
}
```

2.5.28 Incident container: gas cell effects correction

A data type to hold information about a correction for effects of the gas cell on the incident particle production. Such effects include leaks, punctures, and production of H gas within the gas cell.

Specifications for: gas cell effects correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of gas cell effects correction

```
"gas_cell_effects_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "negligible",
    "energy_dependent": "no"
  }
}
```

2.5.29 Incident container: anisotropy correction

A data type to hold information about a correction for anisotropy in the production of incident particles. The example is taken from EXFOR entry 10633.

Specifications for: anisotropy correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of anisotropy correction

```
"anisotropy_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "description": " Data corrected for:
. dead time,
. detector efficiency,
. attenuation,
. multiple scattering,
. finite geometry,
. neutron source anisotropy,
. contaminants."
}
```

2.5.30 Incident container: fluence values

A data type to hold the absolute value of the fluence (a single value or as a function of incident energy or angle) if given. For experiments that are done in ratio to a standard or a monitor reaction, this should not be used.

Specifications for: fluence values

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of fluence values

```
"fluence_values": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "uncertainty": {
    "attributes": {
      "source": ["EXFOR"],
      "unit": "percent",
      "unc_type": "relative"
    },
    "body": [5.0]
  }
}
```

2.5.31 Incident container: current values

A data type to hold the absolute value of the current (a single value or as a function of incident energy or angle), if given. For experiments that are done in ratio to a standard or a monitor reaction, this should not be used.

Specifications for: current values

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of current values

```
"current_values": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "uncertainty": {
    "attributes": {
      "source": ["EXFOR"],
      "unit": "percent",
      "unc_type": "relative"
    },
    "body": [5.0]
  }
}
```

2.6 Detector

The `detectors` container is a list of individual detector containers, each of which describe a single detector system, such as a detector array or a single-standing detector.

```
"detectors": [
    {detector container},
    {detector container}
]
```

The type keyword is based on the EXFOR detector keyword [which is described in the LEXFOR compilation manual \(Schwerer, 2015\)](#), and the values can be found in the EXFOR dictionaries.

Section Contents:

2.6.1	Detectors container: type	94
2.6.2	Detectors container: use	94
2.6.3	Detectors container: manufacturer	95
2.6.4	Detectors container: pmt	96
2.6.5	Detectors container: daq system	97
2.6.6	Detectors container: number detectors	98
2.6.7	Detectors container: geometry	99
2.6.8	Detectors container: volume	100
2.6.9	Detectors container: radius	101
2.6.10	Detectors container: throughtube diameter	102
2.6.11	Detectors container: thickness	103
2.6.12	Detectors container: particle detected	104
2.6.13	Detectors container: efficiency method	105
2.6.14	Detectors container: energy calibration method	105
2.6.15	Detectors container: calibration sources	106
2.6.16	Detectors container: detector shielding	107
2.6.17	Detectors container: solid angle	108
2.6.18	Detectors container: deadtime correction	109
2.6.19	Detectors container: live time	110
2.6.20	Detectors container: pulse pileup correction	111
2.6.21	Detectors container: discriminator correction	113
2.6.22	Detectors container: electronic noise correction	114
2.6.23	Detectors container: forward boost correction	115
2.6.24	Detectors container: angular distortion correction	117
2.6.25	Detectors container: delayed gamma correction	118
2.6.26	Detectors container: false fission correction	119
2.6.27	Detectors container: french effect correction	121
2.6.28	Detectors container: hs content correction	122
2.6.29	Detectors container: manganese capture correction	124
2.6.30	Detectors container: false capture correction	126
2.6.31	Detectors container: spectrum unfolding correction	127
2.6.32	Detectors container: sum peaks correction	128

2.6.33	Detectors container: gamma angular distribution correction . . .	130
2.6.34	Detectors container: counting geometry correction	131
2.6.35	Detectors container: coincidence measurement	133
2.6.36	Detectors container: time gate open	134
2.6.37	Detectors container: gas	135
2.6.38	Detectors container: gas pressure	136
2.6.39	Detectors container: reaction	137
2.6.40	Detectors container: stopping power correction	137
2.6.41	Detectors container: pulse height extrapolation correction . . .	139
2.6.42	Detectors container: efficiency	140
2.6.43	Detectors container: energy resolution	142
2.6.44	Detectors container: time resolution	143

2.6.1 Detectors container: type

A data type to hold the EXFOR detector codes keyword, which are given in Dictionary 022. The example is taken from EXFOR entry 14463.

Specifications for: type
<p>type: text</p> <p>properties:</p> <ul style="list-style-type: none"> attributes: [object] Attributes of text container. Properties: <ul style="list-style-type: none"> source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, Or year-name. description: [string] Any free-text description available for the information in the container. body: [string, required, enumerated] The type of detector.

JSON example of type
<pre> "type": { "attributes": { "source": ["EXFOR"], "description": "A twin Frisch-gridded ionization chamber was chosen for fission detection, mainly due to its high efficiency, good energy resolution (0.5 - 1.0%) and nearly 4PI solid angle coverage. The ionization chamber was operated with P10 fill gas (90% argon and 10% methane) at 1.25 kPa." }, "body": "IOCH" } </pre>

2.6.2 Detectors container: use

A data type to hold what the detector is used for - the measurement of the observable, the monitor or standard, or the flux directly.

Specifications for: use

type: text**properties:****attributes:** [object] Attributes of text container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required, enumerated] What the detector was used for.

JSON example of use

```

"use": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "observable"
}

```

2.6.3 Detectors container: manufacturer

A data type to hold the manufacturer of the detector. If the detector was not created commercially, use in-house. The example is taken from EXFOR entry 14576.

Specifications for: manufacturer

type: text**properties:****attributes:** [object] Attributes of text container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required] The manufacturer.

JSON example of manufacturer

```

"manufacturer": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "Two custom made scintillation detectors
based on EJ-301 liquid scintillator with dimensions of 17.8cm x 35.1cm x
12.7cm were used for neutron detection. Each module was read out using
two Photonis XP4572B 12.7-cm-diameter photomultiplier tubes. Two
detectors covered neutron beam area of 35.6 x 35.1 cm2."
  },
  "body": "in-house"
}

```

2.6.4 Detectors container: pmt

A data type to hold the manufacturer of the Photomultiplier Tube (PMT). If the detector was not created commercially, use in-house. The example is taken from EXFOR entry 14576.

Specifications for: pmt

type: commercial product

properties:

attributes: [object] Attributes of commercial product container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

content: [object, required] Content of the commercial product container. Properties:

manufacturer: [string, required] The name of the manufacturer, or in house.

model number: [string] The model number of the product

use: [string] What the product is used for

description: [string] Any free-text description available for the information in the container.

JSON example of pmt

```

"pmt": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "manufacturer": "Photonis",
  "model_number": "XP4572B",
  "use": "observable",
  "description": "Two custom made scintillation detectors based
on EJ-301 liquid scintillator with dimensions of 17.8cm x 35.1cm x 12.7cm
were used for neutron detection. Each module was read out using two
Photonis XP4572B 12.7-cm-diameter photomultiplier tubes. Two detectors
covered neutron beam area of 35.6 x 35.1 cm2."
}

```

2.6.5 Detectors container: daq system

A data type to hold details about a single Digital Acquisition (DAQ) system used in the measurement. The example is taken from the main reference for EXFOR entry 14576.

Specifications for: daq system

type: commercial product

properties:

attributes: [object] Attributes of commercial product container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

content: [object, required] Content of the commercial product container. Properties:

manufacturer: [string, required] The name of the manufacturer, or in house.

model number: [string] The model number of the product

use: [string] What the product is used for

description: [string] Any free-text description available for the information in the container.

JSON example of daq system

```

"daq_system": {
  "attributes": {
    "source": ["publication"]
  },
  "manufacturer": "ORTEC",
  "model_number": "ORTEC935",
  "use": "constant fraction discriminator",
  "description": "The analog signals from the photomultiplier tubes
were conditioned by collecting the output pulses from the detector and
converting them to logic pulses using an ORTEC 935 constant fraction
discriminator, which allowed for optimizing time resolution over the wide
range of pulse amplitudes."
}

```

2.6.6 Detectors container: number detectors

A data type to hold the number of detectors or segments if the detector system is an array of detectors or is a large detector with multiple segments. This key should not be used if the detector “system” is a single detector. The examples are taken from EXFOR entries 14041 and 14682.

Specifications for: number detectors

type: number

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [number, required] The number of detectors.

JSON example of number detectors

```

"number_detectors": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "GEANIE Ge array, 26 Ge detectors"
  },
  "body": 26
}

```

JSON example of number detectors

```

"number_detectors": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "Fission neutrons with energy below
1.59MeV were detected using array of 22 Li-glass scintillation detectors.
21 detectors used GS20 Li-glass enriched with 95% of 6Li, one detector
used GS30 Li-glass enriched with 99.9% of 7Li. Five of these detectors
contained 1.27 cm thick Li-glass, the others were 1.80 cm thick. Each
Li-glass was mounted to an R1250A Hamamatsu photomultiplier tube. The
nominal flight path from the PPAC to a Li- glass detector was
(0.400+-0.005) m. Time resolution was (1.05+-1) ns."
  },
  "body": 22
}

```

2.6.7 Detectors container: geometry

A data type to hold the geometry of the detector. The example is taken from EXFOR entry 14721.

Specifications for: geometry

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required, enumerated] The geometry.

JSON example of geometry

```

"geometry": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "Time projection chamber (TPC) was used to
measure fission fragments tracks. TPC consists of two cylindrical gas
filled detector volumes viewing a central target. The two volumes are
filled with a common drift gas, a mixture of argon with 5% isobutane at a
pressure of 550 torr."
  },
  "body": "cylinder"
}

```

2.6.8 Detectors container: volume

A data type to hold the volume of the detector, if applicable. The example is taken from EXFOR entry 14576.

Specifications for: volume

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of volume

```

"volume": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "cm3",
    "value_type": "mean"
  },
  "value": [7934.0],
  "description": "Two custom made scintillation detectors
based on EJ-301 liquid scintillator with dimensions of 17.8cm x 35.1cm x
12.7cm were used for neutron detection. Each module was read out using
two Photonis XP4572B 12.7-cm-diameter photomultiplier tubes. Two
detectors covered neutron beam area of 35.6 x 35.1 cm2."
}

```

2.6.9 Detectors container: radius

A data type to hold the radius of the detector, if given.

Specifications for: radius

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of radius

```

"radius": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "m",
    "value_type": "mean"
  },
  "value": [0.05]
}

```

2.6.10 Detectors container: throughtube diameter

A data type to hold the diameter of the throughtube for nubar measurements, if given. The example is taken from the main reference for EXFOR entry 10759.

Specifications for: throughtube diameter

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of throughtube diameter

```

"throughtube_diameter": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "m",
    "value_type": "mean"
  },
  "value": [0.178],
  "description": "The liquid scintillator used at 21.6 m had a
volume of about 710 l and the diameter of the through tube was
about 17.8 cm."
}

```

2.6.11 Detectors container: thickness

A data type to hold the thickness of the detector along the axis of the beam. The example is taken from EXFOR entry 14682.

Specifications for: thickness

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of thickness

```



```

2.6.12 Detectors container: particle detected

A data type to hold the type of particle detected by this detector.

Specifications for: particle detected

type: isotope

properties:

attributes: [object] Attributes of isotope container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required] The isotope in ZZZ-XX-AAA-LL format.

JSON example of particle detected

```

"particle_detected": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "001-H-001"
}

```

2.6.13 Detectors container: efficiency method

A data type to hold the method or methods used to determine the efficiency of the detection system. The example is taken from EXFOR entry 32713.

Specifications for: efficiency method
<p>type: text array</p> <p>properties:</p> <ul style="list-style-type: none"> attributes: [object] Attributes of the text array container. Properties: <ul style="list-style-type: none"> source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name. description: [string] Any free-text description available for the information in the container. body: [array, required, enumerated] List of efficiency methods.

JSON example of efficiency method
<pre> "efficiency_method": { "attributes": { "source": ["EXFOR"], "description": "The crystal geometry is 70.1 x 72.3 mm in diameter and length. The relative efficiency is about 68%, and energy resolution is 1.69 keV at 1332 keV gamma ray from Co-60. The efficiency of the detector was pre-calibrated using various standard gamma sources, which were Mn-54, Co-57, Co-60, Cd-109, Ba-133, Cs-137, Eu-152, Am-241 and Ra-226." }, "body": ["calibration_sources"] } </pre>

2.6.14 Detectors container: energy calibration method

A data type to hold the method or methods used to determine the energy calibration of the detection system. The example is taken from EXFOR entry 32713.

Specifications for: energy calibration method

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required, enumerated] List of calibration methods.

JSON example of energy calibration method

```
"energy_calibration_method": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "The crystal geometry is 70.1 x 72.3 mm in
diameter and length. The relative efficiency is about 68%, and energy
resolution is 1.69 keV at 1332 keV gamma ray from Co-60. The efficiency
of the detector was pre-calibrated using various standard gamma sources,
which were Mn-54, Co-57, Co-60, Cd-109, Ba-133, Cs-137, Eu-152, Am-241
and Ra-226."
  },
  "body": ["calibration_sources"]
}
```

2.6.15 Detectors container: calibration sources

A data type to hold the calibration sources used for energy calibration of the detection system. The example is taken from EXFOR entry 32713.

Specifications for: calibration sources

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required] List of calibration sources.

JSON example of calibration sources

```

"calibration_sources": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "The crystal geometry is 70.1 x 72.3 mm in
diameter and length. The relative efficiency is about 68%, and energy
resolution is 1.69 keV at 1332 keV gamma ray from Co-60. The efficiency
of the detector was pre-calibrated using various standard gamma sources,
which were Mn-54, Co-57, Co-60, Cd-109, Ba-133, Cs-137, Eu-152, Am-241
and Ra-226."
  },
  "body": ["025-Mn-54", "027-Co-57", "027-Co-60", "048-Cd-109", ".."]
}

```

2.6.16 Detectors container: detector shielding

A data type to hold information about the shielding used in the detector, as a list of the shielding materials. The example is taken from EXFOR entry 40143.

Specifications for: detector shielding

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required] List of detector shielding materials.

JSON example of detector shielding

```

"detector_shielding": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "Scintillation detector on eight NaI(Tl)
crystals and B-10 samples in case of neutrons. For capture and
self-indication - detector of two NaI(Tl) crystals with lithium shielding"
  },
  "body": ["lithium"]
}

```

2.6.17 Detectors container: solid angle

A data type to hold the solid angle coverage of the detector. The example is taken from EXFOR entry C1345.

Specifications for: solid angle

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of solid angle

```
"solid_angle": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "msr",
    "value_type": "mean"
  },
  "value": [2.886],
  "description": "Two detectors were used to allow simultaneous collection of Elastic Recoil Spectrometry (ERS) and Rutherford Backscattering Spectra (RBS). The RBS detector was at 170 degree with a solid angle of 2.88 msr at the sample. The angle of the ERS detector was adjustable, and its active surface was 11.8 mm x 2.1 mm; its solid angle was 0.846 msr."
}
```

2.6.18 Detectors container: deadtime correction

A data type to hold information about a correction for dead time in the detector. The example is taken from EXFOR entry 10227.

Specifications for: deadtime correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of `deadtime correction`

```

"deadtime_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "magnitude": {
    "attributes": {
      "value_type": "mean",
      "unit": "percent"
    },
    "value": "0.15"
  },
  "description": "Corrected for in-scattering (0.31%), and for
deadtime losses (0.15%)."
}

```

2.6.19 Detectors container: `live time`

A data type to hold the time that the detector was live. The example is taken from EXFOR entry 14254.

Specifications for: `live time`

type: measured quantity

properties:

- attributes:** [object, required] Attributes of measured quantity container. Properties:
 - source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.
 - unit:** [string, required, enumerated] The unit.
 - value type:** [string, required, enumerated] The type of the value.
- content:** [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:
 - value:** [object, required] A data type to hold a measured value, which has a unit.
 - uncertainty:** [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.
 - description:** [string] Any free-text description available for the information in the container.

JSON example of live time

```
"live_time": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "uncertainty": {
    "attributes": {
      "unit": "percent",
      "unc_type": "relative"
    },
    "body": [1.0]
  },
  "description": "(ERR-2,,1.) Systematic uncertainty in live time
fraction"
}
```

2.6.20 Detectors container: pulse pileup correction

A data type to hold information about a correction for pulse pileup. The example is taken from EXFOR entry 30751.

Specifications for: pulse pileup correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of pulse pileup correction

```
"pulse_pileup_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "yes"
  },
  "magnitude": {
    "attributes": {
    },
    "uncertainty": {
      "attributes": {
        "unc_type": "relative",
        "unit": "percent"
      },
      "body": [0.7, 0.6, 0.7]
    }
  },
  "description": "(ERR-3) Error of pulse pile-up correction"
}
```

2.6.21 Detectors container: discriminator correction

A data type to hold information about a correction for discriminators used in the data analysis. The example is taken from EXFOR entry D0185.

Specifications for: discriminator correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of discriminator correction

```
"discriminator_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "negligible",
    "energy_dependent": "no"
  },
  "description": "reaction producing charged-particle which
entered the telescope with the energy above the discriminator level
(negligible)"
}
```

2.6.22 Detectors container: electronic noise correction

A data type to hold information about a correction for electronic noise, if done separately from the other corrections. The example is taken from EXFOR entry 20403.

Specifications for: electronic noise correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of electronic noise correction

```
"electronic_noise_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "description": "random counts due to electronic noise and
background neutrons and gammas, were considered to be corrected."
}
```

2.6.23 Detectors container: forward boost correction

A data type to hold information about a correction for forward boost. If fission cross sections or neutron multiplicities are measured above 10 MeV, the measured observable needs to be corrected for the forward-boost of the fission fragments that leads to a different angular distribution of fission fragments in the fission chamber. This preferential direction of fission fragments can lead to a decrease of the detector efficiency

due to the finite angular coverage of detectors. The example is taken from the main reference for EXFOR entry 14721.

Specifications for: forward boost correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of forward boost correction

```
"forward_boost_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "no"
  },
  "description": "Kinematic boost and fission anisotropy; this
will affect the relative fraction of fragments leaving the target at any
given angle,"
}
```

2.6.24 Detectors container: angular distortion correction

A data type to hold information about a correction for angular distortion. This correction applies to fission cross section and neutron multiplicity measurements above 0.2 MeV. Fission fragments have an inherent angular distribution that is non-isotropic above this energy dependent on the isotope at hand. Given this preferential direction of fission fragments and the fact that fission detectors are not covering all angles, more or fewer fission fragments might be detected due to their angular distribution as a function of the particular incident neutron energy.

Specifications for: angular distortion correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of angular distortion correction

```
"angular_distortion_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "no"
  }
}
```

2.6.25 Detectors container: delayed gamma correction

A data type to hold information about a correction for delayed gammas. This correction applies to coincidence measurements of the neutron multiplicities where a coincidence between fission detection and neutron measurements is taken to derive the neutron multiplicity. Neutrons are often measured via scintillators that transform neutrons to photons and then detect those photons. If delayed gammas arise from the decay of isomeric states, these can be also measured with the scintillator and need to be corrected for. The example is taken from EXFOR entry 12833.

Specifications for: delayed gamma correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of delayed gamma correction

```
"delayed_gamma_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "description": "Corrections were done for
- displacement of samples from centre of detector,
- false fissions,
- deadtime and background of detection system,
- neutron detection efficiency,
- impurity fission,
- delayed gamma rays from fission to detector system."
}
```

2.6.26 Detectors container: false fission correction

A data type to hold information about a correction for false fissions. This correction applies to coincidence measurements of the fission neutron multiplicities where a coincidence between fission detection and neutron measurements is taken to derive the neutron multiplicity. They are related to a random coincidences between background counts (e.g., alpha activity) that trigger the fission detector to open the counting gate and background neutrons/ gammas detected in the liquid scintillator. The example is taken from EXFOR entry 12833.

Specifications for: false fission correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of false fission correction

```
"false_fission_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "description": "Corrections were done for
- displacement of samples from centre of detector,
- false fissions,
- deadtime and background of detection system,
- neutron detection efficiency,
- impurity fission,
- delayed gamma rays from fission to detector system."
}
```

2.6.27 Detectors container: french effect correction

A data type to hold information about a correction for the “French effect”. This correction only applies to neutron multiplicities measured absolutely with liquid scintillators. It originates from a triple coincidence that opens the neutron gate (fission fragments detected by the fission chamber, prompt gammas and proton recoils detected by the scintillator). The example is taken from EXFOR entry 31761.

Specifications for: french effect correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of french effect correction

```
"french_effect_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "no"
  },
  "magnitude": {
    "attributes": {
      "source": ["EXFOR"],
      "uncertainty": {
        "attributes": {
          "unc_type": "relative",
          "unit": "percent"
        },
        "body": [0.1]
      }
    }
  },
  "description": "(ERR-6) French effect - 0.10%"
}
```

2.6.28 Detectors container: hs content correction

A data type to hold information about a correction for hydrogen and sulfur content. This correction only applies to Mn-bath measurement, where one needs to correct for the ratio of thermalized hydrogen and sulfur to Mn. The example is taken from EXFOR entry 41595.

Specifications for: hs content correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of hs content correction

```
"hs_content_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "no"
  },
  "magnitude": {
    "attributes": {
      "source": ["EXFOR"]
    },
    "uncertainty": {
      "attributes": {
        "unit": "percent",
        "unc_type": "relative"
      },
      "body": [0.1]
    }
  },
  "description": "(ERR-12) Mn bath chemical purity."
}
```

2.6.29 Detectors container: manganese capture correction

A data type to hold information about a correction for manganese capture. This correction only applies to Mn-bath measurement, where one accounts for uncertainties in the resonance-capture term of the neutrons in the Mn in the bath. The example is taken from EXFOR entry 41595.

Specifications for: manganese capture correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of manganese capture correction

```
"manganese_capture_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "no"
  },
  "magnitude": {
    "attributes": {
      "source": ["EXFOR"]
    },
    "uncertainty": {
      "attributes": {
        "unit": "percent",
        "unc_type": "relative"
      },
      "body": [0.1]
    }
  },
  "description": "(ERR-11) Mn resonance integral."
}
```

2.6.30 Detectors container: false capture correction

A data type to hold information about a correction for “false capture” in a detector signal. This is used in capture yield measurements and represents the signal from neutrons captured in materials that are not the sample. The example is taken from EXFOR entry 14709.

Specifications for: false capture correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of false capture correction

```
"false_capture_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "description": "Corrections for dead time (<2.5%) and false
capture were made"
}
```

2.6.31 Detectors container: spectrum unfolding correction

A data type to hold information about a correction for the unfolding of a spectrum measured in the detector. The example is taken from EXFOR entry 20831.

Specifications for: spectrum unfolding correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of spectrum unfolding correction

```
"spectrum_unfolding_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "description": "(DATA-ERR) The dominating uncertainty is from
gamma spectrum unfolding, detector efficiency, response line shape, and
statistical uncertainties, typically 25%. Total neutron energy spread
less than 200 keV, except at 14 MeV where it is 400 keV."
}
```

2.6.32 Detectors container: sum peaks correction

A data type to hold information about a correction for sum peaks, which are detector signals that represent the detection of multiple particles. The example is taken from EXFOR entry 32615.

Specifications for: sum peaks correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of sum peaks correction

```
"sum_peaks_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "magnitude": {
    "attributes": {
      "source": ["EXFOR"],
      "uncertainty": {
        "attributes": {
          "unit": "percent",
          "unc_type": "relative"
        },
        "body": [3]
      }
    }
  },
  "description": "Corrections were made for:
- fluctuation of neutron flux
- neutron scattering background
- gamma-ray self-absorption in sample and sum peaks for Eu-152m
(ERR-5) correction of sum peaks (3%)"
}
```

2.6.33 Detectors container: gamma angular distribution correction

A data type to hold information about a correction for non-isotropic distribution of gammas emitted by the sample. This key should only be used for experiments that detect gammas. The example is taken from EXFOR entry R0021.

Specifications for: gamma angular distribution correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of gamma angular distribution correction

```
"gamma_angular_distribution_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "negligible",
    "energy_dependent": "no"
  },
  "description": "It was assumed that the angular distribution of
background continuum lying under the broad peak of interest was similar
to that observed near 50 MeV excitation energy."
}
```

2.6.34 Detectors container: counting geometry correction

A data type to hold information about a correction for the geometry of the counting for activation measurements. The example is taken from EXFOR entry 12898.

Specifications for: counting geometry correction**type:** correction**properties:****attributes:** [object, required] Attributes of correction container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**applied:** [string, required, enumerated] Whether or not the correction was applied.**energy dependent:** [string, required, enumerated] Whether or not the correction is energy-dependent.**content:** [object, required] The contents of the correction container. Properties:**methods:** [array, enumerated] The methods used to perform the correction.**components:** [array, enumerated] The components of the experimental setup that were included in the correction.**code:** [array, enumerated] The primary code(s) used to perform or calculate the correction.**reference:** [object] A data type to hold a single reference.**magnitude:** [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.**description:** [string] Any free-text description available for the information in the container.

JSON example of counting geometry correction

```

"counting_geometry_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "description": "CORRECTIONS WERE MADE FOR-
. DETECTOR EFFICIENCY
. DEAD-TIME
. BACKGROUND EFFECTS
. FISSION EVENTS OF LOW-ENERGY
. FISSION PRODUCED BY BACKGROUND NEUTRONS
. NEUTRON MULTIPLE SCATTERING
. GEOMETRY FACTORS
. NEUTRON SOURCE PROPERTIES
. NEUTRON ABSORPTION
. SECONDARY NEUTRONS FROM LI(P,N) AND D(D,N) SOURCES"
}

```

2.6.35 Detectors container: coincidence measurement

A data type to hold whether or not the measurement was done in coincidence. The example is taken from EXFOR entry 23724.

Specifications for: coincidence measurement

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required, enumerated] Whether the measurement was done in coincidence.

JSON example of coincidence measurement

```

"coincidence_measurement": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "(COINC) the time gate applied to the data
= 6 ns."
  },
  "body": "yes"
}

```

2.6.36 Detectors container: time gate open

A data type to hold the time gate used in a coincident measurement. This key should only be used for coincidence measurements. The example is taken from EXFOR entry 23724.

Specifications for: time gate open

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of time gate open

```

"time_gate_open": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "s",
    "value_type": "mean"
  },
  "value": [6e-09],
  "description": "(COINC) the time gate applied to the data
= 6 ns."
}

```

2.6.37 Detectors container: gas

A data type to hold the type of gas used in the detector. The example is taken from EXFOR entry 14463.

Specifications for: gas

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required] The type of gas in the detector.

JSON example of gas

```

"gas": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "A twin Frisch-gridded ionization chamber
was chosen for fission detection, mainly due to its high efficiency, good
energy resolution (0.5 - 1.0%) and nearly 4PI solid angle coverage. The
ionization chamber was operated with P10 fill gas (90% argon and 10%
methane) at 1.25 kPa."
  },
  "body": "P10"
}

```

2.6.38 Detectors container: gas pressure

A data type to hold the pressure of the gas used in the detector. The example is taken from EXFOR entry 14463.

Specifications for: gas pressure

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of gas pressure

```
"gas_pressure": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "Pa",
    "value_type": "mean"
  },
  "value": [1250.0],
  "description": "A twin Frisch-gridded ionization chamber
was chosen for fission detection, mainly due to its high efficiency, good
energy resolution (0.5 - 1.0%) and nearly 4PI solid angle coverage. The
ionization chamber was operated with P10 fill gas (90% argon and 10%
methane) at 1.25 kPa."
}
```

2.6.39 Detectors container: reaction

A data type to hold the reaction or reactions within the detector used to detect the particles.

Specifications for: reaction

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required] The reaction.

JSON example of reaction

```
"reaction": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "17-C1-037(n,p)"
}
```

2.6.40 Detectors container: stopping power correction

A data type to hold information about a correction for the stopping power in charged particle detectors and fission chambers. The example is taken from EXFOR entry O2306.

Specifications for: stopping power correction**type:** correction**properties:****attributes:** [object, required] Attributes of correction container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**applied:** [string, required, enumerated] Whether or not the correction was applied.**energy dependent:** [string, required, enumerated] Whether or not the correction is energy-dependent.**content:** [object, required] The contents of the correction container. Properties:**methods:** [array, enumerated] The methods used to perform the correction.**components:** [array, enumerated] The components of the experimental setup that were included in the correction.**code:** [array, enumerated] The primary code(s) used to perform or calculate the correction.**reference:** [object] A data type to hold a single reference.**magnitude:** [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.**description:** [string] Any free-text description available for the information in the container.

JSON example of stopping power correction

```
"stopping_power_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "no",
    "energy_dependent": "unknown"
  },
  "magnitude": {
    "attributes": {
      "source": ["EXFOR"]
    },
    "uncertainty": {
      "attributes": {
        "unit": "percent",
        "unc_type": "relative"
      },
      "body": [4]
    }
  },
  "description": "(ERR-4) Error in stopping power"
}
```

2.6.41 Detectors container: pulse height extrapolation correction

A data type to hold information about a correction for the extrapolation of pulse height below the discriminator. The example is taken from EXFOR entry 10237.

Specifications for: pulse height extrapolation correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of pulse height extrapolation correction

```
"pulse_height_extrapolation_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "description": "Corrected for second 7Li(p,n) group, other
uranium isotopes, inelastic scattering in sample backings, pulse-height
extrapolation, and background. Effects due to elastic scattering in
sample supports, sample geometry, and momentum effects were eliminated by
the normalization or by measurements in both counter orientations."
}
```

2.6.42 Detectors container: efficiency

A data type to hold the efficiency values, if given, as a function of particle energy, detection angle, mass, etc. For measurements that are done in ratio to a monitor reaction

where the same detector system is used in both, there is usually no efficiency given. (But an efficiency correction should be given.) The example is taken from EXFOR entry 32713.

Specifications for: efficiency

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of efficiency

```
"efficiency": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "percent",
    "value_type": "mean",
    "description": "The crystal geometry is 70.1 x 72.3 mm in
diameter and length. The relative efficiency is about 68%, and energy
resolution is 1.69 keV at 1332 keV gamma ray from Co-60. The efficiency
of the detector was pre-calibrated using various standard gamma sources,
which were Mn-54, Co-57, Co-60, Cd-109, Ba-133, Cs-137, Eu-152, Am-241
and Ra-226."
  },
  "value": [68]
}
```

2.6.43 Detectors container: energy resolution

A data type to hold the value(s) of the energy resolution of the detector. The example is taken from EXFOR entry 32713.

Specifications for: energy resolution

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of energy resolution

```
"energy_resolution": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "eV",
    "value_type": "mean",
    "description": "The crystal geometry is 70.1 x 72.3 mm in
diameter and length. The relative efficiency is about 68%, and energy
resolution is 1.69 keV at 1332 keV gamma ray from Co-60. The efficiency
of the detector was pre-calibrated using various standard gamma sources,
which were Mn-54, Co-57, Co-60, Cd-109, Ba-133, Cs-137, Eu-152, Am-241
and Ra-226."
  },
  "value": [1690.0]
}
```

2.6.44 Detectors container: time resolution

A data type to hold the value(s) of the time resolution of the detector. The example is taken from EXFOR entry 14682.

Specifications for: time resolution

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of time resolution

```
"time_resolution": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "s",
    "value_type": "mean"
  },
  "value": [1.05e-09],
  "uncertainty": {
    "attributes": {
      "source": ["EXFOR"],
      "unc_type": "absolute",
      "unit": "s"
    },
    "body": [1e-09]
  },
  "description": "Fission neutrons with energy below
1.59MeV were detected using array of 22 Li-glass scintillation detectors.
21 detectors used GS20 Li-glass enriched with 95% of 6Li, one detector
used GS30 Li-glass enriched with 99.9% of 7Li. Five of these detectors
contained 1.27 cm thick Li-glass, the others were 1.80 cm thick. Each
Li-glass was mounted to an R1250A Hamamatsu photomultiplier tube. The
nominal flight path from the PPAC to a Li- glass detector was
(0.400+-0.005) m. Time resolution was (1.05+-1) ns."
}
```

2.7 Sample

The `samples` container is a list of individual `sample container`, each of which describe a single sample used (such as the target(s), and any number of normalization or reference samples).

```
"samples": [
    {sample container},
    {sample container}
]
```

Section Contents:

2.7.1	Samples container: use	146
2.7.2	Samples container: incident energy	146
2.7.3	Samples container: incident angle	147
2.7.4	Samples container: configuration	148
2.7.5	Samples container: composition methods	149
2.7.6	Samples container: impurity correction	150
2.7.7	Samples container: type	152
2.7.8	Samples container: geometry	153
2.7.9	Samples container: production methods	154
2.7.10	Samples container: thickness	155
2.7.11	Samples container: mass	156
2.7.12	Samples container: radius	157
2.7.13	Samples container: areal density	158
2.7.14	Samples container: mass density	159
2.7.15	Samples container: number density	160
2.7.16	Samples container: height	161
2.7.17	Samples container: width	162
2.7.18	Samples container: density methods	163
2.7.19	Samples container: particulate radius	164
2.7.20	Samples container: thermodynamic temperature	165
2.7.21	Samples container: effective temperature	166
2.7.22	Samples container: irradiation time	167
2.7.23	Samples container: cooling time	168
2.7.24	Samples container: counting time	169
2.7.25	Samples container: uniformity correction	170
2.7.26	Samples container: roughness correction	171
2.7.27	Samples container: displacement correction	172
2.7.28	Samples container: multiple scattering correction	174
2.7.29	Samples container: neutron absorption correction	175
2.7.30	Samples container: beam overlap correction	176
2.7.31	Samples container: gamma attenuation correction	178
2.7.32	Samples container: resonance self shielding correction	180
2.7.33	Samples container: multiple sample effects correction	182
2.7.34	Samples container: radioactivity correction	183

2.7.1 Samples container: use

A data type to hold what the sample is used for - the measurement of the observable, the monitor or standard, or the flux directly.

Specifications for: use

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required, enumerated] What the sample was used for.

JSON example of use

```
"use": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "observable"
}
```

2.7.2 Samples container: incident energy

A data type to hold the incident energy seen by the sample, if different samples are irradiated with different incident energies. This can occur in measurements with multiple samples with charged particle beams which degrade in energy, or neutron generators that have different energy spectra at different angles. If the sample sees the entire energy spectrum described in the `incident_particle` container, this key should not be used. The example is taken from EXFOR entry 14556.

Specifications for: incident energy

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, Or year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of incident energy

```
"incident_energy": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "eV",
    "value_type": "range"
  },
  "value": [2.42, 2.74],
  "description": "Samples of approx. 250 mg reagent grade
NaCl were pressed into pellets using a hydraulic press, with the resulting
pellets having a thickness of 1 mm and diameter of 11 mm. Behind each
sample a reference foil, either In or 58Ni was placed. Two experiments
used single pellet at 0 deg with In monitor, third experiment used pellets
at four different positions to cover incident energy range 2.42 - 2.74
MeV and 58Ni monitor."
}
```

2.7.3 Samples container: incident angle

A data type to hold the value of the incident particle angle, if the samples are irradiated at different angles. The example is taken from EXFOR entry 14556.

Specifications for: incident angle

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, Or year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of incident angle

```
"incident_angle": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "degree",
    "value_type": "mean"
  },
  "value": [0],
  "description": "Samples of approx. 250 mg reagent grade
NaCl were pressed into pellets using a hydraulic press, with the resulting
pellets having a thickness of 1 mm and diameter of 11 mm. Behind each
sample a reference foil, either In or 58Ni was placed. Two experiments
used single pellet at 0 deg with In monitor, third experiment used pellets
at four different positions to cover incident energy range 2.42 - 2.74
MeV and 58Ni monitor."
}
```

2.7.4 Samples container: configuration

A data type to hold the configuration of the samples. If there is only a single sample, use `single`. The example is taken from EXFOR entry 14556.

Specifications for: configuration

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required, enumerated] List of configuration values.

JSON example of configuration

```
"configuration": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "Samples of approx. 250 mg reagent grade
NaCl were pressed into pellets using a hydraulic press, with the resulting
pellets having a thickness of 1 mm and diameter of 11 mm. Behind each
sample a reference foil, either In or 58Ni was placed. Two experiments
used single pellet at 0 deg with In monitor, third experiment used pellets
at four different positions to cover incident energy range 2.42 - 2.74
MeV and 58Ni monitor."
  },
  "body": ["back_to_back"]
}
```

2.7.5 Samples container: composition methods

A data type to hold the methods used to determine the composition of the sample. The example is taken from EXFOR entry A0480.

Specifications for: composition methods

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required, enumerated] List of composition methods.

JSON example of composition methods

```
"composition_methods": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "The U-233 (isotopic purity of approximately
96%, U-238 - 3%, U-234 <1%). Most of the targets were prepared by
electrodeposition of 0.1 to 2 mg of hydrated uranium oxide over an area
of about 1 cm**2 on a dish-shaped aluminum disk. The amount of deposited
material was determined by direct alpha counting, weighting, or both.
These targets were then mounted in a water-cooled micro-target holder
which also served as a Faraday cup for beam intensity measurements."
  },
  "body": ["alpha_counting", "weighting"]
}
```

2.7.6 Samples container: impurity correction

A data type to hold information about a correction for impurities in the sample. The example is taken from EXFOR entry 22796.

Specifications for: impurity correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of impurity correction

```

"impurity_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "magnitude": {
    "attributes": {
      "source": ["EXFOR"],
      "uncertainty": {
        "attributes": {
          "unit": "percent",
          "unc_type": "relative"
        },
        "body": [1.5]
      }
    }
  },
  "description": "(ERR-4) Isotopic impurity correction (1.5%)"
}

```

2.7.7 Samples container: type

A data type to hold the type of sample. The example is taken from EXFOR entry 14556.

Specifications for: type

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required, enumerated] The sample type.

JSON example of type

```

"type": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "Samples of approx. 250 mg reagent grade
NaCl were pressed into pellets using a hydraulic press, with the resulting
pellets having a thickness of 1 mm and diameter of 11 mm. Behind each
sample a reference foil, either In or 58Ni was placed. Two experiments
used single pellet at 0 deg with In monitor, third experiment used pellets
at four different positions to cover incident energy range 2.42 - 2.74
MeV and 58Ni monitor."
  },
  "body": "powder"
}

```

2.7.8 Samples container: geometry

A data type to hold the geometry of the sample. The example is taken from EXFOR entry A0480.

Specifications for: geometry

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required, enumerated] The sample geometry.

JSON example of geometry

```

"geometry": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "The U-233 (isotopic purity of approximately
96%, U-238 - 3%, U-234 <1%). Most of the targets were prepared by
electrodeposition of 0.1 to 2 mg of hydrated uranium oxide over an area
of about 1 cm**2 on a dish-shaped aluminum disk. The amount of deposited
material was determined by direct alpha counting, weighting, or both.
These targets were then mounted in a water-cooled micro-target holder
which also served as a Faraday cup for beam intensity measurements."
  },
  "body": "disk"
}

```

2.7.9 Samples container: production methods

A data type to hold the method or methods used to create the sample. The example is taken from EXFOR entry A0480.

Specifications for: production methods

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required, enumerated] The sample production methods.

JSON example of production methods

```

"production_methods": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "The U-233 (isotopic purity of approximately
96%, U-238 - 3%, U-234 <1%). Most of the targets were prepared by
electrodeposition of 0.1 to 2 mg of hydrated uranium oxide over an area
of about 1 cm*2 on a dish-shaped aluminum disk. The amount of deposited
material was determined by direct alpha counting, weighting, or both.
These targets were then mounted in a water-cooled micro-target holder
which also served as a Faraday cup for beam intensity measurements."
  },
  "body": ["electro_deposited"]
}

```

2.7.10 Samples container: thickness

A data type to hold the thickness of the sample in the direction of the beam. The example is taken from EXFOR entry 14556.

Specifications for: thickness

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of thickness

```



```

2.7.11 Samples container: mass

A data type to hold the mass of the sample. The example is taken from EXFOR entry 14556.

Specifications for: mass

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of mass

```

"mass": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "g",
    "value_type": "approximate"
  },
  "value": [0.25],
  "description": "Samples of approx. 250 mg reagent grade
NaCl were pressed into pellets using a hydraulic press, with the resulting
pellets having a thickness of 1 mm and diameter of 11 mm. Behind each
sample a reference foil, either In or 58Ni was placed. Two experiments
used single pellet at 0 deg with In monitor, third experiment used pellets
at four different positions to cover incident energy range 2.42 - 2.74
MeV and 58Ni monitor."
}

```

2.7.12 Samples container: radius

A data type to hold the radius of the sample. The example is taken from EXFOR entry 14556.

Specifications for: radius

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of radius

```

"radius": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "m",
    "value_type": "mean"
  },
  "value": [0.011],
  "description": "Samples of approx. 250 mg reagent grade
NaCl were pressed into pellets using a hydraulic press, with the resulting
pellets having a thickness of 1 mm and diameter of 11 mm. Behind each
sample a reference foil, either In or 58Ni was placed. Two experiments
used single pellet at 0 deg with In monitor, third experiment used pellets
at four different positions to cover incident energy range 2.42 - 2.74
MeV and 58Ni monitor."
}

```

2.7.13 Samples container: areal density

A data type to hold the areal density of the sample. The example is taken from EXFOR entry 13901.

Specifications for: areal density

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of areal density

```

"areal_density": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "g/m^2",
    "value_type": "mean"
  },
  "value": [7300.0],
  "description": "Two sets of runs made with different samples;
all samples enriched to 99.798% 238U:
1) 1998: 238U foil, 730 mg/cm2 thick.
2) 1999: two 238U foils, 840 mg/cm2 thick in total, along with 4 natural
Fe foils, 2 in front and 2 in back of the 238U foils."
}

```

2.7.14 Samples container: mass density

A data type to hold the mass density of the sample.

Specifications for: mass density

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of mass density

```

"mass_density": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "g/m^3",
    "value_type": "mean"
  },
  "value": [0.03998],
  "uncertainty": {
    "attributes": {
      "source": ["EXFOR"],
      "unit": "g/m^3",
      "unc_type": "absolute"
    },
    "body": [1e-05]
  }
}

```

2.7.15 Samples container: number density

A data type to hold the number density of the sample in the direction of the beam.

Specifications for: number density

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of number density

```

"number_density": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "1/m^3",
    "value_type": "mean"
  },
  "value": [0.03998],
  "uncertainty": {
    "attributes": {
      "source": ["EXFOR"],
      "unit": "1/m^3",
      "unc_type": "absolute"
    },
    "body": [1e-05]
  }
}

```

2.7.16 Samples container: height

A data type to hold the height of the sample. The example is taken from EXFOR entry 22794.

Specifications for: height

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of height

```

"height": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "m",
    "value_type": "maximum"
  },
  "value": [0.066],
  "description": "Stacked target with diameter < 2.5 cm, height
< 6.6 cm"
}

```

2.7.17 Samples container: width

A data type to hold the width of the sample.

Specifications for: width

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of width

```

"width": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "m",
    "value_type": "mean"
  },
  "value": [0.01],
  "uncertainty": {
    "attributes": {
      "source": ["EXFOR"],
      "unit": "percent",
      "unc_type": "relative"
    },
    "body": [1]
  }
}

```

2.7.18 Samples container: density methods

A data type to hold the method or methods used to determine the dimensions, density, and/or number of atoms of the sample. The example is taken from EXFOR entry 14594.

Specifications for: density methods

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required, enumerated] List of density methods.

JSON example of density methods

```

"density_methods": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "Two type of samples: light and heavy
uranium disks. The light sample was prepared by the electrolytic
deposition of uranium on platinum from a solution of UO2(NO3)2*6H2O in
absolute alcohol. The sample was distributed uniformly over a circle 1.3
cm in diameter and had a low specific mass, which was determined by alpha
counting. This sample was used for determination of the number of
fissions by counting the fission recoil fragments in an air ionization
chamber. The heavy samples (disks of uranium metal of the 1.3 cm
diameter) were used for chemically separation of individual fission
products, and the number of atoms present was determined by beta counting."
  },
  "body": ["alpha_counting", "beta_counting"]
}

```

2.7.19 Samples container: particulate radius

A data type to hold the nominal radius of the particulates in the sample.

Specifications for: particulate radius

type: measured quantity

properties:

- attributes:** [object, required] Attributes of measured quantity container. Properties:
 - source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.
 - unit:** [string, required, enumerated] The unit.
 - value type:** [string, required, enumerated] The type of the value.
- content:** [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:
 - value:** [object, required] A data type to hold a measured value, which has a unit.
 - uncertainty:** [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.
 - description:** [string] Any free-text description available for the information in the container.

JSON example of particulate radius

```

"particulate_radius": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "m",
    "value_type": "approximate"
  },
  "value": [5e-09]
}

```

2.7.20 Samples container: thermodynamic temperature

A data type to hold the thermodynamic temperature of the sample. The example is taken from EXFOR entry 14566.

Specifications for: thermodynamic temperature

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of thermodynamic temperature

```
"thermodynamic_temperature": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "K",
    "value_type": "mean"
  },
  "value": [15.6],
  "uncertainty": {
    "attributes": {
      "unit": "K",
      "unc_type": "absolute"
    },
    "body": [0.6]
  },
  "description": "A 16-liter liquid hydrogen target at a
temperature of 15.6+/-0.6 K. The molecular hydrogen has two spin states,
labeled orthohydrogen (J=odd) and parahydrogen (J=even). The lowest
orthohydrogen state (J=1) lies 14.5 meV above the lowest parahydrogen
state (J=0)."
```

2.7.21 Samples container: effective temperature

A data type to hold the effective temperature of the sample. This value is used for resonance measurements where the sample is approximated as a gas at an “effective temperature”. The example is taken from EXFOR entry 10577.

Specifications for: effective temperature

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, Or year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of effective temperature

```
"effective_temperature": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "K",
    "value_type": "mean"
  },
  "value": [301],
  "description": "(92-U-235,ENR=0.002) Metal. Surface dimension (mm x mm, diameter in mm,...) Thickness (atom/barn): 0.00758+-0.00008. Nominal thickness (mm): 1.58+-0.02. Sample thermodynamic temperature was 293+-2 K, and effective temperature was 301K (Thesis table VII-III, page 158)."
```

2.7.22 Samples container: irradiation time

A data type to hold the cumulative time the sample was irradiated. The example was taken from EXFOR entry O1511.

Specifications for: irradiation time

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, Or year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of irradiation time

```
"irradiation_time": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "s",
    "value_type": "range"
  },
  "value": [3600, 5400],
  "description": "The beams currents were nominally either 50 or
100-nA. The irradiation times varied between 1 and 1.5-Hr"
}
```

2.7.23 Samples container: cooling time

A data type to hold the amount of time between irradiation and counting for samples that are counted separately from the beam. The example is taken from EXFOR entry 21624.

Specifications for: cooling time

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of cooling time

```
"cooling_time": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "s",
    "value_type": "range"
  },
  "value": [1800, 8640000.0],
  "description": "Six irradiations from 2 hr to 12 days - at Cf
spectrum; 3 sample irradiations at thermal spectrum; cooling times
between 1/2 hr and 100 days"
}
```

2.7.24 Samples container: counting time

A data type to hold the counting time for samples that are counted separately from the beam. The example is taken from EXFOR entry 22822.

Specifications for: counting time

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of counting time

```
"counting_time": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "uncertainty": {
    "attributes": {
      "unit": "percent",
      "unc_type": "relative"
    },
    "body": [1]
  },
  "description": "(ERR-5,,1.) Uncertainty in counting time error-
1%."
}
```

2.7.25 Samples container: uniformity correction

A data type to hold information about a correction for non-uniformity in the thickness of the sample. The example is taken from EXFOR entry 40547.

Specifications for: uniformity correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of uniformity correction

```
"uniformity_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "description": "For the solid angle
For neutron scattering (flux attenuation)
For nonuniformity of sample
For efficiency of detection"
}
```

2.7.26 Samples container: roughness correction

A data type to hold information about a correction for roughness on the surface of the sample. This is important when charged particles or fission cross sections are measured. The example is taken from EXFOR entry 23102.

Specifications for: roughness correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of roughness correction

```
"roughness_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "no",
    "energy_dependent": "no"
  },
  "description": "Samples of different material and thickness
(0.01-1 mm). Sample holder at 2.220 m distance from the chopper. Surface
roughness of the samples was not quantified."
}
```

2.7.27 Samples container: displacement correction

A data type to hold information about a correction for displacement of the samples between runs. This can also apply to displacement of main and monitor samples with respect to each other. The example is taken from EXFOR entry 12833.

Specifications for: displacement correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of displacement correction

```
"displacement_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "description": "Corrections were done for
- displacement of samples from centre of detector,
- false fissions,
- deadtime and background of detection system,
- neutron detection efficiency,
- impurity fission,
- delayed gamma rays from fission to detector system."
}
```

2.7.28 Samples container: multiple scattering correction

A data type to hold information about a correction for multiple scattering in the sample. The example is taken from EXFOR entry 22796.

Specifications for: multiple scattering correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of multiple scattering correction

```
"multiple_scattering_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "yes"
  },
  "magnitude": {
    "attributes": {
      "source": ["EXFOR"],
      "unit": "percent",
      "value_type": "maximum"
    },
    "value": [5],
    "uncertainty": {
      "attributes": {
        "source": ["EXFOR"],
        "unit": "percent",
        "unc_type": "relative"
      },
      "body": [10]
    }
  },
  "code": "SESH",
  "description": "multiple scattering by the SESH code (<4% for even isotopes and 5% for odd isotopes with uncertainties of ~10%)."
}
```

2.7.29 Samples container: neutron absorption correction

A data type to hold information about a correction for neutron absorption in the sample. The example is taken from EXFOR entry 40979.

Specifications for: neutron absorption correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of neutron absorption correction

```
"neutron_absorption_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "description": "Corrections were made on neutron absorption and
multiple scattering, angular resolution and so on"
}
```

2.7.30 Samples container: beam overlap correction

A data type to hold information about a correction for the physical overlap between the beam cross section and the sample in the irradiation. The example is taken from EXFOR entry 14721.

Specifications for: beam overlap correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of beam overlap correction

```
"beam_overlap_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "yes"
  },
  "magnitude": {
    "attributes": {
      "source": ["EXFOR"],
      "value_type": "mean",
      "unit": "dimensionless"
    },
    "value": [0.72, 0.59, 0.66]
  },
  "description": "Following corrections were applied
- wraparound correction
- contamination correction
- overlap correction
- beam attenuation and scattering correction
(ERR-5) Uncertainty of the overlap correction"
}
```

2.7.31 Samples container: gamma attenuation correction

A data type to hold information about a correction for attenuation of gammas created in the sample. The example is taken from EXFOR entry 10025.

Specifications for: gamma attenuation correction**type:** correction**properties:****attributes:** [object, required] Attributes of correction container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**applied:** [string, required, enumerated] Whether or not the correction was applied.**energy dependent:** [string, required, enumerated] Whether or not the correction is energy-dependent.**content:** [object, required] The contents of the correction container. Properties:**methods:** [array, enumerated] The methods used to perform the correction.**components:** [array, enumerated] The components of the experimental setup that were included in the correction.**code:** [array, enumerated] The primary code(s) used to perform or calculate the correction.**reference:** [object] A data type to hold a single reference.**magnitude:** [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.**description:** [string] Any free-text description available for the information in the container.

JSON example of gamma attenuation correction

```
"gamma_attenuation_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "magnitude": {
    "attributes": {
      "source": ["EXFOR"],
      "value_type": "range",
      "unit": "percent"
    },
    "value": [1, 5]
  },
  "description": "Uncertainty in correction for gamma attenuation
varies from 1% for light elements to 5% for Pu-239."
}
```

2.7.32 Samples container: resonance self shielding correction

A data type to hold information about a correction for resonance self-shielding for the sample. The self-shielding correction is a theoretical correction for the non-linear relationship between the transmission and cross section. In the unresolved resonance region, the energy-averaged transmission is measured and converted into energy-averaged cross section. This correction is required to account for the non-linear relationship. The examples are taken from EXFOR entries 13841 and 14797.

Specifications for: resonance self shielding correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of resonance self shielding correction

```
"resonance_self_shielding_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "no",
    "energy_dependent": "no"
  },
  "description": "Data corrected for:
. small background due to spontaneous alpha decay of 147Sm,
. effects of alpha straggling in samples (14%) using code SRIM. Data in
resonance region have not been corrected for self shielding and multiple
scattering."
}
```

JSON example of resonance self shielding correction

```

"resonance_self_shielding_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "code": "SESH",
  "description": "Correction for multiple scattering and
resonance self-shielding were calculated using the code SESH"
}

```

2.7.33 Samples container: multiple sample effects correction

A data type to hold information about a correction for effects due to having multiple samples.

Specifications for: multiple sample effects correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of multiple sample effects correction

```

"multiple_sample_effects_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "no"
  }
}

```

2.7.34 Samples container: radioactivity correction

A data type to hold information about a correction for natural radioactivity of the sample in the measured signal. The example is taken from EXFOR entry 22940.

Specifications for: radioactivity correction

type: correction

properties:

attributes: [object, required] Attributes of correction container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

applied: [string, required, enumerated] Whether or not the correction was applied.

energy dependent: [string, required, enumerated] Whether or not the correction is energy-dependent.

content: [object, required] The contents of the correction container. Properties:

methods: [array, enumerated] The methods used to perform the correction.

components: [array, enumerated] The components of the experimental setup that were included in the correction.

code: [array, enumerated] The primary code(s) used to perform or calculate the correction.

reference: [object] A data type to hold a single reference.

magnitude: [object] A measured quantity data type to hold the magnitude and uncertainty of the correction applied to the data.

description: [string] Any free-text description available for the information in the container.

JSON example of radioactivity correction

```
"radioactivity_correction": {
  "attributes": {
    "source": ["EXFOR"],
    "applied": "yes",
    "energy_dependent": "unknown"
  },
  "description": "Corrected for
- background due to capture of scattered neutrons by detector materials
- background due to in-beam gamma rays
- samples radioactivity
- ambient background
- dead-time and pile-up"
}
```

2.8 Response Functions

This section will describe response functions, such as the resolution function needed for the modeling of resonance region data.

Section Contents:

2.8.1	Response functions container: type	186
2.8.2	Response functions container: format	186
2.8.3	Response functions container: code	187
2.8.4	Response functions container: code input	187
2.8.5	Response functions container: description	188

2.8.1 Response functions container: type

A data type to hold what type of response function the container is representing.

Specifications for: type

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required, enumerated] The type of response function

JSON example of type

```
"type": {  
  "attributes": {  
    "source": ["EXFOR"]  
  },  
  "body": "resolution_function"  
}
```

2.8.2 Response functions container: format

A data type to hold how the response function is presented.

Specifications for: format

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required, enumerated] The response function format.

JSON example of format

```

"format": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "equation"
}

```

2.8.3 Response functions container: code

A data type to hold the code that the input file is for. The example is taken from EXFOR entry 21465.

Specifications for: code

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required, enumerated] The code.

JSON example of code

```

"code": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "The response functions for the detector
were calculated using the Monte-Carlo code MATRI, which is a modified
version of the code SPEC-4."
  },
  "body": "MATRI"
}

```

2.8.4 Response functions container: code input

A data type to hold whether or not the input file is provided in the database.

Specifications for: code input

type: text**properties:****attributes:** [object] Attributes of text container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required, enumerated]

JSON example of code input

```

"code_input": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "yes"
}

```

2.8.5 Response functions container: description

A data type to hold any additional text description that explains the response function. The example is taken from EXFOR entry 21465.

Specifications for: description

type: text**properties:****attributes:** [object] Attributes of text container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required] Text description.

JSON example of description

```
"description": {  
  "attributes": {  
    "source": ["EXFOR"]  
  },  
  "body": "The response functions for the detector were  
calculated using the Monte-Carlo code MATRI, which is a modified version  
of the code SPEC-4."  
}
```

2.9 Reference Data

This section describes the reference data used in the experiment. When possible, the entry will link to the data itself. The `data_sources` container is a list of individual `data_source` containers, each of which describe a single source of reference data used in the experiment.

Section Contents:

2.9.1	Reference data container: type	191
2.9.2	Reference data container: reference	191
2.9.3	Reference data container: particle detected	192
2.9.4	Reference data container: reaction	193
2.9.5	Reference data container: decay type	194
2.9.6	Reference data container: decay parent	194
2.9.7	Reference data container: decay daughter	195
2.9.8	Reference data container: values	196
2.9.9	Reference data container: energy	197

2.9.1 Reference data container: type

A data type to hold the type of reference data. The example is taken from EXFOR entry 14718.

Specifications for: type

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required, enumerated] The type of reference data.

JSON example of type

```
"type": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "(57-LA-139(N,G)57-LA-140,,SIG,,SPA) Flux
monitor.
((MONIT)11447094,L.Seren+,J,PR,72,888,1947)"
  },
  "body": "monitor_reaction"
}
```

2.9.2 Reference data container: reference

A data type to hold the main reference for the reference data set. The example is taken from EXFOR entry 14718.

Specifications for: reference

type: reference

properties:

attributes: [object, required] Attributes of reference container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

reference type: [string, enumerated] The type of reference.

content: [object, required] The contents of the reference container. Properties:

formatted: [string] The reference formatted in any bibliographic style.

link: [object] In many cases the file will refer to information outside of the entry, such as DOI's for papers or references to data in other entries. For these, a link container should be created. The information in this container should allow the user to find the extra information easily. The link can be a URL, a library, or a Nuclear Science References (NSR) link.

description: [string] Any free-text description available for the information in the container.

JSON example of reference

```
"reference": {
  "attributes": {
    "source": ["EXFOR"],
    "type": "journal",
    "description": "(57-LA-139(N,G)57-LA-140,,SIG,,SPA) Flux
monitor. ((MONIT)11447094,L.Seren+,J,PR,72,888,1947)"
  },
  "formatted": "Seren L., Friedlander H.N, Turkel S.H., Thermal
Neutron Activation Cross Sections, Phys. Rev. 72, 888 (1947)",
  "link": {
    "attributes": {
      "source": ["EXFOR"],
      "type": "DOI"
    },
    "body": "10.1103/PhysRev.72.888"
  }
}
```

2.9.3 Reference data container: particle detected

A data type to hold the type of particle detected.

Specifications for: particle detected

type: isotope**properties:****attributes:** [object] Attributes of isotope container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required] The isotope in ZZZ-XX-AAA-LL format.

JSON example of particle detected

```

"particle_detected": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "000-n-001"
}

```

2.9.4 Reference data container: reaction

A data type to hold the reaction for monitor reaction cross section, PFNS, or thermal capture gamma intensity reference data. The example is taken from EXFOR entry 14718.

Specifications for: reaction

type: text**properties:****attributes:** [object] Attributes of text container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required] The reaction.

JSON example of reaction

```

"reaction": {
  "attributes": {
    "source": ["EXFOR"],
    "description": "(57-LA-139(N,G)57-LA-140,,SIG,,SPA) Flux
monitor.
((MONIT)11447094,L.Seren+,J,PR,72,888,1947)"
  },
  "body": "57-La-139(n,g)57-La-140"
}

```

2.9.5 Reference data container: decay type

A data type to hold the type of decay.

Specifications for: decay type

type: text

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [string, required, enumerated] The type of decay.

JSON example of decay type

```

"decay_type": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "beta"
}

```

2.9.6 Reference data container: decay parent

A data type to hold the parent isotope of the decay.

Specifications for: decay parent

type: isotope**properties:****attributes:** [object] Attributes of isotope container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required] The isotope in ZZZ-XX-AAA-LL format.

JSON example of decay parent

```

"decay_parent": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "055-Cs-137"
}

```

2.9.7 Reference data container: decay daughter

A data type to hold the daughter nucleus of the decay.

Specifications for: decay daughter

type: isotope**properties:****attributes:** [object] Attributes of isotope container. Properties:**source:** [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.**description:** [string] Any free-text description available for the information in the container.**body:** [string, required] The isotope in ZZZ-XX-AAA-LL format.

JSON example of decay daughter

```

"decay_daughter": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": "056-Ba-137"
}

```

2.9.8 Reference data container: values

A data type to hold the values of the reference data used, if the values are not given by the reference. The example is taken from EXFOR entry 14718.

Specifications for: values

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of values

```

"values": {
  "attributes": {
    "source": ["EXFOR", "publication"],
    "unit": "b",
    "value type": "mean"
  },
  "value": [8.4],
  "description": "(57-LA-139(N,G)57-LA-140,,SIG,,SPA) Flux monitor
((MONIT)11447094,L.Seren+,J,PR,72,888,1947)"
}

```

2.9.9 Reference data container: energy

A data type to hold the value or list of values for the energy values. This can represent incident energy values, such as for monitor cross sections, or can be an energy identifier, such as the energy of a decay gamma.

Specifications for: energy

type: measured quantity

properties:

attributes: [object, required] Attributes of measured quantity container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

unit: [string, required, enumerated] The unit.

value type: [string, required, enumerated] The type of the value.

content: [object, required] The contents of the measured quantity container, which represent the value and covariance for the quantity. Properties:

value: [object, required] A data type to hold a measured value, which has a unit.

uncertainty: [object] This data type holds the information for the uncertainty on a measured value - the value of the uncertainty itself, the unit, and whether the uncertainty is absolute or relative.

description: [string] Any free-text description available for the information in the container.

JSON example of energy

```
"energy": {
  "attributes": {
    "source": ["EXFOR"],
    "unit": "eV",
    "value type": "mean"
  },
  "value": [320076.0],
  "uncertainty": {
    "attributes": {
      "source": ["EXFOR"],
      "unit": "eV",
      "unc type": "absolute"
    },
    "body": [6]
  },
  "description": "decay gamma energy"
}
```

2.10 Correlation

The `correlations` container is a list of individual `correlation` containers, each of which describe a single source of correlation between two data sets.

```
"correlations": [  
    {correlation container},  
    {correlation container}  
]
```

Section Contents:

2.10.1 Correlation container: entry	200
2.10.2 Correlation container: factor	200
2.10.3 Correlation container: shape	201
2.10.4 Correlation container: value	201

2.10.1 Correlation container: entry

A data type to hold the entry number of the correlated data set.

Specifications for: entry

type: number

properties:

attributes: [object] Attributes of text container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [number, required] The entry number.

JSON example of entry

```
"entry": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": 10000
}
```

2.10.2 Correlation container: factor

A data type to hold what factor or factors cause the correlation.

Specifications for: factor

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, or year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required] List of correlation factors.

JSON example of factor

```

"factor": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": ["mass"]
}

```

2.10.3 Correlation container: shape

A data type to hold the shape (energy-dependence) of the correlation in incident energy.

Specifications for: shape

type: text array

properties:

attributes: [object] Attributes of the text array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required, enumerated] List of applicable shapes.

JSON example of shape

```

"shape": {
  "attributes": {
    "source": ["EXFOR"]
  },
  "body": ["constant"]
}

```

2.10.4 Correlation container: value

A data type to hold the value(s) of the correlation.

Specifications for: value

type: numerical array

properties:

attributes: [object] Attributes of array container. Properties:

source: [array] Where the information was obtained from, such as EXFOR, publication, author, library-version, OR year-name.

description: [string] Any free-text description available for the information in the container.

body: [array, required] The value(s) of the correlation.

JSON example of value

```
"value": {  
  "attributes": {  
    "source": ["EXFOR"]  
  },  
  "body": [0.8]  
}
```

Acknowledgments

Work at LANL was carried out under the auspices of the NNSA of the U.S. Department of Energy under contract 89233218CNA000001.

A. Allowed Values

A.1 General Datatypes

A.1.1 Link type

Table A.1: Values allowed for the general type link type.

Allowed value	Description
DOI	DOI link
URL	URL link
NSR	NSR link
library	evaluated library release
database entry	another entry in the database
other	another kind of link

A.1.2 Unit

Table A.2: Values allowed for the general type unit.

Allowed value	Description
g	grams
m	meters
cm3	centimeters cubed
b	barns
eV	electron-volts
s	seconds
year	years
Hz	Hertz
W	Watts
kPa	kilo-Pascal
atm	atmosphere
C	Celsius
K	Kelvin
degree	degree
unitless	dimensionless value

A.1.3 Unc type

Table A.3: Values allowed for the general type unc type.

Allowed value	Description
relative	relative to the measured value
absolute	in the same unit as the measured value
resolution	resolution of the measurement value

A.1.4 Value type

Table A.4: Values allowed for the general type value type.

Allowed value	Description
mean	the expectation value of the quantity
max	the maximum value of the quantity
min	the minimum value of the quantity
range	range of values the quantity falls in
unresolved	several quantities that cannot be resolved from each other
edges	bin edges for histograms
unknown	it is unknown what values represent

A.1.5 Reference type

Table A.5: Values allowed for the general type reference type.

Allowed value	Description
journal	journal article
conference	conference proceeding
report	institutional report
thesis	PhD thesis
personal	personal communication
other	another type of reference
unknown	the type of reference is unknown

A.1.6 Applied

Table A.6: Values allowed for the general type applied.

Allowed value	Description
yes	the correction was applied
no	the correction was not applied

partial	application was incomplete
unknown	it is unknown if the correction was applied
negligible	the effect was negligible, so the correction did not need to be applied.

A.1.7 Energy dependent

Table A.7: Values allowed for the general type energy dependent.

Allowed value	Description
yes	it is energy dependent
no	it is not energy dependent
unknown	it is unknown whether it is energy dependent

A.1.8 methods

Table A.8: Values allowed for the general type methods.

Allowed value	Description
fit	the correction magnitude was fit
subtracted	the correction was performed by subtraction
measured	the correction magnitude was measured
modeled	the correction magnitude was modeled or simulated
filter	the correction effect was reduced using a filter in the beam
saturated_resonance	the correction was performed by the saturated resonance technique
kinematics	the correction was performed by kinematics calculations
rotating_chamber	The chamber is rotated in order to average over the kinematics of the forward boost.
data_table	the correction values were taken from a data table
other	another method was used
unknown	the correction determination and application methods are unknown

A.1.9 components

Table A.9: Values allowed for the general type components.

Allowed value	Description
floor	the floor of the facility
ceiling	the ceiling of the facility
sample_holder	the sample holder in the experimental setup
other	other components were included
unknown	the components included in the correction are unknown

A.1.10 code

Table A.10: Values allowed for the general type code.

Allowed value	Description
SAMMY	Code System for Multilevel Matrix Fits to Neutron and Charged-Particle Cross-Section Data Using Bayes' Equations (Larson, ORNL/TM-9179/R8)
MCNP	Monte Carlo N-Particle code (https://mcnp.lanl.gov/)
REFIT	Multilevel Resonance Parameter Least Square Fit of N Transmission, Capture, Fission & Self Indication Data (Moxon, NEA-0914/09)
SESH	Code for calculating the self-shielding and multiple scattering effects for neutron cross section data interpretation in the unresolved resonance region (Frohner, GA-8380)
GEANT-4	Particle transport code (https://geant4.web.cern.ch/)
SRIM	The Stopping of Energetic Light Ions in Elemental Matter (Ziegler, J Appl Phys 85)
MATRI	A modified version of the SPEC-4 code
CGMF	Fission fragment emission simulation code.
CoH3	CoH3 fast region reaction code.
TALYS	TALYS fast region reaction code.
EMPIRE	EMPIRE fast region reaction code.
other	Another code was used
unknown	the code is unknown

A.2 Observable

A.2.1 type

Table A.11: Values allowed for the observable type type.

Allowed value	Description
CS	Cross section data
CSP	Partial cross section data
CST	Temperature dependent cross section data
DA	Differential data with respect to angle
DAE	Differential data with respect to angle and energy
DAP	Partial differential data with respect to angle
DE	Differential data with respect to energy
DEP	Partial differential data with respect to energy
E	Kinetic energies
FY	Fission product yields
INT	Cross section integral over incident energy
L	Scattering length
MFQ	Differential fission neutron multiplicities
MLT	Outgoing particle multiplicities
NQ	Nuclear quantities
NU	Fission neutron multiplicities
POL	Polarization data
PY	Product yields
RI	Resonance integrals
RP	Resonance parameters
RR	Reaction rates
SP	Gamma spectra
SQ	Special quantities
TT	Thick target yields
TTD	Differential thick target yields
TTP	Partial thick target yields
ZAP	Most probable charge or mass

A.2.2 mf

Table A.12: Values allowed for the observable type mf.

Allowed value	Description
---------------	-------------

2	resonance parameters
3	neutron cross section
4	neutron angular distribution
5	neutron energy distribution
6	neutron energy-angle distribution
13	gamma yield
14	gamma angular distribution
15	gamma energy distribution

A.2.3 mt

A data type to hold the analogous MT number, if applicable, which can be found at the NNDC website. If the measured reaction does not have a single analogous MT number, this key should not be used. The full list is composed of over 60 values, so it is not reproduced in this document.

A.2.4 exfor branch

A data type to hold the branch, which is the 5th quantity in the reaction string. This key can be used to indicate that a partial channel or cumulative reaction is measured. These values are given in Dictionary 031. The full list is composed of over 60 values, so it is not reproduced in this document.

A.2.5 exfor parameter

A data type to hold the parameter, which is the 6th quantity in the EXFOR reaction string. This quantity describes the type of data presented. These values are given in Dictionary 032. The full list is composed of over 60 values, so it is not reproduced in this document.

A.2.6 exfor modifier

A data type to hold the modifier, which is the 8th quantity in the EXFOR reaction string. This quantity holds information about how the data is represented. These values are given in Dictionary 034. The full list is composed of over 60 values, so it is not reproduced in this document.

A.2.7 exfor data code

Table A.13: Values allowed for the observable type exfor data code.

Allowed value	Description
CALC	Calculated data
CRCTD	Data corrected by other than the author

DERIV	Derived data
DEROT	Data derived by other than the author
EVAL	Evaluated data
EXP	Experimental data
RECOM	Recommended data

A.2.8 absolute

Table A.14: Values allowed for the observable type `absolute`.

Allowed value	Description
yes	data is absolute
no	data is relative to another quantity
unknown	it is unknown if the data is absolute

A.2.9 polarized

Table A.15: Values allowed for the observable type `polarized`.

Allowed value	Description
yes	the incident particles were polarized
no	the incident particles were not polarized
unknown	it is unknown if the incident particles were polarized

A.2.10 observable method

Table A.16: Values allowed for the observable type `observable method`.

Allowed value	Description
ABSFY	Absolute fission yield measurement
ACTIV	Activation
AMS	Accelerator mass spectrometry
ASEP	Off-line mass separation of a product
ASPEC	Alpha spectrometry
ASSOP	Associated particle
BCINT	Beam current integrated
BGCT	Beta-gamma coincidence technique
BSPEC	Beta ray spectrometry

BURN	Burn-up
CADMB	Cadmium bath
CHARG	Measurements in gas discharge
CHRFL	Christiansen filter
CHSEP	Chemical separation of products
COINC	Coincidence
DIFFR	Diffraction
DSCAT	Double scattering
EDE	Particle identification by 'E/Delta E' measurement
EDEG	Energy degradation by foils
EXTB	Irradiation with external beam
FISCT	Absolute fission counting
FLUX	Neutron flux monitoring
FNB	Filtered neutron beam
FPGAM	Direct gamma-ray spectrometry
GSPEC	Gamma ray spectrometry
HADT	Heavy atom difference technique
HATOM	Hot atom method
HE-AC	Helium accumulation method
HEJET	Collection by He jet
INTB	Irradiation with internal beam
JET	Collection by gas jet
LRASY	Left-right asymmetry
MAGFR	Magnetic field rotation
MANGB	Manganese bath
MOMIX	Mixed monitor
MOSEP	Separate monitor foil
OLMS	On-line mass separation of a product
PHD	Pulse-height discrimination
PHWT	Pulse-height weighting technique
PLSED	Pulse die-away
PSD	Pulse-shape discrimination
REAC	Reactivity measurement
REC	Collection of recoils
REFL	Total reflection from mirrors
RELFY	Relative fission yield measurement
RINGR	Ring ratio method
RVAL	R-value measurement

SFLIP	Spin flip
SHELT	Shell transmission
SITA	Single target irradiation
SLODT	Slowing-down time
STATD	Statistically determined
STTA	Stacked target irradiation
THERM	Gas thermochromatographic separation
TOF	Time-of-flight
TOFDE	Particle identification by 'TOF-Delta E' measurement
TRN	Transmission method
TTM	Thick-target method
XSPEC	X-ray spectrometry

A.2.11 analysis method

Table A.17: Values allowed for the observable type analysis method.

Allowed value	Description
4PI1A	4pi times differential cross section at one angle
ANC	Asymptotic normalization constant
AREA	Area analysis
CHGDS	Corrected for charge distribution
CORAB	Correction on isotopic abundance
DECAY	Decay curve analysis
DIFFR	Difference spectrum
DTBAL	Detailed balance
ERCSN	Extracted from Ericson fluctuation
INTAD	Integration of angular distribution
INTED	Integration of energy distribution
INTEF	Integration of excitation function
INTPD	Integration of momentum distribution
LEAST	Least structure method
MLA	Multilevel analysis
NTRTH	Normalized to Rutherford scattering cross section
PES	Extracted from equilibrium particle emission spectra
PGS	Extracted from primary gamma spectra
PHDIF	Photon difference
PLA	Penfold-Leiss method
REDUC	Reduction method

REGUL	Regularization method
RFN	R-function formalism
SHAPE	Shape analysis
SLA	Single level analysis
SURGT	Surrogate reaction method
THIES	Thies's method
TROJA	Trojan-horse method
TTUNF	Calculated from thick target using unfolding proced.
UNFLD	Unfolding procedure
WSP	Woods-Saxon potential

A.2.12 reference frame

Table A.18: Values allowed for the observable type reference frame.

Allowed value	Description
laboratory	laboratory frame
COM	center of mass frame
inverse_kinematics	inverse kinematics
unknown	the reference frame is unknown

A.2.13 measured dependency

Table A.19: Values allowed for the observable type measured dependency.

Allowed value	Description
energy	particle energy
mass	particle mass
angle	particle angle
yield	particle yield
unknown	the dependency is unknown

A.2.14 exit dependency

Table A.20: Values allowed for the observable type exit dependency.

Allowed value	Description
energy	particle energy
mass	particle mass

angle	particle angle
yield	particle yield
unknown	the dependency is unknown

A.3 Bibliography

A.3.1 institute

A data type to hold the EXFOR institute codes keyword, which are given in Dictionary 003. The example is taken from EXFOR entry 13801. The full list is composed of over 60 values, so it is not reproduced in this document.

A.3.2 status

Table A.21: Values allowed for the bibliography type status.

Allowed value	Description
APRVD	Approved by author
BERMN	Data converted from file of B.Berman
COREL	Data correlated with another data set
CPX	Data taken from data file of McGowan, et al.
CRCTD	Data corrected by other than author
CURVE	Data read from a curve
DASTR	Data converted from DASTAR file
DEP	Dependent data
ECSIL	Data converted from ECSIL library
NACRE	Data converted from NACRE files
NGHKD	Authenticity not confirmed
NDD	Data converted from NEUDADA file
OUTDT	Normalization out-of-date
PENTZ	Data converted from GMA input file of W.P.Poenitz
PRELM	Preliminary data
RCALC	Ratio to standard calculated by other than author
RIDER	Data converted from file of B.F.Rider
RNORM	Data renormalized by other than author
SCSRS	Data converted from SCISRS file
SGMBS	Data converted from Sigmabase
SPSDD	Data superseded or withdrawn
SUPPL	Supplemental information compiled separately
TABLE	Data presented by authors

UNOBT	Data unobtainable from author
-------	-------------------------------

A.3.3 alteration

Table A.22: Values allowed for the bibliography type alteration.

Allowed value	Description
rebinned	data in entry has a different bin structure
other	another alteration was made

A.4 Incident

A.4.1 facility

Table A.23: Values allowed for the incident type facility.

Allowed value	Description
ACCEL	Accelerator
BETAT	Betatron
CCW	Cockcroft-Walton accelerator
CHOPF	Fast chopper
CHOPS	Slow chopper
CRASS	Critical assembly
CYCLO	Cyclotron
CYCTM	Tandem cyclotrons
CYGFF	Cyclograaff
DYNAM	Dynamitron
ESTRG	Electron storage ring
FNS	Fusion neutron source
ICTR	Insulated core transformer accelerator
INTFM	Interferometer
ISOCY	Isochronous cyclotron
LASER	Laser system
LCEXP	Location of experiment
LINAC	Linear accelerator
MICRT	Microtron
NGEN	Neutron generator
OLMS	On-line mass separator
OSCIP	Pile oscillator
REAC	Reactor

SELVE	Velocity selector
SPECC	Crystal spectrometer
SPECDD	Double mass spectrometer
SPECCM	Mass spectrometer
SRING	Storage ring
SYNCH	Synchrotron
SYNCY	Synchro cyclotron
VDG	Van de Graaff
VDGT	Tandem van de Graaff

A.4.2 particle source

Table A.24: Values allowed for the incident type particle source.

Allowed value	Description
A-BE	Alpha-Beryllium
AM-BE	Americium-Beryllium neutron source
ARAD	Annihilation radiation
ATOMI	Atomic beam source
B11-H	$1\text{H}(11\text{B},n)$
BRST	Bremsstrahlung
CF252	Spont.fission of Californium-252
CM-BE	Curium-Beryllium neutron source
CM244	Spont.fission of Curium-244
CM246	Spont.fission of Curium-246
CM248	Spont.fission of Curium-248
COMPT	Compton scattering
COULX	Coulomb excitation
D-BE	$9\text{Be}(d,n)$
D-C12	$12\text{C}(d,n)$
D-C14	$14\text{C}(d,n)$
D-D	$2\text{H}(d,n)$
D-LI	$\text{Li}(d,n)$
D-LI7	$7\text{Li}(d,n)$
D-N14	$14\text{N}(d,n)$
D-N15	$15\text{N}(d,n)$
D-T	$3\text{H}(d,n)$
EVAP	Evaporation neutrons
EXPLO	Nuclear explosive device

FRAGM	Fragmentation
HARD	Hardened
ISOL	Online isotope separation
KINDT	Kinematically determined
LAMB	Lamb-shift source
LASER	Laser scattering
LCS	Laser Compton Scattered Photons
MPH	Monoenergetic photons
P-BE	$^9\text{Be}(p,n)$
P-D	$^2\text{H}(p,n)$
P-LI7	$^7\text{Li}(p,n)$
P-N15	$^{15}\text{N}(p,n)$
P-T	$^3\text{H}(p,n)$
PHOTO	Photo-neutron
PO-BE	Polonium-Beryllium neutron source
POLIS	Polarized ion source
POLNS	Polarized neutron source
POLPS	Polarized photon source
POLTR	Polarized target
PU-BE	Plutonium-Beryllium neutron source
PU240	Spont.fission of Plutonium-240
PU242	Spont.fission of Plutonium-242
QMPH	Quasi-monoenergetic photons
RA-BE	Radium-Beryllium neutron source
REAC	Reactor
RN-BE	Radon-Beryllium neutron source
SPALL	Spallation
TAGD	Electron tagged
TH-BE	Thorium-Beryllium neutron source
THCOL	Thermal column
THRDT	Determined by threshold technique
VPH	Virtual photons

A.4.3 energy calibration

Table A.25: Values allowed for the incident type energy calibration.

Allowed value	Description
---------------	-------------

known_resonance	the neutron energy is calibrated using known resonance energies
coincidence	coincidence measurements
alpha_monitor	the alpha-monitor method was used
mono_energetic	the incident particle source is mono-energetic
angle	the angle of the outgoing particle
associated_particle	the associated particle method was used
other	another method was used
unknown	the method of energy calibration is unknown

A.4.4 fluence method

Table A.26: Values allowed for the incident type fluence method.

Allowed value	Description
direct_measurement	a detector is used to directly measure the flux
monitor_reaction	a monitor reaction was used to normalize the flux
associated_particle	the associated particle method was used
manganese_bath	a manganese bath was used to determine the flux
simulation	the flux was simulated with a transport code
calculation	the flux was calculated without a transport code
other	another method was used
unknown	the method of flux determination is unknown

A.4.5 beam stability monitored

Table A.27: Values allowed for the incident type beam stability monitored.

Allowed value	Description
yes	beam stability was monitored
no	beam stability was not monitored
unknown	it is unknown if the beam stability was monitored

A.4.6 accelerator stability monitored

Table A.28: Values allowed for the incident type accelerator stability monitored.

Allowed value	Description
yes	accelerator stability was monitored
no	accelerator stability was not monitored
unknown	it is unknown if the accelerator stability was monitored

A.4.7 stable target

Table A.29: Values allowed for the incident type `stable target`.

Allowed value	Description
yes	target was stable
no	target was not stable
unknown	it is unknown if the target was stable

A.5 Detector

A.5.1 type

Table A.30: Values allowed for the detectors type `type`.

Allowed value	Description
BAF2	Scintillator BaF2
BF3	Boron Trifluoride neutron detector
BGO	Bismuth-Germanate crystal detector
BPAIR	Electron-pair spectrometer
BUBLC	Bubble chamber
CEREN	Cerenkov detector
COIN	Coincidence counter arrangement
COMPL	Compton Polarimeter
CSICR	Cesium-Iodide crystal
D4PI	4pi detector
DRFTC	Drift chamber
FISCH	Fission chamber
GE	Germanium detector
GE-IN	Germanium intrinsic detector
GELI	Germanium-Lithium detector
GEMUC	Geiger-Mueller counter
GLASD	Glass detector

HE3SP	He-3 spectrometer
HORBU	Hornyak button detector
HPGE	Hyperpure Germanium detector
IMPSI	Passivated implanted planar Si detector
IOCH	Ionization chamber
LABR3	LaBr3 scintillator
LEGE	Low energy Germanium Detector
LONGC	Long counter
MAGSP	Magnetic spectrometer
MCPLT	Microchannel plate
MOXR	Moxon-Rae detector
MTANK	Moderating tank detector
MWDC	Multi-wire drift chamber
MWPC	Multi-wire proportional counter
MWSC	Multi-wire spark counter
NAICR	Sodium-Iodide crystal
PGAC	Parallel-grid avalanche detector
PHVC	Photovoltaic Cell
PLATE	Nuclear plates
PPAC	Parallel plate avalanche counter
PROPC	Proportional counter
PS	Position sensitive detector
SCIN	Scintillation detector
SI	Silicon detector
SIBAR	Silicon surface barrier detector
SILI	Silicon-Lithium detector
SISD	Silicon strip detector
SOLST	Solid-state detector
SPEC	Large spectrometer system
STANK	Scintillator tank
SWPC	Single-wire proportional counter
TELES	Counter telescope
TFBC	Thin-film Breakdown Counter
THRES	Threshold detector
TPC	Time projection chamber
TRD	Track detector
XHPGE	Extended range Germanium gamma detector

A.5.2 use

Table A.31: Values allowed for the detectors type use.

Allowed value	Description
observable	measurement of the observable
monitor	measurement of a monitor reaction for normalization
flux	measurement of the flux
beam_stability	measurement of the beam or current
background	measurement of the background
detector_response	measurement of a detector response function
other	measurement of another quantity
unknown	the use of the detector is unknown

A.5.3 geometry

Table A.32: Values allowed for the detectors type geometry.

Allowed value	Description
sphere	the detector is a sphere.
cylinder	the detector is a cylinder.
plate	the detector is a plate.
cube	the detector is a cube.
other	the detector is of another geometry
unknown	the geometry is unknown.

A.5.4 efficiency method

Table A.33: Values allowed for the detectors type efficiency method.

Allowed value	Description
calibration_sources	calibration sources
simulation	the efficiency was simulated
alpha_spectroscopy	alpha spectroscopy method
pulse_height_mass_estimate	the pulse height discrimination mass estimate method
rotating_chamber	the chamber is rotated in order to average over the kinematics of the forward boost.
ratio_measurement	the efficiency was determined by a ratio to a well-known standard.
other	another method was used
unknown	the efficiency calibration method is unknown

A.5.5 energy calibration method

Table A.34: Values allowed for the detectors type energy calibration method.

Allowed value	Description
TOF	time-of-flight
calibration_sources	calibration sources
simulation	the energy was simulated
other	another method was used
unknown	the energy calibration method is unknown

A.5.6 coincidence measurement

Table A.35: Values allowed for the detectors type coincidence measurement.

Allowed value	Description
yes	the measurement was done in coincidence
no	the measurement was not done in coincidence
unknown	it is unknown if the measurement was done in coincidence

A.6 Sample

A.6.1 use

Table A.36: Values allowed for the samples type use.

Allowed value	Description
observable	measurement of the observable
monitor	measurement of a monitor reaction for normalization
flux	measurement of the flux
beam_stability	measurement of the beam or current
other	measurement of another quantity
unknown	the use of the sample is unknown

A.6.2 configuration

Table A.37: Values allowed for the samples type configuration.

Allowed value	Description
single	a single sample was used
back_to_back	the samples were placed back to back
stacked	the samples were stacked
parallel_plate	parallel plates were used
multi_plate	a multi-plate configuration
other	another configuration was used
unknown	the sample configuration is unknown

A.6.3 composition methods

Table A.38: Values allowed for the samples type composition methods.

Allowed value	Description
alpha_counting	the alpha counting method
weighting	the weighting method
mass_spectroscopy	mass spectroscopy was used
chemical_analysis	chemical analysis was used
destructive_analysis	destructive analysis was used
other	another method was used
unknown	it is unknown what composition methods were used

A.6.4 type

Table A.39: Values allowed for the samples type type.

Allowed value	Description
metal	metal sample
gas	gas sample
liquid	liquid sample
powder	powder sample
other	another sample type
unknown	the sample type is unknown

A.6.5 geometry

Table A.40: Values allowed for the samples type geometry.

Allowed value	Description
cylinder	cylindrical sample
sphere	spherical sample
plate	a rectangular plate sample
disk	a round disk sample
cube	a cube sample
other	another geometry
unknown	the sample geometry is unknown

A.6.6 production methods

Table A.41: Values allowed for the samples type production methods.

Allowed value	Description
rolled	the sample was rolled
pressed	the sample was pressed powder
electroplated	the sample was electroplated
electro_sprayed	the sample was electro-sprayed
electro_deposited	the sample was electro-deposited
vapor_deposited	the sample was vapor-deposited
painted	the sample was painted
vacuum_evaporation	the sample was vacuum evaporated
other	another method was used
unknown	the production method was unknown

A.6.7 density methods

Table A.42: Values allowed for the samples type density methods.

Allowed value	Description
scale	the weight was measured with a scale
micro_scale	the weight was measured with a micro-scale
calipers	the dimensions were measured with calipers
alpha_counting	the density was measured with alpha counting
beta_counting	the density was measured with alpha counting
isotopic_composition	the density was calculated based on the composition
destructive_analysis	the density was determined by destructive analysis
other	another method was used
unknown	the density methods are unknown

A.7 Response Functions

A.7.1 type

Table A.43: Values allowed for the response functions type type.

Allowed value	Description
resolution_function	the resolution function for the measurement
detector_response	detector response function
other	another type of response function
unknown	the type of response function is unknown

A.7.2 format

Table A.44: Values allowed for the response functions type format.

Allowed value	Description
equation	an equation
tabular	a table of values
input_file	an input file for a code
other	another method was used

A.7.3 code

Table A.45: Values allowed for the response functions type code.

Allowed value	Description
SAMMY	Code System for Multilevel Matrix Fits to Neutron and Charged-Particle Cross-Section Data Using Bayes' Equations (Larson, ORNL/TM-9179/R8)
MCNP	Monte Carlo N-Particle code (https://mcnp.lanl.gov/)
REFIT	Multilevel Resonance Parameter Least Square Fit of N Transmission, Capture, Fission & Self Indication Data (Moxon, NEA-0914/09)
SESH	Code for calculating the self-shielding and multiple scattering effects for neutron cross section data interpretation in the unresolved resonance region (Frohner, GA-8380)
GEANT-4	Particle transport code (https://geant4.web.cern.ch/)
SRIM	The Stopping of Energetic Light Ions in Elemental Matter (Ziegler, J Appl Phys 85)

MATRI	A modified version of the SPEC-4 code
other	Another code was used
unknown	the code is unknown

A.7.4 code input

Table A.46: Values allowed for the response functions type code input.

Allowed value	Description
yes	a code input is available
no	a code input is not available

A.8 Reference Data

A.8.1 type

Table A.47: Values allowed for the reference data type type.

Allowed value	Description
monitor_reaction	cross section for a monitor reaction
decay_half_life	the half-life of a decay
decay_intensity	the intensity of a decay emission
atomic_mass	atomic mass
PFNS	a prompt fission neutron spectrum
thermal_capture_gamma_intensity	the intensity of a gamma from thermal capture
other	another type of reference data

A.8.2 decay type

Table A.48: Values allowed for the reference data type decay type.

Allowed value	Description
beta	beta decay
alpha	alpha decay
gamma	gamma emission
electron_capture	electron capture decay
internal_conversion	internal conversion decay
spontaneous_fission	spontaneous fission

other	another type of decay
unknown	the decay type is unknown

A.9 Correlation

A.9.1 shape

Table A.49: Values allowed for the correlation type shape.

Allowed value	Description
constant	constant correlation across energy
gaussian	the correlation has a Gaussian shape (is higher for values closer in energy, and lower for values further apart in energy)
table	the correlation is presented as a table of values
other	another shape
unknown	the correlation shape is unknown

References

- Herman, M., & Trkov, A. (2010). ENDF-6 Formats Manual (tech. rep. BNL-90365-2009). Brookhaven National Laboratory.
- Json 2020-12 release. (2020). <https://json-schema.org/draft/2020-12/release-notes#2020-12-release-notes>
- Otsuka, N. (2011). EXFOR Formats Manual (tech. rep. IAEA-NDS-207). International Atomic Energy Agency. www-nds.iaea.org/nrdc/basics
- Otsuka, N., Dupont, E., Semkova, V., et al. (2014). Towards a More complete and accurate experimental nuclear reaction data library (EXFOR): International collaboration between nuclear reaction data centres (NRDC). Nuclear Data Sheets, 120, 272–276. <https://doi.org/10.1016/j.nds.2014.07.065>
- Schwerer, O. (2015). LEXFOR (EXFOR Compiler's Manual) (tech. rep. IAEA-NDS-208). International Atomic Energy Agency.

Index

- Associated Particle
 - gas cell effects correction, 87
 - gas cell pressure, 84
 - gas cell temperature, 85
 - gas cell temperature correction, 86
 - particle detected, 81
 - target atoms, 82
 - target stability, 83
 - target thickness, 81
- beam
 - collimators, 79
 - energy, 62, 76
 - fluctuations, 64
 - incident particle, 24
 - moderators, 80
 - uniformity, 65
- correction
 - angular distortion, 117
 - anisotropy, 88
 - background, 44
 - beam overlap, 176
 - counting geometry, 131
 - deadtime, 109
 - delayed gamma, 118
 - discriminator, 113
 - displacement, 172
 - electronic noise, 114
 - false capture, 126
 - false fission, 119
 - fluctuation, 64
 - forward boost, 115
 - french effect, 121
 - gamma angular distribution, 130
 - gamma attenuation, 178
 - gas cell effects, 87
 - gas cell temperature, 86
 - hs content, 122
 - impurity, 150
 - manganese capture, 124
 - multiple sample effects, 182
 - multiple scattering, 174
 - neutron absorption, 175
 - pulse height extrapolation, 139
 - pulse pileup, 111
 - resonance self shielding, 180
 - roughness, 171
 - secondary group, 67
 - spectrum unfolding, 127
 - stopping power, 137
 - sum peaks, 128
 - uniformity, 65, 170
- EXFOR keywords
 - analysis method, 36
 - comments, 56
 - critique, 56
 - detector, 94
 - facility, 60
 - history, 55
 - institute, 53
 - observable method, 35
 - reaction branch, 30
 - reaction data, 33
 - reaction modifier, 32
 - reaction parameter, 31
 - reaction particle considered, 32
 - reaction process, 29
 - reaction quantity, 24
 - source, 60
 - status, 57
- fission chamber
 - pulse height extrapolation correction, 139

stopping power correction, 137

gamma

angular distribution correction, 130

attenuation correction, 178

delayed gamma correction, 118

TOF

false capture correction, 126

flight path, 70

micro pulse width, 73

overlap correction, 67

pulse width, 72

repetition rate, 74

secondary group correction, 67

wrap around correction, 67

zero time offset, 75