4.10. DDL2 dictionary

BY J. D. WESTBROOK AND S. R. HALL

This is version 2.1.3 of the dictionary definition language (DDL2) that provides a machine-readable description of the attributes of data items in the mmCIF and related dictionaries (Chapters 4.5 to 4.7). This version of DDL is described in Chapter 2.6.

The category groups in the DDL2 dictionary are: ddl_group (component categories of the macromolecular DDL); data-block_group (categories that describe the characteristics of data blocks); category_group (categories that describe the characteristics of categories); sub_category_group (categories that describe the characteristics of subcategories); item_group (categories that describe the characteristics of data items); dictionary_group (categories that describe the dictionary); and compliance_group (categories that are retained specifically for compliance with older versions of the DDL).

CATEGORY

Attributes defining the functionality for the entire category.
Category group(s): **ddl_group**
category_group
Category key(s): **_category.id**

***_category.description** (*text*)
Text description of a category.

[category]

***_category.id** (*idname*)
The identity of the data category. Data items may only be looped with items of the same category.
The following item(s) have an equivalent role in their respective categories:
_category_examples.id,
_category_group.category_id,
_category_key.id,
_category_methods.category_id,
_item.category_id. **[category]**

_category.implicit_key
An identifier that may be used to distinguish the contents of like categories between data blocks.

***_category.mandatory_code** (*code*)
Whether the category must be specified in a dictionary.

[category]

CATEGORY_EXAMPLES

Example applications and descriptions of data items in this category.
Category key(s): **_category_examples.id**
_category_examples.case

***_category_examples.case** (*text*)
A case of examples involving items in this category.

[category_examples]

_category_examples.detail (*text*)
A description of an example given in **_category_examples.case**.

[category_examples]

CATEGORY_GROUP

Provides a list of category groups to which the base category belongs.
Category group(s): **ddl_group**
category_group
Category key(s): **_category_group.id**
_category_group.category_id

CATEGORY_GROUP_LIST

This category provides the definition of each category group. A category group is a collection of related categories.
Category group(s): **ddl_group**
category_group
Category key(s): **_category_group_list.id**

***_category_group_list.description** (*text*)
Text description of a category group.

[category_group_list]

***_category_group_list.id** (*idname*)
The name of a category group.
The following item(s) have an equivalent role in their respective categories:
_category_group.id,
_category_group_list.parent_id. **[category_group_list]**

_category_group_list.parent_id
The name of the optional parent category group.

CATEGORY_KEY

This category holds a list of the item names that uniquely identify the elements of the category.
Category group(s): **ddl_group**
category_group
Category key(s): **_category_key.name**
_category_key.id

_category_key.name
The name of a data item that serves as a key identifier for the category (*e.g.* a component of the primary key).

CATEGORY_METHODS

Attributes specifying the association between categories and methods.
Category group(s): **ddl_group**
category_group
Category key(s): **_category_methods.method_id**
_category_methods.category_id

Affiliations: JOHN D. WESTBROOK, Protein Data Bank, Research Collaboratory for Structural Bioinformatics, Rutgers, The State University of New Jersey, Department of Chemistry and Chemical Biology, 610 Taylor Road, Piscataway, NJ, USA; SYDNEY R. HALL, School of Biomedical and Chemical Sciences, University of Western Australia, Crawley, Perth, WA 6009, Australia.

DATABLOCK

Attributes defining the characteristics of a data block.
Category group(s): **ddl_group**
datablock_group
Category key(s): **_datablock.id**

*__datablock.description__ (*text*)
Text description of the data block.
[datablock]

(*)__datablock.id__ (*code*)
The identity of the data block.
The following item(s) have an equivalent role in their respective categories:
_datablock_methods.datablock_id,
_dictionary.datablock_id,
_category.implicit_key. **[datablock]**

DATABLOCK_METHODS

Attributes specifying the association between data blocks and methods.
Category group(s): **ddl_group**
datablock_group
Category key(s): **_datablock_methods.method_id**
_datablock_methods.datablock_id

DICTIONARY

Attributes for specifying the dictionary title, version and data-block identifier.
Mandatory category.
Category group(s): **ddl_group**
datablock_group
dictionary_group
Category key(s): **_dictionary.datablock_id**

__dictionary.datablock_id__
The identifier for the data block containing the dictionary.

*__dictionary.title__ (*char*)
Title identifier of the dictionary.
[dictionary]

__dictionary.version__
A unique version identifier for the dictionary.

DICTIONARY_HISTORY

Attributes for specifying the revision history of the dictionary.
Category group(s): **ddl_group**
dictionary_group
Category key(s): **_dictionary_history.version**

*__dictionary_history.revision__ (*text*)
Text description of the dictionary revision.
[dictionary_history]

*__dictionary_history.update__ (*yyyy-mm-dd*)
The date that the last dictionary revision took place.
[dictionary_history]

*__dictionary_history.version__ (*char*)
A unique version identifier for the dictionary revision.
The following item(s) have an equivalent role in their respective categories:
_dictionary.version. **[dictionary_history]**

ITEM

Attributes which describe the characteristics of a data item.
Category group(s): **ddl_group**
item_group
Category key(s): **_item.name**

*__item.mandatory_code__ (*code*)
Signals whether the defined item is mandatory for the proper description of its category.
The data value must be one of the following:

yes	required item in this category
no	optional item in this category
implicit	required item but may be determined from context

[item]

(*)__item.name__ (*name*)
Data name of the defined item.
The following item(s) have an equivalent role in their respective categories:
_category_key.name,
_item_aliases.name,
_item_default.name,
_item_dependent.name,
_item_dependent.dependent_name,
_item_description.name,
_item_enumeration.name,
_item_examples.name,
_item_linked.child_name,
_item_linked.parent_name,
_item_methods.name,
_item_range.name,
_item_related.name,
_item_related.related_name,
_item_type.name,
_item_type_conditions.name,
_item_structure.name,
_item_sub_category.name,
_item_units.name. **[item]**

ITEM_ALIASES

This category holds a list of possible alias names or synonyms for each data item. Each alias name is identified by the name and version of the dictionary to which it belongs.
Category key(s): **_item_aliases.alias_name**
_item_aliases.dictionary
_item_aliases.version

*__item_aliases.alias_name__ (*aliasname*)
Alias name for the data item.
[item_aliases]

*__item_aliases.dictionary__ (*char*)
The dictionary in which the alias name is defined.
[item_aliases]

*__item_aliases.version__ (*char*)
The version of the dictionary in which the alias name is defined.
[item_aliases]

ITEM_DEFAULT

Attributes specifying the default value for a data item.
Category group(s): `ddl_group`
`item_group`
Category key(s): `_item_default.name`

`_item_default.value` (*any*)
The default value for the defined item if it is not specified explicitly. If a data value is not declared, the default is assumed to be the most likely or natural value.

`[item_default]`

ITEM_DEPENDENT

Attributes which identify other data items that must be specified for the defined data item to be valid.
Category key(s): `_item_dependent.name`
`_item_dependent.dependent_name`

`_item_dependent.dependent_name`
Data name of a dependent item.

ITEM_DESCRIPTION

This category holds the descriptions of each data item.
Mandatory category.
Category group(s): `ddl_group`
`item_group`
Category key(s): `_item_description.name`
`_item_description.description`

***`_item_description.description`** (*text*)
Text description of the defined data item.

`[item_description]`

ITEM_ENUMERATION

Attributes which specify the permitted enumeration of the items.
Category group(s): `ddl_group`
`item_group`
Category key(s): `_item_enumeration.name`
`_item_enumeration.value`

`_item_enumeration.detail` (*text*)
A description of a permissible value for the defined item.

`[item_enumeration]`

***`_item_enumeration.value`** (*any*)
A permissible value, character or number for the defined item.

`[item_enumeration]`

ITEM_EXAMPLES

Attributes for describing examples of applications of the data item.
Category group(s): `ddl_group`
`item_group`
Category key(s): `_item_examples.name`
`_item_examples.case`

`_item_examples.case` (*text*)
An example application of the defined data item.

`[item_examples]`

`_item_examples.detail` (*text*)
A description of an example specified in `_item_examples.case`.

`[item_examples]`

ITEM_LINKED

Attributes which describe how equivalent data items are linked within categories and across different categories.
Category group(s): `ddl_group`
`item_group`
Category key(s): `_item_linked.child_name`

`_item_linked.child_name`
Name of the child data item.

`_item_linked.parent_name`
Name of the parent data item.

ITEM_METHODS

Attributes specifying the association between data items and methods.
Category group(s): `ddl_group`
`item_group`
Category key(s): `_item_methods.method_id`
`_item_methods.name`

ITEM_RANGE

The range of permissible values of a data item. When multiple ranges are specified, they are interpreted sequentially using a logical OR. To specify that an item value may be equal to a boundary value, specify an item range where the maximum and mimimum values equal the boundary value.
Category group(s): `ddl_group`
`item_group`
Category key(s): `_item_range.name`
`_item_range.minimum`
`_item_range.maximum`

`_item_range.maximum` (*any*)
Maximum permissible value of a data item or the upper bound of a permissible range (maximum value $>$ data value).

`[item_range]`

`_item_range.minimum` (*any*)
Minimum permissible value of a data item or the lower bound of a permissible range (minimum value $<$ data value).

`[item_range]`

ITEM_RELATED

Attributes which specify recognized relationships between data items.
Category group(s): `ddl_group`
`item_group`
Category key(s): `_item_related.name`
`_item_related.related_name`
`_item_related.function_code`

*`_item_related.function_code` (*code*)

The code for the type of relationship between the item identified by `_item_related.name` and the defined item.

'alternate' indicates that the item identified in `_item_related.related_name` is an alternative expression in terms of its application and attributes to the item in this definition. 'alternate_exclusive' indicates that the item identified in `_item_related.related_name` is an alternative expression in terms of its application and attributes to the item in this definition. Only one of the alternative forms may be specified.

'convention' indicates that the item identified in `_item_related.related_name` differs from the defined item only in terms of a convention in its expression.

'conversion_constant' indicates that the item identified in `_item_related.related_name` differs from the defined item only by a known constant. 'conversion_arbitrary' indicates that the item identified in `_item_related.related_name` differs from the defined item only by an arbitrary constant.

'replaces' indicates that the defined item replaces the item identified in `_item_related.related_name`. 'replacedby' indicates that the defined item is replaced by the item identified in `_item_related.related_name`.

'associated_value' indicates that the item identified in `_item_related.related_name` is meaningful when associated with the defined item. 'associated_esd' indicates that the item identified in `_item_related.related_name` is the estimated standard deviation of the defined item. 'associated_error' indicates that the item identified in `_item_related.related_name` is the estimated error of the defined item.

The data value must be one of the following:

alternate	alternate form of the item
alternate_exclusive	mutually exclusive alternate form of the item
convention	depends on defined convention
conversion_constant	related by a known conversion factor
conversion_arbitrary	related by an arbitrary conversion factor
replaces	a replacement definition
replacedby	an obsolete definition
associated_value	a meaningful value when related to the item
associated_esd	an estimated standard deviation of the item
associated_error	an estimated error of the item

[item_related]

ITEM_STRUCTURE

This category holds the association between data items and named vector/matrix declarations.

Category group(s): **ddl_group**
item_group
Category key(s): **_item_structure.name**

*`_item_structure.organization` (*code*)

Identifies whether the structure is defined in column- or row-major order. Only the unique elements of symmetric matrices are specified.

The data value must be one of the following:

columnwise	column-major order
rowwise	row-major order

ITEM_STRUCTURE_LIST

This category holds a description for each structure type.

Category group(s): **ddl_group**
item_group
Category key(s): **_item_structure_list.code**
_item_structure_list.index

*`_item_structure_list.code` (*code*)

The name of the matrix/vector structure declaration.

The following item(s) have an equivalent role in their respective categories:

_item_structure.code. **[item_structure_list]**

*`_item_structure_list.dimension` (*int*)

Identifies the length of this row or column of the structure.

The permitted range is $[1, \infty)$. **[item_structure_list]**

*`_item_structure_list.index` (*int*)

Identifies the one-based index of a row or column of the structure.

The permitted range is $[1, \infty)$. **[item_structure_list]**

ITEM_SUB_CATEGORY

This category assigns data items to subcategories.

Category group(s): **sub_category_group**
item_group
Category key(s): **_item_sub_category.id**
_item_sub_category.name

ITEM_TYPE

Attributes for specifying the data-type code for each data item.

Category group(s): **ddl_group**
item_group
Category key(s): **_item_type.name**

ITEM_TYPE_CONDITIONS

Attributes for specifying additional conditions associated with the data type of the item.

Category group(s): **ddl_group**
item_group
compliance_group
Category key(s): **_item_type_conditions.name**

*`_item_type_conditions.code` (*code*)

Codes defining conditions on the `_item_type.code` specification. 'esd' permits a number string to contain an appended standard deviation number enclosed within parentheses, *e.g.* 4.37(5). 'seq' permits data to be declared as a sequence of values separated by a comma <,> or a colon <:>. The sequence $v_1, v_2, v_3 \ldots$ signals that v_1, v_2, v_3 *etc.* are alternative values for the data item. The sequence v_1:v_2 signals that v_1 and v_2 are the boundary values of a continuous range of values. This mechanism was used to specify permitted ranges of an item in previous DDL versions. Combinations of alternate and range sequences are permitted.

The data value must be one of the following:

none	no extra conditions apply to this data item
esd	numbers may have esd values appended within parentheses
seq	data may be declared as a comma- or colon-separated sequence

[item_type_conditions]

ITEM_TYPE_LIST

Attributes which define each type code.

Category group(s): **ddl_group**
item_group
Category key(s): **_item_type_list.code**

*`_item_type_list.code` (*code*)

The codes specifying the nature of the data value.

The following item(s) have an equivalent role in their respective categories:

_item_type.code. **[item_type_list]**

_item_type_list.construct (*text*)
When a data value can be defined as a pre-determined sequence of characters, optional characters or data names (for which the definition is also available), it is specified as a construction. The rules of construction conform to the the regular expression (REGEX) specifications detailed in IEEE (1991). Resolved data names for which **_item_type_list.construct** specifications exist are replaced by these constructions, otherwise the data-name string is not replaced.

Reference: IEEE (1991). *IEEE Standard for Information Technology – Portable Operating System Interface (POSIX) – Part 2: Shell and Utilities*, Vol. 1, IEEE Standard 1003.2-1992. New York: The Institute of Electrical Engineers.

Example: '_year-_month-_day' (typical construction for _date). **[item_type_list]**

_item_type_list.detail (*text*)
An optional description of the data type.

[item_type_list]

***_item_type_list.primitive_code** (*code*)
The codes specifying the primitive type of the data value.

The data value must be one of the following:

numb	numerically interpretable string
char	character or text string (case-sensitive)
uchar	character or text string (case-insensitive)
null	for dictionary purposes only

[item_type_list]

ITEM_UNITS

Specifies the physical units in which data items are expressed.
Category group(s): **ddl_group**
item_group
Category key(s): **_item_units.name**

ITEM_UNITS_CONVERSION

Conversion factors between the various units of measure defined in the ITEM_UNITS_LIST category.
Category group(s): **ddl_group**
item_group
Category key(s): **_item_units_conversion.from_code**
_item_units_conversion.to_code

***_item_units_conversion.factor** (*any*)
The arithmetic operation required to convert between the unit systems: ⟨to_code⟩ = ⟨from_code⟩⟨operator⟩⟨factor⟩.

[item_units_conversion]

_item_units_conversion.from_code
The unit system on which the conversion operation is applied to produce the unit system specified in **_item_units_conversion.to_code**: ⟨to_code⟩ = ⟨from_code⟩⟨operator⟩⟨factor⟩.

***_item_units_conversion.operator** (*code*)
The arithmetic operator required to convert between the unit systems: ⟨to_code⟩= ⟨from_code⟩⟨operator⟩⟨factor⟩.

The data value must be one of the following:

+	addition
-	subtraction
*	multiplication
/	division

[item_units_conversion]

_item_units_conversion.to_code
The unit system produced after an operation is applied to the unit system specified by **_item_units_conversion.from_code**: ⟨to_code⟩ = ⟨from_code⟩⟨operator⟩⟨factor⟩.

ITEM_UNITS_LIST

Attributes which describe the physical units of measurement in which data items may be expressed.
Category group(s): **ddl_group**
item_group
Category key(s): **_item_units_list.code**

***_item_units_list.code** (*code*)
The code specifying the name of the unit of measurement.

The following item(s) have an equivalent role in their respective categories:
_item_units.code,
_item_units_conversion.from_code,
_item_units_conversion.to_code. **[item_units_list]**

_item_units_list.detail (*text*)
A description of the unit of measurement.

[item_units_list]

METHOD_LIST

Attributes specifying the list of methods applicable to data items, subcategories and categories.
Category group(s): **ddl_group**
item_group
category_group
Category key(s): **_method_list.id**

***_method_list.code** (*code*)
A code that describes the function of the method.

Examples: 'calculation' (method to calculate the item), 'verification' (method to verify the data item), 'cast' (method to provide cast conversion), 'addition' (method to define item + item), 'division' (method to define item / item), 'multiplication' (method to define item × item), 'equivalence' (method to define item = item), 'other' (miscellaneous method). **[method_list]**

_method_list.detail (*text*)
Description of application method in **_method_list.id**.

[method_list]

***_method_list.id** (*idname*)
Unique identifier for each method listed.

The following item(s) have an equivalent role in their respective categories:
_item_methods.method_id,
_category_methods.method_id,
_sub_category_methods.method_id,
_datablock_methods.method_id. **[method_list]**

***_method_list.inline** (*text*)
In-line text of a method associated with the data item.

[method_list]

***_method_list.language** (*code*)
Language in which the method is expressed.

Examples: 'BNF', 'C', 'C++', 'FORTRAN', 'LISP', 'PASCAL', 'PERL', 'TCL', 'OTHER'.

[method_list]

SUB_CATEGORY

The purpose of a subcategory is to define an association between data items within a category and optionally to provide a method to validate the collection of items. For example, the subcategory named 'cartesian' might be applied to the data items for the coordinates *x*, *y* and *z*.
Category group(s): **ddl_group**
sub_category_group
Category key(s): **_sub_category.id**

***_sub_category.description** (*text*)
Description of the subcategory.

[sub_category]

***_sub_category.id** (*idname*)
The identity of the subcategory.

The following item(s) have an equivalent role in their respective categories:

_sub_category_examples.id,
_sub_category_methods.sub_category_id,
_item_sub_category.id. **[sub_category]**

***_sub_category_examples.case** (*text*)
An example involving items in this subcategory.

[sub_category_examples]

_sub_category_examples.detail (*text*)
A description of an example given in **_sub_category_examples.case**.

[sub_category_examples]

SUB_CATEGORY_EXAMPLES

Example applications and descriptions of data items in this subcategory.

Category group(s): **ddl_group**
sub_category_group
Category key(s): **_sub_category_examples.id**
_sub_category_examples.case

SUB_CATEGORY_METHODS

Attributes specifying the association between subcategories and methods.

Category group(s): **ddl_group**
sub_category_group
Category key(s): **_sub_category_methods.method_id**
_sub_category_methods.sub_category_id