# DDLm Dictionary

BY SYDNEY R. HALL, NICK SPADACCINI AND JAMES R. HESTER

## ATTRIBUTES

This category is parent of all other categories in the DDLm dictionary.

## ALIAS

The attributes used to specify the aliased names of definitions.

Category key(s): **`_alias.definition_id`**

**`_alias.definition_id`** (*Tag*)

Identifier tag of an aliased definition.

**`_alias.deprecation_date`** (*Date*)

Date that the aliased tag was deprecated as a definition tag.

**`_alias.dictionary_uri`** (*Uri*)

Dictionary URI in which the aliased definition belongs.

## CATEGORY

The attributes used to specify the properties of a 'category' of data items.

**`_category.key_id`** (*Tag*)

Tag of a single data item in a Loop category which is the generic key to access other items in the category. The value of this item must be unique in order to provide unambiguous access to a packet (row) in the table of values. This may be assumed to be a function of the datanames listed in **`_category_key.name`**.

## CATEGORY_KEY

The attributes used to specify (possibly multiple) keys for a given category.

Category key(s): **`_category_key.name`**

**`_category_key.name`** (*Tag*)

A minimal list of tag(s) that together constitute a compound key to access other items in a Loop category. In other words, the combined values of the datanames listed in this loop must be unique, so that unambiguous access to a packet (row) in the table of values is possible. The dataname associated with **`_category.key_id`** is only included in this loop if no other set of datanames can form a compound key.

## DEFINITION

The attributes for classifying dictionary definitions.

**`_definition.class`** (*Code*)

The nature and the function of a definition or definitions.

The data value must be one of the following:

| | |
|---|---|
| `Attribute` | Item used as an attribute in the definition of other data items in DDLm dictionaries. These items never appear in data instance files. |
| `Functions` | Category of items that are transient function definitions used only in dREL methods scripts. These items never appear in data instance files. |
| `Datum` | Item defined in a domain-specific dictionary. These items appear only in data instance files. |
| `Head` | Category of items that is the parent of all other categories in the dictionary. |
| `Loop` | Category of items that in a data file must reside in a loop-list with a key item defined. |
| `Set` | Category of items that form a set (but not a loopable list). These items may be referenced as a class of items in a dREL methods expression. |
| `Ref-loop` | A category containing one item that identifies the a category of items that is repeated in a sequence of save frames. The item, which is specifies as a as a Ref-table value (see type.container), is looped. This construction is for loop categories that contain child categories. If in the instance file, the child items have only one set of values, the Ref-loop item need not be used and child items need not be placed in a save frame. |

**`_definition.id`** (*Code*)

Identifier name of the Item or Category definition contained within a save frame.

**`_definition.scope`** (*Code*)

The extent to which a definition affects other definitions.

The data value must be one of the following:

| | |
|---|---|
| `Dictionary` | applies to all defined items in the dictionary |
| `Category` | applies to all defined items in the category |
| `Item` | applies to a single item definition |

**`_definition.update`** (*Date*)

The date that a definition was last changed.

**`_definition.xref_code`** (*Code*)

Code identifying the equivalent definition in the dictionary referenced by the DICTIONARY_XREF attributes.

## DEFINITION_REPLACED

Attributes used to describe deprecated and replaced definitions.

Category key(s): **`_definition_replaced.id`**

**`_definition_replaced.by`** (*Tag*)

Name of the data item that should be used instead of the defined data item. The defined data item is deprecated and should not be used.

**`_definition_replaced.id`** (*Code*)

An opaque identifier for the replacement.

Affiliations: SYDNEY R. HALL, Biomolecular, Biomedical & Chemical Sciences, Faculty Life & Physical Sciences, M310 University of Western Australia, Nedlands 6009, Australia; NICK SPADACCINI, The University of Western Australia, Crawley, 6009, Australia; JAMES R. HESTER, Australian Nuclear Science and Technology Organisation, New Illawarra Road, Lucas Heights, NSW 2234, Australia.

## DESCRIPTION

The attributes of descriptive (non-machine parsable) parts of definitions.

### `_description.common` (*Text*)

Commonly-used identifying name for the item.

### `_description.key_words` (*Text*)

List of key-words categorising the item.

### `_description.text` (*Text*)

The text description of the defined item.

## DESCRIPTION_EXAMPLE

The attributes of descriptive (non-machine parsable) examples of values of the defined items.

Category key(s): `_description_example.case`

### `_description_example.case` (*Implied*)

An example case of the defined item. Instances of this data item inherit the container, content and purpose type constraints of the defining item.

### `_description_example.detail` (*Text*)

A description of an example case for the defined item.

## DICTIONARY

Attributes for identifying and registering the dictionary. The items in this category are *not* used as attributes of INDIVIDUAL data items.

### `_dictionary.class` (*Code*)

The nature, or field of interest, of data items defined in the dictionary.

The data value must be one of the following:

| | |
|---|---|
| `Reference` | DDLm reference attribute definitions |
| `Instance` | domain-specific data instance definitions |
| `Template` | domain-specific attribute/enumeration templates |
| `Function` | domain-specific method function scripts |

### `_dictionary.date` (*Date*)

The date that the last dictionary revision took place.

### `_dictionary.ddl_conformance` (*Version*)

The version number of the DDL dictionary that this dictionary conforms to.

### `_dictionary.formalism` (*Text*)

The definitions contained in this dictionary are associated with the value of this attribute. Datanames may only be redefined if the value of this attribute is also changed, and any such redefinitions must include the original behaviour as a particular case.

### `_dictionary.namespace` (*Code*)

The namespace code that may be prefixed (with a trailing colon ':') to an item tag defined in the defining dictionary when used in particular applications. Because tags must be unique, namespace codes are unlikely to be used in data files.

### `_dictionary.title` (*Code*)

The common title of the dictionary. Will usually match the name attached to the `data_` statement of the dictionary file.

### `_dictionary.uri` (*Uri*)

An absolute uniform resource identifier (URI) for this dictionary.

### `_dictionary.version` (*Version*)

A unique version identifier for the dictionary.

## DICTIONARY_AUDIT

Attributes for identifying and registering the dictionary. The items in this category are *not* used as attributes of individual data items.

Category key(s): `_dictionary_audit.version`

### `_dictionary_audit.date` (*Date*)

The date of each dictionary revision.

### `_dictionary_audit.revision` (*Text*)

A description of the revision applied for the `_dictionary_audit.version`.

### `_dictionary_audit.version` (*Version*)

A unique version identifier for each revision of the dictionary.

## DICTIONARY_VALID

Data items which are used to specify the contents of definitions in the dictionary in terms of the `_definition.scope` and the required and prohibited attributes.

Category key(s): `_dictionary_valid.application`

### `_dictionary_valid.application` (*Code*[2])

Provides the information identifying the definition scope ( from the `_definition.scope` enumeration list) and the validity options (from the `_dictionary_valid.option` enumeration list), as a two element list. This list signals the validity of applying the attributes given in `_dictionary_valid.attributes`.

### `_dictionary_valid.attributes` (*Code*[])

A list of the attribute names and categories that are assessed for application in the item, category and dictionary definitions.

### `_dictionary_valid.option` (*Code*)

Option codes for applicability of attributes in definitions.

The data value must be one of the following:

| | |
|---|---|
| `Mandatory` | attribute must be present in definition frame |
| `Recommended` | attribute is usually in definition frame |
| `Prohibited` | attribute must not be used in definition frame |

### `_dictionary_valid.scope` (*Code*)

The scope to which the specified restriction on usable attributes applies.

The data value must be one of the following:

| | |
|---|---|
| `Dictionary` | restriction applies to dictionary definition data frame |
| `Category` | restriction applies to a category definition save frame |
| `Item` | restriction applies to an item definition save frame |

## DICTIONARY_XREF

Data items which are used to cross reference other dictionaries that have defined the same data items. Data items in this category are *not* used as attributes of individual data items.

Category key(s): `_dictionary_xref.code`

### `_dictionary_xref.code` (*Code*)

A code identifying the cross-referenced dictionary.

### `_dictionary_xref.date` (*Date*)

Date of the cross-referenced dictionary.

**_dictionary_xref.format** (*Text*)
Format of the cross referenced dictionary.

**_dictionary_xref.name** (*Text*)
The name and description of the cross-referenced dictionary.

**_dictionary_xref.uri** (*Uri*)
The source URI of the cross referenced dictionary data.

## ENUMERATION

The attributes for restricting the values of defined data items.

**_enumeration.def_index_id** (*Tag*)
Specifies the data name with a value used as an index to the DEFAULT enumeration list (in category ENUMERATION_DEFAULT) in order to select the default enumeration value for the defined item. The value of the identified data item must match one of the **_enumeration_default.index** values.

**_enumeration.default** (*Implied*)
The default value for the defined item if it is not specified explicitly.

**_enumeration.mandatory** (*Code*)
Yes or No flag on whether the enumerate states specified for an item in the current definition (in which item appears) *must* be used on instantiation.

The data value must be one of the following:

| | |
|---|---|
| Yes | Use of state is mandatory |
| No | Use of state is unnecessary |

**_enumeration.range** (*Range*)
The inclusive range of values 'from:to' allowed for the defined item. If items have associated SU, the reported value may fall outside these limits.

## ENUMERATION_DEFAULT

Loop of pre-determined default enumeration values indexed to a data item by the item **_enumeration.def_index_id**.
Category key(s): **_enumeration_default.index**

**_enumeration_default.index** (*Code*)
Index key in the list default values referenced to by the value of **_enumeration.def_index_id** .

**_enumeration_default.value** (*Implied*)
Default enumeration value in the list referenced by the value of **_enumeration.def_index_id**. The reference index key is given by the value of **_enumeration_default.index** value.

## ENUMERATION_SET

Attributes of data items which are used to define a set of unique pre-determined values.
Category key(s): **_enumeration_set.state**

**_enumeration_set.detail** (*Text*)
The meaning of the code (identified by **_enumeration_set.state**) in terms of the value of the quantity it describes.

**_enumeration_set.state** (*Text*)
Permitted value state for the defined item.

**_enumeration_set.xref_code** (*Code*)
Identity of the equivalent item in the dictionary referenced by the DICTIONARY_XREF attributes.

**_enumeration_set.xref_dictionary** (*Code*)
Code identifying the dictionary in the DICTIONARY_XREF list.

## IMPORT

Used to import the values of specific attributes from other dictionary definitions within and without the current dictionary.

**_import.get** (*ByReference*)
A list of tables of attributes defined individually in the category IMPORT_DETAILS, used to import definitions from other dictionaries.

## IMPORT_DETAILS

Items in IMPORT_DETAILS describe individual attributes of an import operation.
Category key(s): **_import_details.order**

**_import_details.file_id** (*Uri*)
A URI reference as per RFC3986 giving the location of the source dictionary. When a relative URI is used, the base URI for the URI reference is the **_dictionary.uri** of the importing dictionary.

**_import_details.file_version** (*Version*)
The required version number for **_dictionary.version** of the imported dictionary. Dictionaries with the same major version number are compatible. If absent or null, any version is permitted.

**_import_details.frame_id** (*Code*)
The framecode of the definition frame to be imported.

**_import_details.if_dupl** (*Code*)
Code identifying the action taken if the requested definition block already exists within the importing dictionary.

The data value must be one of the following:

| | |
|---|---|
| Ignore | ignore imported definitions if id conflict |
| Replace | replace existing with imported definitions |
| Exit | issue error exception and exit |

**_import_details.if_miss** (*Code*)
Code identifying the action taken if the requested definition block is missing from the source dictionary.

The data value must be one of the following:

| | |
|---|---|
| Ignore | ignore import |
| Exit | issue error exception and exit |

**_import_details.mode** (*Code*)
Code identifying how the definition referenced by **_import_details.frame_id** is to be imported. 'Full' imports the entire definition together with any child definitions (in the case of categories) found in the target dictionary. The importing definition becomes the parent of the imported definition. As such, the 'Full' mode must only be used in category definitions. As a special case, a 'Head' category importing a 'Head' category is equivalent to importing all children of the imported 'Head' category as children of the importing 'Head' category. A 'Head' category can only be imported in 'Full' mode and only by another 'Head' category. 'Contents' imports only the attributes found in the imported definition.

The data value must be one of the following:

| | |
|---|---|
| Full | import requested definition together with any child definitions |
| Contents | import contents of requested definition |

### _import_details.order (*Integer*)

The order in which the import described by the referenced row should be executed.

### _import_details.single (*Text*)

A Table mapping attributes defined individually in category IMPORT to their values; used to import definitions from other dictionaries.

### _import_details.single_index (*Code*)

One of the indices permitted in the entries of values of attribute **_import_details.single**.

The data value must be one of the following:

| | |
|---|---|
| file | URI of source dictionary |
| version | version of source dictionary |
| save | save framecode of source definition |
| mode | mode for including save frames |
| dupl | option for duplicate entries |
| miss | option for missing duplicate entries |

## LOOP

Attributes for looped lists.

### _loop.level (*Integer*)

Specifies the level of the loop structure in which a defined item must reside if it used in a looped list.

## METHOD

Methods used for evaluating, validating and defining items.

Category key(s): **_method.purpose**

### _method.expression (*Text*)

The method expression for the defined item.

### _method.purpose (*Code*)

The purpose and scope of the method expression.

The data value must be one of the following:

| | |
|---|---|
| Evaluation | method evaluates an item from related item values |
| Definition | method generates attribute value(s) in the definition |
| Validation | method compares an evaluation with existing item value |

## NAME

Attributes for identifying items and item categories.

### _name.category_id (*Name*)

The name of the category in which a category or item resides. For Head categories this is the **_dictionary.title** given in the enclosing data block.

### _name.linked_item_id (*Tag*)

Dataname of an equivalent item which has a common set of values, or, in the definition of a type SU item is the name of the associated Measurement item to which the standard uncertainty applies.

### _name.object_id (*Name*)

The object name of a category or name unique within the category or family of categories.

## TYPE

Attributes which specify the ‘typing’ of data items.

### _type.container (*Code*)

The CONTAINER type of the defined data item value. ‘Implied’ may only be used in an attribute dictionary for attributes whose types are linked to the values of other attributes.

The data value must be one of the following:

| | |
|---|---|
| Single | single value |
| Multiple | values as List or by boolean ,\|&\|* or range : ops |
| List | ordered set of values. Elements need not be of same contents type. |
| Array | ordered set of numerical values. Operations across arrays are equivalent to operations across elements of the Array. |
| Matrix | ordered set of numerical values for a tensor. Tensor operations such as dot and cross products, are valid cross matrix objects. A matrix with a single dimension is interpreted as a row or column vector as required. |
| Table | An unordered set of id:value elements |
| Implied | (For use in the attribute dictionary only). Determined by the values of other attributes. |

### _type.contents (*Code*)

Syntax of the value elements within the container type. This may be a single enumerated code, or, in the case of a list, a comma-delimited sequence of codes, or, if there are alternate types, a boolean-linked (or range) sequence of codes. The typing of elements is determined by the replication of the minimum set of states declared. Where the definition is of a ‘Table’ container this attribute describes the construction of the value elements within those (Table) values. The CIF2 characterset referenced below consists of the following Unicode code points:
[U+0009], [U+000A], [U+000D], [U+0020–U+007E], [U+00A0–U+D7FF], [U+E000–U+FDCF], [U+FDF0–U+FFFD], [U+10000–U+1FFFD], [U+20000–U+2FFFD], [U+30000–U+3FFFD], [U+40000–U+4FFFD], [U+50000–U+5FFFD], [U+60000–U+6FFFD], [U+70000–U+7FFFD], [U+80000–U+8FFFD], [U+90000–U+9FFFD], [U+A0000–U+AFFFD], [U+B0000–U+BFFFD], [U+C0000–U+CFFFD], [U+D0000–U+DFFFD], [U+E0000–U+EFFFD], [U+F0000–U+FFFFD], [U+100000–U+10FFFD]
Two ‘case insensitive’ strings are considered identical when they match under the Unicode canonical caseless matching algorithm. In all cases, ‘whitespace’ refers to ASCII whitespace only, that is [U+0009],[U+000A],[U+000D] and [U+0020]. Note that descriptions of text syntax are relevant only to those formats that encode data values as text.

The data value must be one of the following:

| | |
|---|---|
| Text | case-sensitive sequence of CIF2 characters |
| Code | case-insensitive sequence of CIF2 characters containing no ASCII whitespace. |
| Name | case-insensitive sequence of ASCII alpha-numeric characters or underscore |
| Tag | case-insensitive CIF2 character sequence with leading underscore and no ASCII whitespace |
| Uri | A Uniform Resource Identifier per RFC 3986 |
| Date | ISO standard date format ⟨*yyyy*⟩-⟨*mm*⟩-⟨*dd*⟩ |
| DateTime | A timestamp. Text formats must use date-time or full-date productions of RFC3339 ABNF |
| Version | version digit string of the form ⟨*major*⟩.⟨*version*⟩.⟨*update*⟩ |
| Dimension | Size of an Array/Matrix/List expressed as a text string. The text string itself consists of zero or more non-negative integers separated by commas placed within bounding square brackets. Empty square brackets represent a list of unknown size |
| Range | inclusive range of numerical values min:max |
| Count | unsigned integer number (deprecated) |
| Index | unsigned non-zero integer number (deprecated) |
| Integer | positive or negative integer number |
| Real | floating-point real number |
| Imag | floating-point imaginary number |
| Complex | a complex number |

`Symop` a string composed of an integer optionally followed by an underscore or space and three or more digits

`Implied` implied by the context of the attribute

`ByReference` The contents have the same form as those of the attribute referenced by **`_type.contents_referenced_id`**.

Examples: '`Integer`' (content is a single or multiple integer(s)), '`Real,Integer`' (List elements of a real number and an integer), '`List(Real,Code)`' (List of Lists of a real number and a code), '`Text|Real`' (content is either text OR a real number)

### _type.contents_referenced_id (*Tag*)

The value of the **`_definition.id`** attribute of an attribute definition whose type is to be used also as the type of this item. Meaningful only when this item's **`_type.contents`** attribute has value 'ByReference'.

### _type.dimension (*Dimension*)

The dimensions of a list or matrix of elements expressed as a text string. A Matrix with a single dimension is interpreted as a vector.

Examples: '`[3,3]`' (3x3 matrix of elements), '`[6]`' (list of 6 elements), '`[]`' (unknown number of list elements)

### _type.indices (*Code*)

Used to specify the syntax construction of indices of the entries in the defined object when the defined object has 'Table' as its **`_type.container`** attribute. Values are a subset of the codes and constructions defined for attribute **`_type.contents`**, accounting for the fact that syntactically, indices are always case-sensitive quoted strings. Meaningful only when the defined item has **`_type.container`** 'Table'. See the definition for **`_type.contents`** for the characterset definition.

The data value must be one of the following:

`Text` A case-sensitive sequence of CIF2 characters

`Code` case-insensitive sequence of CIF2 characters containing no ASCII whitespace.

`Date` ISO standard date format ⟨*yyyy*⟩-⟨*mm*⟩-⟨*dd*⟩

`Uri` a Uniform Resource Identifier string, per RFC 3986

`Version` version digit string of the form ⟨*major*⟩.⟨*version*⟩.⟨*update*⟩

`ByReference` Indices have the same form as the contents of the attribute identified by **`_type.indices_referenced_id`**

### _type.indices_referenced_id (*Tag*)

The **`_definition.id`** attribute of a definition whose type describes the form and construction of the indices of entries in values of the present item. Meaningful only when the defined item's **`_type.container`** attribute has value 'Table', and its **`_type.indices`** attribute has value 'ByReference'.

### _type.purpose (*Code*)

The primary purpose or function the defined data item serves in a dictionary or a specific data instance.

The data value must be one of the following:

`Import` **Applied only in the DDLm Reference Dictionary** Used to type the SPECIAL attribute '**`_import.get`**' that is present in dictionaries to instigate the importation of external dictionary definitions.

`Method` **Applied only in the DDLm Reference Dictionary** Used to type the attribute "**`_method.expression`**" that is present in dictionary definitions to provide the text method expressing the defined item in terms of other defined items.

`Audit` **Applied only in the DDLm Reference Dictionary** Used to type attributes employed to record the audit definition information (creation date, update version and cross reference codes) of items, categories and files.

`Identify` **Applied only in the DDLm Reference Dictionary** Used to type attributes that identify an item tag (or part thereof), save frame or the URI of an external file.

`Extend` **Used to extend the DDLm Reference Dictionary** Used in a definition, residing in the 'extensions' save frame of a domain dictionary, to specify a new enumeration state using an Evaluation method.

`Describe` Used to type items with values that are descriptive text intended for human interpretation.

`Encode` Used to type items with values that are text or codes that are formatted to be machine parsable.

`State` Used to type items with values that are restricted to codes present in their '**`_enumeration_set.state`**' lists.

`Key` Used to type an item with a value that is unique within the looped list of these items, and does not contain encoded information.

`Link` Used to type an item that acts as a foreign key between two categories. The definition of the item must additionally contain the attribute '**`_name.linked_item_id`**' specifying the data name of the item with unique values in the linked category. The values of the defined item are drawn from the set of values in the referenced item. Cross referencing items from the same category is allowed.

`Composite` Used to type items with value strings composed of separate parts. These will usually need to be separated and parsed for complete interpretation and application.

`Number` Used to type items that are numerical and exact (*i.e.* no standard uncertainty value).

`Measurand` Used to type an item with a numerically estimated value that has been recorded by measurement or derivation. This value must be accompanied by its standard uncertainty (SU) value, expressed either as:
(1) appended integers, in parentheses (), at the precision of the trailing digits, or
(2) a separately defined item with the same name as the measurand item but with an additional suffix '**`_su`**'.

`SU` Used to type an item with a numerical value that is the standard uncertainty of another dataname. The definition of an SU item must include the attribute '**`_name.linked_item_id`**' which explicitly identifies the associated measurand item. SU values must be non-negative.

`Internal` Used to type items that serve only internal purposes of the dictionary in which they appear. The particular purpose served is not defined by this state.

### _type.source (*Code*)

The origin or source of the defined data item, indicating by what recording process it has been added to the domain instance.

The data value must be one of the following:

`Recorded` A value (numerical or otherwise) recorded by observation or measurement during the experimental collection of data. This item is **primitive**.

`Assigned` A value (numerical or otherwise) assigned as part of the data collection, analysis or modelling required for a specific domain instance. These assignments often represent a decision made that determines the course of the experiment (and therefore may be deemed **primitive**) or a particular choice in the way the data was analysed (and therefore may be considered **not primitive**).

`Related` A value or tag used in the construction of looped lists of data. Typically identifying an item whose unique value is the reference key for a loop category and/or an item which has values in common with those of another loop category and is considered a Link between these lists.

`Derived` A quantity derived from other data items within the domain instance. This item is **not primitive**.

## UNITS

The attributes for specifying units of measure.

### _units.code (*Code*)

A code which identifies the units of measurement.