

DDLm Attributes

Main Features of *DDLm*

DDLm is an new dictionary definition language with the following main features:

- Provides methods expressions that relate defined items and enable data evaluation, validation and dynamic definition.
- Permits importation of definition information from other files and encourages unique modular dictionaries.
- Has richer data typing to facilitate preciser definitions.
- Provides for full definition hierarchy and associated definition inheritance.

Main *DDLm* Attribute Categories

- **ALIAS** - equivalent definitions in other dictionaries
- **CATEGORY** - groups or classes of attributes
- **DEFINITION** - definition id and classification information
- **DESCRIPTION** - human-readable descriptive information
- **DICTIONARY** - dictionary id and classification information
- **ENUMERATION** - constraints on the scope of the defined item
- **IMPORT** - import definitions into instance dictionary
- **LOOP** - loop_ level constraints
- **METHOD** - method expressions relating defined items
- **NAME** - dataname (tag) construction definition
- **TYPE** - data typing definition
- **UNITS** - measurement units definition

ALIAS Attributes

_alias.definition_id
_alias.dictionary_uri

tag of the aliased item
dictionary URI of the aliased item

DDL1

_related_item
_related_function

DDL2

_item_aliases.name
_item_aliases.alias_name
_item_aliases.dictionary
_item_aliases.version

CATEGORY Attributes

<code>_category.parent_id</code>	tag of parent category
<code>_category.parent_join</code>	Y/N if category can be merged with the parent category

DDL1

`_category`

DDL2

`_category.id`

`_category.description`

CATEGORY_KEY Attributes

_category_key.generic
_category_key.primitive

tag of generic list key in category
tag(s) of primitive key(s) in category

DDL1

DDL2

_category.implicit_key

CATEGORY_MANDATORY Attributes

_category_mandatory.item_id tag of mandatory item(s)
in category

DDL1

_list_mandatory

DDL2

_category.mandatory_code

DEFINITION Attributes

_definition.id	tag of item defined in definition
_definition.class	class of definition (audit, attribute, datum,...)
_definition.scope	scope of definition (item, category, dictionary)
_definition.update	date of definition
_definition.xref_code	code of equivalent def in DICTIONARY_XREF

DDL1

_name

DDL2

_item.name

DESCRIPTION Attributes

_description.key_words	key-words for defined item
_description.common	common name of defined item
_description.text	text description of defined item

DDL1

_definition

DDL2

_item_description.name

_item_description.description

DESCRIPTION_EXAMPLE Attributes

<code>_description_example.case</code>	example of defined item
<code>_description_example.detail</code>	description of example case

DDL1

`_example`

DDL2

`_item_examples.name`

`_item_examples.case`

`_item_examples.detail`

DICTIONARY Attributes

_dictionary.title	common title of dictionary
_dictionary.class	dictionary type (attr, instance,..)
_dictionary.date	date of current version
_dictionary.version	version code of dictionary
_dictionary.filename	filename of dictionary
_dictionary.uri	URI of dictionary file
_dictionary.namespace	prefix namespace of defined items
_dictionary.ddl_conformance	conformance DDL version

DICTIONARY_AUDIT Attributes

_dictionary_audit.version	version code of dictionary revision
_dictionary_audit.date	date of dictionary revision
_dictionary_audit.revision	description of dictionary revision

DDL1

_dictionary_history

DDL2

_dictionary_history.version
_dictionary_history.update
_dictionary_history.revision

DICTIONARY_VALID Attributes

_dictionary_valid.attributes
_dictionary_valid.scope

valid attributes for scope
definition scope (item,category ,dictionary)

DDL1

DDL2

DICTIONARY_XREF Attributes

<code>_dictionary_xref.code</code>	identity of x-referenced dictionary
<code>_dictionary_xref.date</code>	date revision of x-ref dictionary
<code>_dictionary_xref.format</code>	format of x-ref dictionary
<code>_dictionary_xref.name</code>	common name of x-ref dictionary
<code>_dictionary_xref.uri</code>	URI source of x-ref dictionary

DDL1

DDL2

ENUMERATION Attributes

<code>_enumeration.default</code>	default value if item not specified
<code>_enumeration.def_index_id</code>	tag of value used as default index
<code>_enumeration.range</code>	enumeration range (min:max)
<code>_enumeration.mandatory</code>	y/n if states are obligatory

DDL1

`_enumeration_default`

DDL2

`_item_default.name`
`_item_default.value`

ENUMERATION_DEFAULT Attributes

<code>_enumeration_default.index</code>	index of the default value
<code>_enumeration_default.value</code>	default value for this index

DDL1

DDL2

ENUMERATION_SET Attributes

<code>_enumeration_set.state</code>	enumeration state
<code>_enumeration_set.construct</code>	REGEX rules of state
<code>_enumeration_set.detail</code>	description of the state
<code>_enumeration_set.xref_code</code>	equiv state in x-ref dictionary
<code>_enumeration_set.xref_dictionary</code>	code of x-ref dictionary

DDL1

`_enumeration`
`_enumeration_detail`

DDL2

`_item_enumeration.name`
`_item_enumeration.value`
`_item_enumeration.detail`

IMPORT Attributes

_import.scope	scope of imports (<i>Dic, Cat, Grp, Def, Att, Sta, Val</i>)
_import.block	name of the imported definition block
_import.file	name of file containing definition block
_import.if_dupl	action taken if duplicate definition detected
_import.if_miss	action taken if import definition missing

DDL1

DDL2

IMPORT_LIST Attribute

_import_list.id list [] of _import.* attributes

DDL1

DDL2

LOOP Attribute

_loop.level loop level that defined item must reside

DDL1
_list

DDL2

METHOD Attributes

_method.purpose code specifying method expression purpose
_method.expression method expression

DDL1

DDL2

_item_methods.name
_item_methods.method_id

NAME Attributes

_name.category_id

category code in the item tag <cat>.<obj>

_name.object_id

object code in the item tag <cat>.<obj>

_name.linked_item_id

tag of an equivalent item in another category with common set of values

DDL1

DDL2

_item.category_id

TYPE Attributes

<code>_type.container</code>	container type of item (<i>single, multiple, list, tuple, ...</i>)
<code>_type.contents</code>	data type of item elements (<i>real, integer, text, ...</i>)
<code>_type.purpose</code>	type purpose/origin (<i>import, method, state, measure, ...</i>)
<code>_type.dimension</code>	dimension of a multi-element container

DDL1

`_type`
`_type_construct`
`_type_conditions`

DDL2

`_item_type.name`
`_item_type.code`
`_item_type_conditions.name`
`_item_type_conditions.code`

UNITS Attributes

_units.code enumerated state designating measurement units

DDL1

_units

DDL2

_item_units.name

_item_units.code

save_type.container	
_definition.id	'_type.container'
_definition.update	2007-07-18
_definition.class	Attribute
_name.category_id	type
_name.object_id	container
_type.purpose	State
_type.container	Single
_type.contents	Code
loop_	
_enumeration_set.state	
_enumeration_set.detail	
Single	'a single value'
Multiple	'values related by boolean ', &!*' or range ":" ops'
List	'list of values bounded by []; separated by commas'
Array	'fixed list of numbers bounded by []; separated by commas'
Tuple	'immutable List bounded by (); nested tuples allowed'
Table	'key:value elements bounded by { }; separated by commas'
Implied	'implied by type.container of associated value'
_enumeration.default	Single
save_	

Data TYPE specifying
the container of
the defined item

New multi-line delimiters....

Tuples	(val,...(val,...))
Tables	{key:val,...{...}}
Lists	[val,...[val,...]]
Arrays	[val,...[val,...]]

save_type.purpose	
_definition.id	'_type.purpose'
_type.purpose	State
_type.container	Single
_type.contents	Code
loop_	
_enumeration_set.state	
_enumeration_set.detail	

**Data TYPE specifying
the *purpose or origin*
of the defined item**

Import	Import definition lines from other dictionaries.
Method	Method expression in a dictionary definition relating defined items.
Audit	Audit information about the creation or conformance of a file.
Identify	Identify another item or file.
Describe	Free-form descriptive item intended for human interpretation only .
Limit	Numerical item used to limit the values of other items.
State	Codified item within a discrete list of enumerated states.
Key	Codified key to identifying packets of items in the same category.
Link	Codified linking key identifying packets in another category.
Assigned	Data item assigned in the modelling of measured/observed items.
Observed	Data item determined by observation/deduction. <u>No SU value.</u>
Measured	Numerical data measured or derived from a measurement.

With a SU value

- 1) appended in (..) at precision of trailing digits, *or*
- 2) separate item with same tag with a '_su' appended.

save_

CS = case-sensitive
 CI = case-insensitive

<i>loop_</i>	<i>_enumeration_set.state</i>	<i>_enumeration_set.detail</i>	<i>_enumeration_set.construct</i>
Achar	'alphanumeric character'		'[A-Za-z]'
ANchar	'alpha-numeric character'		'[A-Za-z0-9]'
Pchar	'printable character'		'[()\[\]_.,:~!@#%?+*=A-Za-z0-9 ^~]'
Ctag	'CI category tag'		' ANtag [] +'
Otag	'CI object tag'		'[()\[\]_&<>~!@#%?+*=A-Za-z0-9 ^~]'+'
Tag	'CI item tag'		'_ Ctag [.] Otag'
Code	'CI code used to index data'		'[()\[\]_&<>~!@#%?+*=A-Za-z0-9 ^~]'+'
Text	'CS string/lines'		'[[\n\t()+_.,:~!@#%?+*=A-Za-z0-9 ^~]*'
Filename	'CS name of external file'		'Otag'
Savename	'CI reference tag of saveframe'		'\$ Otag'
Digit	'single digit unsigned number'		'[0-9]'
Count	'unsigned integer number'		'[0-9]+'
Index	'unsigned non-zero integer'		'[1-9] Digit +'
Integer	'+ or - integer number'		'[+-]? Count'
Float	'floating-point real number'		'-?([0-9]+)([0-9]*[.][0-9]+)([1-9]+)?([eE][+-]?[0-9]+)?'
Real	'floating-point real number'		'Float'
Imag	'floating-point imaginary number'		'Real[jJ]'
Complex	'complex number'		'Real + Imag'
Binary	'binary number'		'0b[0-1]+'
Hexadec	'hexadecimal number'		'0x[0-7a-fA-F]+'
Octal	'octal number'		'0o[0-7]+'
Regex	'a REGEX conformant expression'		'???'
Date	'ISO date format yyyy-mm-dd'		'[0-9][0-9][0-9][0-9]-[0-1]?[0-9]-[0-3][0-9]'
YesorNo	'"yes" or "y" or "no" or "n"'		'[yes]?[y]?[no]?[n]?'
Uri	'universal resource indicator'		'Pchar +'
Version	'<major>.<version>.<update>'		'Count [.] Count [.] Count'
Dimension	'array dimensions List'		'[[Count [,]? +]]'
Range	'Inclusive range min:max'		'Integer ? : Integer ?'
Label	'code identifying an atom site'		'[()\[\]_&<>~!@#%?+*=A-Za-z0-9 ^~]'+'
Element	'element symbol of atom type'		'Achar +'
Formula	'code for a chemical formula'		'[()\[\]+-=*A-Za-z0-9]'+'
Sympop	'symmetry/lattice code for site'		'[0-1]?[0-9]?[0-9]_[0-9][0-9][0-9]'

**Data TYPE specifying the
 _type.contents of the defined
 item residing in com_val.dic**

Main CIF Dictionary shell

data_CIF_DIC

<code>_dictionary.title</code>	CIF_DIC
<code>_dictionary.class</code>	Instance
<code>_dictionary.version</code>	1.2.04
<code>_dictionary.date</code>	2008-08-05
<code>_dictionary.filename</code>	cif.dic
<code>_dictionary.uri</code>	www.iucr.org/cif/dic/cif.dic
<code>_dictionary.ddl_conformance</code>	3.7.12
<code>_dictionary.namespace</code>	CifDic:
<code>_description.text</code>	

`_import_list.id`

```
(('Dic', 'CIF_CORE', 'cif_core.dic', 'Exit', 'Exit'), # common core  
'Dic', 'CIF_SMOL', 'cif_smol.dic', 'Exit', 'Exit'), # small molecule  
'Dic', 'CIF_MMOL', 'cif_mmol.dic', 'Exit', 'Exit'), # macromolecule  
'Dic', 'CIF_POWD', 'cif_powd.dic', 'Exit', 'Exit'), # powder diffraction  
'Dic', 'CIF_RHOD', 'cif_rhod.dic', 'Exit', 'Exit')) # charge density
```

CORE CIF Dictionary shell

```
data_CIF_CORE
  _dictionary.title          CIF_CORE
  _dictionary.class         Instance
  _dictionary.version       1.2.04
  _dictionary.date          2008-08-05
  _dictionary.filename      cif_core.dic
  _dictionary.uri           www.iucr.org/cif/dic/cif_core.dic
  _dictionary.ddl_conformance 3.7.12
  _dictionary.namespace     CifCore:
  _description.text
```

;

Dictionary shell for the definitions of COMCIFS-approved CORE data used within the Crystallographic Information Framework.

;

```
_import_list.id
  (('Dic', 'CORE_CRYST', 'core_cryst.dic', 'Exit', 'Exit'), # core crystal data
  ('Dic', 'CORE_DIFFR', 'core_diffr.dic', 'Exit', 'Exit'), # core diffraction data
  ('Dic', 'CORE_STRUC', 'core_struc.dic', 'Exit', 'Exit'), # core structure data
  ('Dic', 'CORE_MODEL', 'core_model.dic', 'Exit', 'Exit'), # core modelling data
  ('Dic', 'CORE_PUBLN', 'core_publn.dic', 'Exit', 'Exit')) # core publication data
```

CORE_CRYSTAL Dictionary definition
Test example #1

```
save_cell.atomic_mass
  _definition.id      '_cell.atomic_mass'
  _definition.update  2006-06-20
  _description.text
```

```
;
```

Atomic mass of the contents of the unit cell. This is calculated from the atom sites present in the ATOM_TYPE list.

```
;
```

```
_description.common  'Cell Atomic Mass'
_name.category_id    cell
_name.object_id      atomic_mass
_type.purpose          Assigned
_type.container       Single
_type.contents        Real
_enumeration.range   0.:
_units.code           daltons
```

```
loop_
  _method.purpose
  _method.expression
  Evaluation
; mass = 0.
Loop a as atom_type {
  mass += a.number_in_cell * a.atomic_mass }
  _cell.atomic_mass = mass
;
```

```
save_
```

CORE_CRYSTAL Dictionary definition
Test example #2

```
save_cell.metric_tensor
  _definition.id      '_cell.metric_tensor'
  _definition.update  2006-06-20
  _description.text
;
  The direct space (covariant) metric tensor used to transform
  vectors and coordinates from real (direct) to reciprocal space.
;
  _description.common  'Cell Metric Tensor'
  _name.category_id   cell
  _name.object_id     metric_tensor
  _type.purpose         Measured
  _type.container     Array
  _type.contents      Real
  _type.dimension     [3,3]
  loop_
  _method.purpose
  _method.expression
    Evaluation
;    with v as cell_vector
      _cell.metric_tensor = Array ([[ v.a*v.a, v.a*v.b, v.a*v.c ],
                                     [ v.b*v.a, v.b*v.b, v.b*v.c ],
                                     [ v.c*v.a, v.c*v.b, v.c*v.c ]])
;
save_
```