BWmeta 1.2.0 Schema Reference

(Revision 13514)

BWmeta is a general-purpose metadata format capable of describing entities such as: academic articles, books, audio recordings, laws, or molecular sequences. A distinguishing feature of BWmeta is non-redundant, hierarchical storage of information. For example, metadata for an article is typically distributed across four records: one for the article, one for the issue in which the article was published, one for the volume, and one for the journal. The root tag of any BWmeta document is <bwmeta/>, which may contain any number of top-level objects (<person/>, <element/>, <license/>, etc.). BWmeta is a flexible format: there are no enumerations of accepted values (most notably in various "type" attributes). Instead, data providers are encouraged to define their own vocabularies.

affiliation

Contributors' affiliation. Contains an identifier (see: id attribute) that may be referenced by one or more <affiliation-ref/> tags. Presentational form is stored in <text/>. The optional <identity/> child tag references an institution (see: <institution/> tag).

Attributes:

✔ **id** (required): Affiliation identifier. Referenced by <affiliation-ref/>.

Contents:

(attribute*, identity?, text)

affiliation-ref

Reference to an affiliation.

Attributes:

✔ **ref** (required): Identifier of an affiliation.

Contents:

(empty)

ancestor

Ancestor element in element structure.

Attributes:

✔ **level** (required): Level on which the ancestor is located.

✔ **identity** (optional): Reference to the ancestor element.

✔ **position** (optional): Position within the level.

Contents:

(attribute*, description*, name*)

attribute

A tree-like structure with a key-value pair in each node. May be used to hold information that does not map well to other tags.

Attributes:

✔ **key** (required): Attribute's key.

✔ **value** (optional): Attribute's value.

Contents:

(attribute*)

`bwmeta`

Root of any BWmeta document. Top-level objects, such as persons or elements, are direct descendants.

Contents:

(category | classification | element | hierarchy | id-scheme | institution | level | license | person)*

`category`

Category within a classification. Contains code, name and description of the category, as well as code of the parent category (if applicable).

Attributes:

✔ **id** (required): Object identifier.

✔ **version** (optional): Object version.

✔ **classification** (required): Identifier of the classification to which the category belongs.

✔ **code** (required): Code of the category. Examples: 83A05, 03.30.+p.

✔ **parent** (optional): The parent category's code. If present, both the category and its parent must belong to the same classification.

Contents:

(attribute*, description*, name*)

`category-ref`

Reference to a given category.

Attributes:

✔ **classification** (required): Identifier of the classification containing the referenced category.

✔ **code** (required): Code of the referenced category.

Contents:

(empty)

`classification`

Classification or taxonomy, a hierarchical structure of categories. Notable examples include subject classifications, such as: MSC or PACS.

Attributes:

✔ **id** (required): Object identifier.

✔ **version** (optional): Object version.

Contents:

(attribute*, description*, name*)

contents

Files and directories containing the element.

Contents:

(directory | file)*

contributor

A person or institution that contributed to this element.

Attributes:

✔ **institution** (required): Whether the contributor is an institution or a person. This is a boolean field. Value "true" indicates an institution, "false" indicates a person.

✔ **role** (required): Contributor's role. Examples: author, editor, translator.

Contents:

(affiliation-ref*, attribute*, description*, identity?, name*)

current

Current element in element structure.

Attributes:

✔ **level** (required): Current element's level.

✔ **position** (optional): Current element's relative position within this level.

Contents:

(attribute*)

date

Date of an event associated with a given element. The text present in tag's body contains a presentational form of the date, while year, month and day attributes contain the actual (parsed) date. Lack of day (or day and month) attribute(s) should be interpreted as a date specified with a coarser granularity. For example: <date type="published" year="1809" month="9">September 1809</date> should be interpreted as indication of a month.

Attributes:

✔ **day** (optional): Day.

✔ **month** (optional): Month. If not present, day should not be present either.

✔ **type** (required): Type of date. Examples: accepted, published, received.

✔ **year** (optional): Year. If absent, day and month should be absent as well.

Contents:

(text content)

description

Multi-line description of an object.

Attributes:

✔ **lang** (optional): Language in which the description is written. Only ISO 639-1 and 639-2 language codes are allowed.

✔ **type** (optional): Type of the description. Examples: abstract, comment, summary.

Contents:

(text content)

directory

Directory containing files and other directories. It provides a logical, hierarchical structure of files.

Attributes:

✔ **id** (required): Directory identifier.

✔ **langs** (optional): List of languages of this content. Only ISO 639-1 and 639-2 language codes are allowed.

✔ **type** (required): Type of content. Examples: pages.

Contents:

(attribute*, description*, (directory | file)+, name*)

element

Element describes a document, or in the case of compound documents, an atomic part of it. For example — in the case of academic articles in journals — the article, the issue containing it, the volume and the journal, are each respresented by a separate element.

Attributes:

✔ **id** (required): Object identifier.

✔ **version** (optional): Object version.

Contents:

(affiliation*, attribute*, category-ref*, contents?, contributor*, date*, description*, id*, language*, license-ref*, name*, relation*, structure*, tags*)

file

Description of a file containing the element or part of it. File is understood as a fixed array of bytes that can be stored in one or more locations.

Attributes:

✔ **format** (required): MIME format of the file.

✔ **id** (required): File identifier.

✔ **langs** (optional): List of languages of this content. Only ISO 639-1 and 639-2 language codes are allowed.

✔ **size** (optional): Size of the file in bytes.

✔ **type** (required): Type of content. Examples: cover, full-text, plain-text, thumbnail.

Contents:

(attribute*, description*, location+, name*, signature*)

format

Format of identifiers in a given <id-scheme/> as a regular expression.

Contents:

(text content)

hierarchy

Hierarchy defines a logical organization of content in various types of compound documents. Examples: journal-volume-issue-article, series-book-chapter, album-disc-track. See <level/> for more details.

Attributes:

✔ **id** (required): Object identifier.

✔ **version** (optional): Object version.

Contents:

(attribute*, description*, name*)

id

Element identifier in a given scheme (such as: DOI, ISBN).

Attributes:

✔ **scheme** (required): Identifier of the scheme (refers to <id-scheme/>).

✔ **value** (required): Element identifier within a given scheme.

Contents:

(empty)

id-scheme

Describes a standard of identifing documents. Serves as a kind of identifier namespace. Examples of identifier schemes: DOI, ISBN, ISSN, PMID.

Attributes:

✔ **id** (required): Object identifier.

✔ **version** (optional): Object version.

Contents:

(attribute*, description*, format?, name*)

identity

Reference to an object (<person/>, <institution/>) that matches a given textual representation. Provides a semantics to what is otherwise a presentational form of an object. For example, <contributor/> within <element/> may list a John Smith as an author and <identity/> may link this name to an actual <person/>.

Attributes:

✔ **ref** (required): Identifier of the referenced object.

Contents:

(empty)

institution

Information on an institution.

Attributes:

✔ **id** (required): Object identifier.

✔ **version** (optional): Object version.

✔ **parent** (optional): Parent institution.

Contents:

(attribute*, description*, name*)

language

Language of an element.

Attributes:

✔ **lang** (required): ISO 639-1 or ISO 639-2 language code.

Contents:

(empty)

level

Level within a given hierarchy describes an element type. Examples: journal, journal volume, journal issue, article in a journal, book series, book in a series, chapter in a book, music album, disc in an album, track on a disc. Levels are grouped into hierarchies.

Attributes:

✔ **id** (required): Object identifier.

✔ **version** (optional): Object version.

✔ **hierarchy** (required): Identifier of the hierarchy in which the given level is located. Example: journal issue is located in the journal-volume-issue-article hierarchy.

✔ **parent** (optional): Identifier of the parent level. Example: journal volume is the parent of journal issue.

Contents:

(attribute*, description*, name*)

license

Description of a license.

Attributes:

✔ **id** (required): Object identifier.

✔ **version** (optional): Object version.

Contents:

(attribute*, description*, name*)

license-ref

License of a given element (refers to <license/>).

Attributes:

✔ **ref** (required): Identifier of the license on which a given element is distributed.

Contents:

(empty)

location

URL describing location of a given file.

Contents:

(text content)

name

One-line name of an object.

Attributes:

✔ **lang** (optional): Language in which the name is written. Only ISO 639-1 and 639-2 language codes are allowed.

✔ **sort-key** (optional): Key to be used in collation.

✔ **type** (optional): Type of the name. Examples: abbreviation, romanization, subtitle, surname, title.

Contents:

(text content)

person

Information on a person.

Attributes:

✔ **id** (required): Object identifier.

✔ **version** (optional): Object version.

Contents:

(attribute*, description*, name*)

relation

Relation between the given element and some other element.

Attributes:

✔ **id-scheme** (optional): Identifier scheme.

✔ **id-value** (optional): Identifier of the other element withing the given identifier scheme.

✔ **type** (required): Type of relation. Examples: references, updates.

Contents:

(attribute*)

signature

Checksum of a binary file.

Attributes:

✔ **type** (optional): Checksum type. Examples (non-normative): SHA1, MD5.

Contents:

(text content)

`structure`

Structure provides links to upper-level elements according to a given hierarchy.

Attributes:

✔ **hierarchy** (required): Hierarchy of this structure.

Contents:

(ancestor*, current)

`tag`

A single tag within a set of tags. See <tags/>.

Contents:

(text content)

`tags`

A set of tags describing the given element.

Attributes:

✔ **lang** (optional): Language of the tags. Only ISO 639-1 and 639-2 language codes are allowed.
✔ **type** (required): Tag type. Examples: genre, keyword, subject.

Contents:

(tag+)

`text`

Text of an affiliation. This is a presentational form of the affiliation, as in the described document.

Contents:

(text content)