

**APPENDIX A**  
**(VERSION 2 AUGUST 2013)**

**GLOSSARY OF DDLM ATTRIBUTES**

**THE DICTIONARY-DICTIONARY**

The *DDLm* attributes used to construct the definitions of data items in domain-specific dictionaries are themselves defined in a similarly constructed dictionary, known as the *dictionary-dictionary* or simply, the *DDLm dictionary*. In this document, the attribute items are used to define their own function and properties.

We shall first look at the overall contents of the dictionary-dictionary, hereafter referred to as the DDLM dictionary, before describing the function of the individual attribute items.

The list below (generated from the version 3.11.03 DDLM dictionary with the program DICtree) shows the attribute item names grouped in categories (shown with capitalized header names) in two columns. The left-hand column shows how each data name will appear in an instance application, and the right-hand column shows the data object names in their internal DDLm hierarchical representation.

Instance Application -----	Categories & Objects (showing hierarchy) -----
<b>ATTRIBUTES</b>	
loop_	ALIAS
_alias.definition_id	definition_id
_alias.dictionary_uri	dictionary_uri
	CATEGORY
_category.key_id	key_id
	DEFINITION
_definition.class	class
_definition.id	id
_definition.scope	scope
_definition.update	update
_definition.xref_code	xref_code
	DESCRIPTION
_description.key_words	key_words
_description.common	common
_description.text	text
loop_	EXAMPLE
_description_example.case	case
_description_example.detail	detail
	DICTIONARY
_dictionary.class	class
_dictionary.date	date
_dictionary.ddl_conformance	ddl_conformance
_dictionary.namespace	namespace
_dictionary.title	title
_dictionary.uri	uri
_dictionary.version	version
loop_	AUDIT
_dictionary_audit.date	date
_dictionary_audit.revision	revision
_dictionary_audit.version	version

loop_	VALID
_dictionary_valid.attributes	attributes
_dictionary_valid.option	option
_dictionary_valid.application	application
loop_	XREF
_dictionary_xref.code	code
_dictionary_xref.date	date
_dictionary_xref.format	format
_dictionary_xref.name	name
_dictionary_xref.uri	uri
	ENUMERATION
_enumeration.default	default
_enumeration.def_index_id	def_index_id
_enumeration.range	range
_enumeration.mandatory	mandatory
loop_	DEFAULT
_enumeration_default.index	index
_enumeration_default.value	value
loop_	SET
_enumeration_set.state	state
_enumeration_set.table_id	table_id
_enumeration_set.detail	detail
_enumeration_set.xref_code	xref_code
_enumeration_set.xref_dictionary	xref_dictionary
	IMPORT
_import.file_id	file_id
_import.frame_id	frame_id
_import.mode	mode
_import.if_dupl	if_dupl
_import.if_miss	if_miss
_import.get	get
	LOOP
_loop.level	level
loop_	METHOD
_method.purpose	purpose
_method.expression	expression
	NAME
_name.object_id	object_id
_name.category_id	category_id
_type.linked_item_id	linked_item_id
	TYPE
_type.container	container
_type.contents	contents
_type.purpose	purpose
_type.source	source
_type.dimension	dimension
	UNITS
_units.code	code

## USE OF ATTRIBUTES IN DEFINITIONS

The basic organizational framework for the attributes within a definition is consistent across all definitions and domain dictionaries. However, the attributes used in the definition of a data item or category will depend on the definition type (dictionary, category or item), and on the nature of the data

(text, numerical, etc.). An attribute may be essential, optional or unnecessary to the correct and precise definition of an item, a category or a dictionary.

The rules on using different attributes according to the *definition type* are straightforward and are explicitly specified in the *DDL*M dictionary using the attributes `_dictionary_valid.application` and `_dictionary_valid.attributes`. The *DDL*M dictionary is the only dictionary in which these two attributes are specified, and this invocation is shown below.

<code>loop_</code>	<code>_dictionary_valid.application</code>	<code>_dictionary_valid.attributes</code>
["Dictionary", "Mandatory"]		["_dictionary.title", "_dictionary.class", "_dictionary.version", "_dictionary.date", "_dictionary.uri", "_dictionary.ddl_conformance", "_dictionary.namespace"]
["Dictionary", "Recommended"]		["_description.text", "_dictionary_audit.version", "_dictionary_audit.date", "_dictionary_audit.revision"]
["Dictionary", "Prohibited"]		["ALIAS", "CATEGORY", "DEFINITION", "ENUMERATION", "LOOP", "METHOD", "NAME", "TYPE", "UNITS"]
["Category", "Mandatory"]		["_definition.id", "_definition.scope", "_definition.class", "_name.category_id", "_name.object_id"]
["Category", "Recommended"]		["_category.key_id", "_description.text"]
["Category", "Prohibited"]		["ALIAS", "DICTIONARY", "ENUMERATION", "IMPORT", "LOOP", "TYPE", "UNITS"]
["Item", "Mandatory"]		["_definition.id", "_definition.update", "_name.object_id", "_name.category_id", "_type.purpose", "_type.source", "_type.container", "_type.contents"]
["Item", "Recommended"]		["_definition.scope", "_definition.class", "_description.text", "_description.common"]
["Item", "Prohibited"]		["CATEGORY",

"DICTIONARY"]

## THE DESCRIPTION OF DDLM ATTRIBUTES

For the sake of brevity, the descriptions of the DDLm attribute items that follow, in the order they appear in a DDLm dictionary, will exclude certain definition information. That is, only the information that identifies the unique role of each attribute item will be shown.

### • ALIAS Attributes

The ALIAS attributes identify *identically equivalent* tags that may be aliased (i.e. substituted) for the defined tag. These attributes are included when equivalent items exist in this or another dictionary.

#### CATEGORY ALIAS

<code>_definition.class</code>	Loop
<code>_name.category_id</code>	ATTRIBUTES
<code>_name.object_id</code>	ALIAS
<code>_category.key_id</code>	'_alias.definition_id'

#### `_alias.definition_id`

Specifies the data names of items that are identically equivalent to the item in the current definition.

<code>_type.purpose</code>	Key
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Tag

#### `_alias.dictionary_uri`

Specifies the universal resource identifier of the dictionary containing the definition of items aliased to the item in the current definition.

<code>_type.purpose</code>	Identify
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Uri

### • CATEGORY Attributes

The CATEGORY attributes specify the group properties of a related set of items.

#### CATEGORY CATEGORY

<code>_definition.class</code>	Set
<code>_name.category_id</code>	ATTRIBUTES
<code>_name.object_id</code>	CATEGORY

#### `_category.key_id`

Specifies the data name of the item whose value is the key to packets of items in a Loop category.

<code>_type.purpose</code>	Identify
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Tag

## • DEFINITION Attributes

The `DEFINITION` attributes identify the nature and purpose of definition frames in a dictionary.

### CATEGORY DEFINITION

<code>_definition.class</code>	Set
<code>_name.category_id</code>	ATTRIBUTES
<code>_name.object_id</code>	DEFINITION

### `_definition.class`

Specifies the class or purpose of the dictionary, category or item being defined. The allowed definition classes are listed below.

<code>_type.purpose</code>	State
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Code
<code>_loop_</code>	
<code>_enumeration_set.state</code>	
<code>_enumeration_set.detail</code>	
<b>Attribute</b>	
<code>;</code>	Item used as an attribute in the definition of other data items in DDLm dictionaries. These items never appear in data instance files.
<code>;</code>	
<b>Functions</b>	
<code>;</code>	Category of items that are transient function definitions used only in dREL methods scripts. These items never appear in data instance files.
<code>;</code>	
<b>Datum</b>	
<code>;</code>	Item defined in a domain-specific dictionary. These items appear only in data instance files.
<code>;</code>	
<b>Head</b>	
<code>;</code>	Category of items that is the parent of all other categories in the dictionary.
<code>;</code>	
<b>Loop</b>	
<code>;</code>	Category of items that in a data file must reside in a loop-list with a key item defined.
<code>;</code>	
<b>Set</b>	
<code>;</code>	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.
<code>;</code>	
<b>Ref-loop</b>	
<code>;</code>	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 <code>type.container</code> ), 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.
<code>;</code>	

<code>_enumeration.default</code>	Datum
-----------------------------------	-------

### **`_definition.id`**

Specifies the data name of the *item or category* being defined within the current definition frame.

<code>_type.purpose</code>	Identify
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Tag

### **`_definition.scope`**

Specifies the *scope* of item being defined in terms of its inheritance. The allowed definition scopes are shown.

<code>_type.purpose</code>	State
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Code
<code>_loop_</code>	
<code>_enumeration_set.state</code>	
<code>_enumeration_set.detail</code>	
<b>Dictionary</b>	"applies to all defined items in the dictionary"
<b>Category</b>	"applies to all defined items in the category"
<b>Item</b>	"applies to a single item definition"
<code>_enumeration.default</code>	Item

### **`_definition.update`**

Specifies the calendar date (format “yyyy-mm-dd”) that the item definition was last updated.

<code>_type.purpose</code>	Audit
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Date

### **`_definition.xref_code`**

Specifies a code that identifies the same item defined in another dictionary identified by the `DICTIONARY_XREF` category of attributes.

<code>_type.purpose</code>	Identify
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Code

## **• DESCRIPTION Attributes**

The `DESCRIPTION` attributes provide various text descriptions of the defined data item.

### **CATEGORY DESCRIPTION**

<code>_definition.class</code>	Set
<code>_name.category_id</code>	ATTRIBUTES
<code>_name.object_id</code>	DESCRIPTION

### **`_description.key_words`**

Specifies a list of comma-delimited word sequences that are "key words" identifying an item for thematic searches.

<code>_type.purpose</code>	Describe
<code>_type.source</code>	Assigned
<code>_type.container</code>	List
<code>_type.contents</code>	Code

### **`_description.common`**

Specifies the commonly used name of the defined item.

<code>_type.purpose</code>	Describe
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Text

### **`_description.text`**

Specifies the text describing of the defined item.

<code>_type.purpose</code>	Describe
<code>_type.container</code>	Single
<code>_type.contents</code>	Text

## • **DESCRIPTION\_EXAMPLE Attributes**

The `DESCRIPTION_EXAMPLE` attributes provide descriptive example values of the defined item. These values are not machine-interpretable.

### **CATEGORY DESCRIPTION\_EXAMPLE**

<code>_definition.class</code>	Loop
<code>_name.category_id</code>	DESCRIPTION
<code>_name.object_id</code>	EXAMPLE
<code>_category.key_id</code>	'_description_example.case'

### **`_description_example.case`**

Specifies an example value for the defined item.

<code>_type.purpose</code>	Key
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Text

### **`_description_example.detail`**

Specifies the text details of an example value for the defined item.

<code>_type.purpose</code>	Describe
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Text

## • **DICTIONARY Attributes**

The `DICTIONARY` attributes describe aspects of the dictionary as a whole. These attributes are specified within the dictionary block but not within a definition save frame.

## CATEGORY DICTIONARY

<code>_definition.class</code>	Set
<code>_name.category_id</code>	ATTRIBUTES
<code>_name.object_id</code>	DICTIONARY

### **`_dictionary.class`**

Specifies the nature or purpose of the items defined in the dictionary.

<code>_type.purpose</code>	State
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Code
<code>_loop_</code>	
<code>_enumeration_set.state</code>	
<code>_enumeration_set.detail</code>	
<b>Reference</b>	'DDIm reference attribute definitions'
<b>Instance</b>	'domain-specific data instance definitions'
<b>Template</b>	'domain-specific attribute/enumeration templates'
<b>Function</b>	'domain-specific method function scripts'
<code>_enumeration.default</code>	Instance

### **`_dictionary.date`**

Specifies the calendar date (format “yyyy-mm-dd”) that the dictionary was last updated.

<code>_type.purpose</code>	Audit
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Date

### **`_dictionary.ddl_conformance`**

Specifies the version code (nn.mm.ii) for the DDL dictionary to which all definitions in the current dictionary conform.

<code>_type.purpose</code>	Audit
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Version

### **`_dictionary.namespace`**

Specifies a unique name for the dictionary that may be prefixed to an item data name (defined within the specific dictionary) with a separating colon character ":" when used in dictionary applications. Because tags must be unique, dictionary namespace prefixes are unlikely to be used in data files.

<code>_type.purpose</code>	Identify
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Code

### **`_dictionary.uri`**

Specifies the URI location and filename of the current dictionary.

<code>_type.purpose</code>	Identify
<code>_type.container</code>	Single
<code>_type.contents</code>	Uri



### **\_dictionary.title**

Specifies the common title for the current dictionary.

<code>_type.purpose</code>	Identify
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Code

### **\_dictionary.version**

Specifies the version code (nn.mm.ii) of the dictionary. This code must match a value for `_dictionary_audit.version` in the dictionary audit information.

<code>_type.purpose</code>	Audit
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Version

## • **DICTIONARY\_AUDIT Attributes**

The `DICTIONARY_AUDIT` attributes describe the status and the origins of a dictionary.

### **CATEGORY DICTIONARY\_AUDIT**

<code>_definition.class</code>	Loop
<code>_name.category_id</code>	DICTIONARY
<code>_name.object_id</code>	AUDIT
<code>_category.key_id</code>	'_dictionary_audit.version'

### **\_dictionary\_audit.date**

Specifies the calendar date (format “yyyy-mm-dd”) of the last revision of the dictionary.

<code>_type.purpose</code>	Audit
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Date

### **\_dictionary\_audit.revision**

Specifies the description of the revision applied.

<code>_type.purpose</code>	Describe
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Text

### **\_dictionary\_audit.version**

Specifies the code (nn.mm.ii) identifying the version of a dictionary (see `_dictionary.version`) associated with a revision.

<code>_type.purpose</code>	Key
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Version

## • **DICTIONARY\_VALID Attributes**

The `DICTIONARY_VALID` attributes identify when attributes are used in the different definition scopes. That is, whether specific attributes are mandatory or prohibited in the dictionary, category or item definitions. *The `DICTIONARY_VALID` attributes are only used in the DDLM dictionary.*

### CATEGORY `DICTIONARY_VALID`

<code>_definition.class</code>	Loop
<code>_name.category_id</code>	DICTIONARY
<code>_name.object_id</code>	VALID
<code>_category.key_id</code>	'_dictionary_valid.application'

### `_dictionary_valid.attributes`

Specifies a list of the names of attributes that are mandatory, prohibited or encouraged.

<code>_type.purpose</code>	Audit
<code>_type.source</code>	Assigned
<code>_type.container</code>	List
<code>_type.contents</code>	Text

### `_dictionary_valid.option`

Specifies the options for using the attributes in the `_dictionary_valid.attributes` list.

<code>_type.purpose</code>	Audit
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Code
<code>_loop_</code>	
<code>_enumeration_set.state</code>	
<code>_enumeration_set.detail</code>	
<b>Mandatory</b>	'attribute must be present in definition frame'
<b>Recommended</b>	'attribute is usually in definition frame'
<b>Prohibited</b>	'attribute must not be used in definition frame'

### `_dictionary_valid.application`

Specifies the options for using the attributes in the `_dictionary_valid.attributes` list.

<code>_type.purpose</code>	Key
<code>_type.source</code>	Assigned
<code>_type.container</code>	List
<code>_type.contents</code>	Code
<code>_type.dimension</code>	[2]
<code>_method.purpose</code>	
<code>_method.expression</code>	Definition
<code>;</code>	
<code>  _dictionary_valid.application</code>	<code>= [_definition.scope, _dictionary_valid.option]</code>
<code>;</code>	

## • `DICTIONARY_XREF` Attributes

The `DICTIONARY_XREF` attributes identify external dictionaries to which items in the current dictionary are cross-referenced using the `_definition.xref_code` attribute. The cross-referenced dictionary need not be based on the DDLM-STAR construction rules.

### CATEGORY `DICTIONARY_XREF`

<code>_definition.class</code>	Loop
<code>_name.category_id</code>	DICTIONARY
<code>_name.object_id</code>	XREF
<code>_category.key_id</code>	'_dictionary_xref.code'

### **`_dictionary_xref.code`**

Specifies the key code of the cross-referenced dictionary.

<code>_type.purpose</code>	Key
<code>_type.container</code>	Single
<code>_type.contents</code>	Code

### **`_dictionary_xref.date`**

Specifies the calendar date (format “yyyy-mm-dd”) of the cross-referenced dictionary.

<code>_type.purpose</code>	Audit
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Date

### **`_dictionary_xref.format`**

Specifies the format description of the cross-referenced dictionary.

<code>_type.purpose</code>	Audit
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Text

### **`_dictionary_xref.name`**

Specifies the common name of the cross-referenced dictionary.

<code>_type.purpose</code>	Audit
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Text

### **`_dictionary_xref.uri`**

Specifies the universal resource indicator of the cross-referenced dictionary.

<code>_type.purpose</code>	Audit
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Uri

## • ENUMERATION ATTRIBUTES

The `ENUMERATION` attributes specify any prescribed constraints on the values of defined items.

### CATEGORY ENUMERATION

<code>_definition.class</code>	Set
<code>_name.category_id</code>	ATTRIBUTES
<code>_name.object_id</code>	ENUMERATION

### **`_enumeration.default`**

Specifies the default value of the defined item, which is used if a value is not present in the instance data file.

<code>_type.purpose</code>	Limit
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Implied

### **`_enumeration.range`**

Specifies the range of values the defined item must lie within. The minimum and maximum values are separated by a colon ":" character.

<code>_type.purpose</code>	Limit
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Range

### **`_enumeration.def_index_id`**

Specifies the data name of an item whose coded value is used as an index to select a default enumeration value from the `_enumeration_default` loop category (see below). The code value must match one of the `_enumeration_default.index` values.

<code>_type.purpose</code>	Identify
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Tag

### **`_enumeration.mandatory`**

Specifies if it obligatory that the enumeration constraints (set by other attributes) **MUST** be adhered to in any validation process. The default is *Yes*.

<code>_type.purpose</code>	State
<code>_type.container</code>	Single
<code>_type.source</code>	Assigned
<code>_type.contents</code>	Code
loop_	
<code>_enumeration_set.state</code>	
<code>_enumeration_set.detail</code>	
	<b>Yes</b> 'Use of state is mandatory'
	<b>No</b> 'Use of state is unnecessary'
<code>_enumeration.default</code>	Yes

## **• ENUMERATION\_DEFAULT Attributes**

The `ENUMERATION_DEFAULT` attributes specify the allowed default values for the defined item. The single default value applicable for a specific instance document is determined by the value of tag identified by the attribute `_enumeration.def_index_id`. The code value is used as an index to select a default enumeration value from the `_enumeration_default` loop by matching one of the `_enumeration_default.index` values.

## **CATEGORY ENUMERATION\_DEFAULT**

<code>_definition.class</code>	Loop
--------------------------------	------

<code>_name.category_id</code>	ENUMERATION
<code>_name.object_id</code>	DEFAULT
<code>_category.key_id</code>	'_enumeration_default.index'

### **`_enumeration_default.index`**

Specifies the key index codes to the loop of eligible default values. This code is matched at instantiation time with the value of the item identified by the attribute `_enumeration.def_index_id`.

<code>_type.purpose</code>	Key
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Code

### **`_enumeration_default.value`**

Specifies eligible default values. The appropriate default is selected at instantiation time by matching the `_enumeration_default.index` code with that of the item identified by the attribute `_enumeration.def_index_id`.

<code>_type.purpose</code>	Limit
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Implied

## • **ENUMERATION\_SET Attributes**

The `ENUMERATION_SET` attributes specify a set of predetermined values (i.e. states) for an item.

### **CATEGORY ENUMERATION\_SET**

<code>_definition.class</code>	Loop
<code>_name.category_id</code>	ENUMERATION
<code>_name.object_id</code>	SET
<code>_category.key_id</code>	'_enumeration_set.state'

### **`_enumeration_set.state`**

Specifies permitted codes or “states” for a item.

<code>_type.purpose</code>	Key
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Code

### **`_enumeration_set.construct`**

Specifies the construction rules of permitted states in terms regular expression (REGEX) rules.

<code>_type.purpose</code>	Limit
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Regex

### **`_enumeration_set.detail`**

Specifies the description of a permitted enumeration state.

<code>_type.purpose</code>	Describe
<code>_type.source</code>	Assigned

<code>_type.container</code>	Single
<code>_type.contents</code>	Text

### **`_enumeration_set.table_key`**

Specifies the permitted key codes in a Table item for the form `{“key”:”value”, ...}`.

<code>_type.purpose</code>	Key
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Code

### **`_enumeration_set.xref_code`**

Specifies a cross-reference code for a permitted state with respect to the codes used in the dictionary identified with the `DICTIONARY_XREF` category attributes.

<code>_type.purpose</code>	Identify
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Code

### **`_enumeration_set.xref_dictionary`**

Specifies the code for the dictionary identified with the `DICTIONARY_XREF` category attributes.

<code>_type.purpose</code>	Link
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Code

## **• IMPORT Attributes**

The `IMPORT` attributes facilitate the importation of definition lines from external files. These attributes do not contribute to the direct definition of an item but provide a mechanism for inserting external definition material into a dictionary.

### **CATEGORY IMPORT**

<code>_definition.class</code>	Set
<code>_name.category_id</code>	ATTRIBUTES
<code>_name.object_id</code>	IMPORT

### **`_import.file_id`**

Specifies the name or URI of the file containing the definitions to be imported with `_import.get`.

<code>_type.purpose</code>	Identify
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Uri

### **`_import.frame_id`**

Specifies the frame code of the definition save frame containing the definitions to be imported.

<code>_type.purpose</code>	Identify
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Code

### **\_import.mode**

Specifies whether the save frame shell is imported, as well as the save frame contents.

<code>_type.purpose</code>	Identify
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Code
<code>loop_</code>	
<code>_enumeration_set.state</code>	
<code>_enumeration_set.detail</code>	
	<b>Full</b> 'import requested definition with frame'
	<b>Contents</b> 'import contents of requested defn frame'
<code>_enumeration.default</code>	Contents

### **\_import.if\_dupl**

Specifies the action to be taken if the imported definition block already exists in the importing dictionary file. The actions allowed appear as enumerated states.

<code>_type.purpose</code>	State
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Code
<code>loop_</code>	
<code>_enumeration_set.state</code>	
<code>_enumeration_set.detail</code>	
	<b>Ignore</b> 'ignore imported definitions if id conflict'
	<b>Replace</b> 'replace existing with imported definitions'
	<b>Exit</b> 'issue error exception and exit'
<code>_enumeration.default</code>	Exit

### **\_import.if\_miss**

Specifies the action to be taken if the imported definition block is missing from the file identified by `_import.file`. The actions allowed appear as enumerated states.

<code>_type.purpose</code>	State
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Code
<code>loop_</code>	
<code>_enumeration_set.state</code>	
<code>_enumeration_set.detail</code>	
	<b>Ignore</b> 'ignore import'
	<b>Exit</b> 'issue error exception and exit'
<code>_enumeration.default</code>	Exit
<code>save_</code>	

### **\_import.get**

A special “action” attribute used to insert the definition information specified by the other attributes and invoked as a list of tables, each table being responsible for an importation action.

<code>_type.purpose</code>	Import
<code>_type.source</code>	Assigned
<code>_type.container</code>	Table
<code>_type.contents</code>	List (Code)
<code>_type.dimension</code>	[[{}]]
<code>loop_</code>	
<code>_enumeration_set.table_key</code>	
<code>_enumeration_set.detail</code>	

```

        '1'   file  'filename/URI of source dictionary'
        '2'   save  'save framecode of source definition'
        '3'   mode  'mode for including save frames'
        '4'   dupl  'option for duplicate entries'
        '5'   miss  'option for missing duplicate entries'

    loop_
    _method.purpose
    _method.expression
    Evaluation
;
    With i as import
        _import.get = [{"file":i.file_id, "save":i.frame_id, "mode":i.mode,
                        "dupl":i.if_dupl, "miss":i.if_miss}]
;

```

## • LOOP Attributes

The `LOOP` category attributes specify the loop level of the defined item. For CIF data this will always be 1, but for STAR File data nested loops to any level are permitted.

### CATEGORY LOOP

<code>_definition.class</code>	Set
<code>_name.category_id</code>	ATTRIBUTES
<code>_name.object_id</code>	LOOP

### loop.level

Specifies the loop level of the defined item.

<code>_type.purpose</code>	Limit
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Index
<code>_enumeration.range</code>	1:
<code>_enumeration.default</code>	1

## • METHOD Attributes

The `METHOD` category attributes specify methods for expressing relationships between the defined item and other defined items.

### CATEGORY METHOD

<code>_definition.class</code>	Loop
<code>_name.category_id</code>	ATTRIBUTES
<code>_name.object_id</code>	METHOD
<code>_category.key_id</code>	'_method.purpose'

### method.purpose

Specifies the purpose code of the method for the defined item. Three method classes exist: *Evaluation, Definition and Validation.*

<code>_type.purpose</code>	State
<code>_type.container</code>	Single
<code>_type.source</code>	Assigned
<code>_type.contents</code>	Code
<code>loop_</code>	



_enumeration_set.state	
_enumeration_set.detail	
<b>Evaluation</b>	"method evaluates an item from related item values"
<b>Definition</b>	"method generates attribute value(s) in the definition"
<b>Validation</b>	"method compares an evaluation with existing item value"
_enumeration.default	Evaluation

### **\_method.expression**

Specifies the script, in the dREL language, relating the defined item to other items.

_type.purpose	Method
_type.source	Assigned
_type.container	Single
_type.contents	Text

## • **NAME Attributes**

The **NAME** attributes specify the name constructs of the defined item.

### **CATEGORY NAME**

_definition.scope	Set
_name.category_id	ATTRIBUTES
_name.object_id	NAME

### **\_name.object\_id**

Specifies the “object” identifier of the defined item or category that is available for use in methods scripts. This is a unique name string identifying a member of the category specified by

\_name.category\_id.

_type.purpose	Identify
_type.source	Assigned
_type.container	Single
_type.contents	Name

### **\_name.category\_id**

Specifies the “category” identifier of the defined item or category that is available for use in methods scripts. This is a unique name string identifying the parent of the category or item specified by

\_name.object\_id.

_type.purpose	Identify
_type.source	Assigned
_type.container	Single
_type.contents	Name

### **\_name.linked\_item\_id**

Specifies the data name of an item that the defined item is a derivative of, and implicitly dependent on. That is, the existence of the defined item depends on the linked item when used in an instance document. Its inclusion is mandatory for the definition of items of type ‘Su’; this attribute provides the data name of the Measurement item to which the standard uncertainty applies.

_type.purpose	Identify
_type.source	Assigned
_type.container	Single

<code>_type.contents</code>	Tag
-----------------------------	-----

## • TYPE Attributes

The `TYPE` attributes specify the nature and origin of the defined item.

### CATEGORY TYPE

<code>_definition.class</code>	Set
<code>_name.category_id</code>	ATTRIBUTES
<code>_name.object_id</code>	TYPE

### `_type.container`

Specifies the container type of the defined item. This is the simplest type description of the text string representing a value.

<code>_type.purpose</code>	State
<code>_type.container</code>	Single
<code>_type.source</code>	Assigned
<code>_type.contents</code>	Code
<code>_loop_</code>	
<code>_enumeration_set.state</code>	
<code>_enumeration_set.detail</code>	
<b>Single</b>	'single value'
<b>Multiple</b>	'values as List or by boolean , &!* or range : ops'
<b>List</b>	'''ordered set of values bounded by [] and separated by commas. Elements need not be of same contents type.'''
<b>Array</b>	'''ordered set of numerical values bounded by [] and separated by commas. Operations across arrays are equivalent to operations across elements of the Array.'''
<b>Matrix</b>	'''ordered set of numerical values for a tensor bounded by [] and separated by commas. Tensor operations such as dot and cross products, are valid cross matrix objects.'''
<b>Table</b>	'id:value elements bounded by {}; separated by commas'
<b>Ref-table</b>	'''a STAR construction with key:value elements bounded by \${..}\$ and separated by commas. The id tags below are privileged and optional. source - filename or URI block - data blockname frame - framecode or [framecode,framecode,..] item - dataname or [dataname,dataname,..] key - key value if item is in a list '''
<code>_enumeration.default</code>	Single

### `_type.contents`

Specifies the code identifying nature of the defined item. The allowed codes are specified in an enumeration list stored in the external file `com_val.dic`.

<code>_type.purpose</code>	State
<code>_type.source</code>	Assigned
<code>_type.container</code>	Multiple
<code>_type.contents</code>	Code
<code>_loop_</code>	
<code>_enumeration_set.state</code>	
<code>_enumeration_set.detail</code>	
<b>Text</b>	'case-sens strings or lines of STAR characters'

```

Code          'case-insens contig. string of STAR characters'
Name          'case-insens contig. string of alpha-num chars or underscore'
Tag          'case-insens contig. STAR string with leading underscore'
Filename      'case-sens string indentifying an external file'
Uri           'case-sens string as universal resource indicator of a file'
Date          'ISO standard date format <yyyy>-<mm>-<dd>'
Version       'version digit string of the form <major>.<version>.<update>'
Dimension     'integer limits of an Array/Matrix/List in square brackets'
Range         'inclusive range of numerical values min:max'
Count         'unsigned integer number'
Index         'unsigned non-zero integer number'
Integer       'positive or negative integer number'
Real          'floating-point real number'
Imag         'floating-point imaginary number'
Complex       'complex number <R>+j<I>'
Binary        'binary number \b<N>'
Hexadecimal   'hexadecimal number \x<N>'
Octal         'octal number \o<N>'
Implied       'implied by the context of the attribute'

_enumeration.default Text
_loop_
_description_example.case
_description_example.detail
    'Integer'          'elements are integer'
    'List(Real,Code)' 'elements in multiples of real number and codes'
    'Real|Code'        'elements either a real number or a code'

```

### **\_type.purpose**

Specifies the purpose or function code of the defined item.

```

_type.purpose          State
_type.source         Assigned
_type.container      Single
_type.contents       Code
_loop_
_enumeration_set.state
_enumeration_set.detail

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.
;
#.....the PURPOSE types used in all dictionaries .....
;
; 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 parsible.
;
; 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 may be used as a
; reference "key" to identify a specific packet of items
; within the category.
;
; Link
; Used to type an item with a value that is unique within
; a looped list of items belonging to another category.
; The definition of this item must contain the attribute
; "_name.linked_item_id" specifying the data name of the
; key item for this list. The defined item represents a
; a foreign key linking packets in this category list to
; packets in another category.
;
; 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 an item with the identical name
    except for the suffix '_su'. The definition of an SU item
    must include the attribute "_name.linked_item_id" which
    explicitly identifies the associated measurand item.
;
_enumeration.default Describe

```

### **\_type.source**

Specifies the origin or source code of the defined item.

```

_type.purpose           State
_type.source          Assigned
_type.container       Single
_type.contents        Code
_loop_
_enumeration_set.state
_enumeration_set.detail

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 as 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.
;

_enumeration.default Selected

```

### **\_type.dimension**

Specifies the dimensions (number of elements) of the defined item.

```

_type.purpose           Limit
_type.source          Assigned
_type.container       Single
_type.contents        Text
_loop_
_description_example.case
_description_example.detail
    "[3,3]"           '3x3 matrix of elements'
    "[6]"             'list of 6 elements'
    "[]"              'unknown number of list elements'

```

## • UNITS Attributes

The UNITS attributes specify the units of measurement for a defined item.

### CATEGORY UNITS

<code>_definition.class</code>	Set
<code>_definition.scope</code>	Category
<code>_name.category_id</code>	ATTRIBUTES
<code>_name.object_id</code>	UNITS

### `_units.code`

Specifies the name of the units of measurement of the defined. The allowed codes are specified as an enumeration list in the external enumeration template file `templ_enum.cif`.

<code>_type.purpose</code>	State
<code>_type.source</code>	Assigned
<code>_type.container</code>	Single
<code>_type.contents</code>	Code

The allowed `_units.code` values are imported from the external enumeration template file `templ_enum.cif` as follows.

```
loop_
  _enumeration_set.state
  _enumeration_set.detail
'none' "dimensionless - e.g. a ratio, factor, weight or scale"

'coulomb' "electronic charge in Coulombs"

'centimetres' "length 'centimetres (meters * 10^( -2))'"
'millimetres' "length 'millimetres (meters * 10^( -3))'"
'nanometres' "length 'nanometres (meters * 10^( -9))'"
'angstroms' "length 'angstroms (meters * 10^( -10))'"
'picometres' "length 'picometres (meters * 10^( -12))'"
'femtometres' "length 'femtometres (meters * 10^( -15))'"

'reciprocal_centimetres'
"per_length 'reciprocal centimetres (meters * 10^( -2)^-1)'"
'reciprocal_millimetres'
"per_length 'reciprocal millimetres (meters * 10^( -3)^-1)'"
'reciprocal_nanometres'
"per_length 'reciprocal nanometres (meters * 10^( -9)^-1)'"
'reciprocal_angstroms'
"per_length 'reciprocal angstroms (meters * 10^( -10)^-1)'"
'reciprocal_picometres'
"per_length 'reciprocal picometres (meters * 10^( -12)^-1)'"

'nanometre_squared'
"length_squared 'nanometres squared (meters * 10^( -9))^2'"
'angstrom_squared'
"length_squared 'angstroms squared (meters * 10^( -10))^2'"
'8pi_angstroms_squared'
"length_squared '8pi^2 * angstroms squared (meters * 10^( -10))^2'"
'picometre_squared'
"length_squared 'picometres squared (meters * 10^( -12))^2'"
'femtometre_squared'
"length_squared 'femtometres squared (meters * 10^( -12))^2'"

'nanometre_cubed'
"length_cubed 'nanometres cubed (meters * 10^( -9))^3'"
'angstrom_cubed'
"length_cubed 'angstroms cubed (meters * 10^( -10))^3'"
```

```

'picometre_cubed'
"length_cubed 'picometres cubed (meters * 10^(-12))^3'"

'grams_per_centimetre_cubed' "density 'grams per cubic centimetre'"
'kilograms_per_metre_cubed' "density 'kilograms per cubic metre'"
'megagrams_per_metre_cubed' "density 'megagrams per cubic metre'"

'kilopascals'      "pressure      'kilopascals'"
'gigapascals'     "pressure      'gigapascals'"

'hours'           "time          'hours'"
'minutes'         "time          'minutes'"
'seconds'         "time          'seconds'"
'microseconds'   "time          'microseconds'"

'degrees'         "angle         'degrees (of arc)'"

'degree_per_minute' "rotation_per_time 'degrees (of arc) per minute'"

'celsius'        "temperature    'degrees (of temperature) Celsius'"
'kelvins'        "temperature    'degrees (of temperature) Kelvin'"

'electrons'      "electrons      'electrons'"

'electron_squared' "electrons-squared 'electrons squared'"

'electrons_per_nanometre_cubed'
'electron-density 'electrons per nanometres cubed (meters * 10^( -9))^3'"
'electrons_per_angstrom_cubed'
"electron-density 'electrons per angstroms cubed (meters * 10^(-10))^3'"
'electrons_per_picometre_cubed'
"electron-density 'electrons per picometres cubed (meters * 10^(-12))^3'"

'dalton'         "standard atomic mass unit"

'pixel_per_millimetre' "area resolution unit"

'kilowatts'     "power          'kilowatts'"
'milliamperes'  "current        'milliamperes'"
'kilovolts'     "emf            'kilovolts'"

'volt_squared'  "emf            'volts squared'"

'arbitrary'     "arbitrary      'arbitrary system of units'"

enumeration.default Arbitrary

```