4.9. DDL1 dictionary

BY S. R. HALL

This is version 1.4.1 of the dictionary definition language (DDL1) that provides a machine-readable description of the attributes of data items in the core CIF and related dictionaries (Chapters 4.1 to 4.4). This version of DDL is described in Chapter 2.5.

_category

Character string which identifies the natural grouping of data items to which the specified data item belongs. If the data item belongs in a looped list, then it must be grouped only with items from the same category, but there may be more than one looped list of the same category provided that each loop has its own independent reference item (see _list_reference).

_definition

The text description of the defined item.

_dictionary_history

A chronological record of the changes to the dictionary file containing the definition. Normally this item is stored in the separate data block labelled data_on_this_dictionary.

_dictionary_name

The name string which identifies the generic identity of the dictionary. The standard construction for these names is <application code>_<dictionary version>.dic. Normally this item is stored in the separate data block labelled data_on_this_dictionary. Examples: 'ddl_core.dic','cif_core.dic'.

_dictionary_update

The date that the dictionary was last updated. Normally this item is stored in the separate data block labelled data_on_this_dictionary.

_dictionary_version

The dictionary version number. Version numbers cannot decrease with updates. Normally this item is stored in the separate data block labelled data_on_this_dictionary.

_enumeration

Permitted value(s) for the defined item. May appear in list as essential element of loop structure.

_enumeration_default

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.

_enumeration_range

The range of values permitted for a defined item. This can apply to ‘numb’ or ‘char’ items which have a preordained sequence (e.g. numbers or alphabetical characters). The minimum value ‘min’ and maximum value ‘max’ are separated by a colon character. If ‘max’ is omitted, then the item can have any permitted value greater than or equal to ‘min’.

_example

An example value of the defined item.

_example_detail

A description of an example value for the defined item.

_list

Signals whether the defined item is declared in a looped list.

_list_level

Specifies the level of the loop structure in which a defined item with the attribute _list ‘yes’ or ‘both’ must be declared. The permitted range is 1 → ∞. Where no value is given, the assumed value is ‘1’.

_list_link_child

Identifies data item(s) by name which must have a value which matches that of the defined item. These items are referred to as ‘child’ references because they depend on the existence of the defined item.

_list_link_parent

Identifies a data item by name which must have a value which matches that of the defined item, and which must be present in the same data block as the defined item. This provides for a reference to the ‘parent’ data item.

_list_mandatory

Signals whether the defined item must be present in the loop structure containing other items of the designated_category. This property is transferrable to another data item which is identified by _related_item and has _related_function set as ‘alternate’.

_list_reference

Identifies the data item, or items, which must be present (collectively) in a looped list with the defined data item for the loop structure to be valid. The data item(s) identified by _list_reference provide a unique access code to each loop packet. Note that this property may be transferred to another item with _related_function ‘alternate’.

Affiliation: SYDNEY R. HALL, School of Biomedical and Chemical Sciences, University of Western Australia, Crawley, Perth, WA 6009, Australia.

471
LIST_UNIQUENESS

Identifies data items which, collectively, must have unique values for the loop structure of the designated category items to be valid. This attribute is specified in the definition of a data item with _list_mandatory set to ‘yes’.

May appear in list.

_name

The data name(s) of the defined item(s). If data items are closely related or represent an irreducible set, their names may be declared as a looped sequence in the same definition.

May appear in list.


_related_function

Specifies the relationship between the defined item and the item specified by _related_item. The following classifications are recognized: ‘alternate’ signals that the item referred to in _related_item has attributes that permit it to be used as an alternative to the defined item for validation purposes. ‘conversion’ signals that the item referred to in _related_item is equivalent to the defined item except for a predefined convention which requires a different _enumeration set. ‘conversion’ signals that the item referred to in _related_item is equivalent to the defined item except that different scaling or conversion factors are applied. ‘replace’ signals that the item referred to in _related_item may be used identically to replace the defined item.

Appears in list containing _related_item.

The data value must be one of the following:

alternate used alternatively for validation tests
conversion equivalent except for defined conversion
replace new definition replaces the current one

_related_item

Identifies data item(s) which have a classified relationship to the defined data item. The nature of this relationship is specified by _related_function.

May appear in list as essential element of loop structure.

_type

The type specification of the defined item. Type ‘numb’ identifies items which must have values that are identifiable numbers. The acceptable syntax for these numbers is application-dependent, but the formats illustrated by the following identical numbers are considered to be interchangeable: 42, 42.000, 0.42E2, 4.2E1, 420000D-4, 0.0000042D+07. Type ‘char’ identifies items which need not be interpretable numbers. The specification of these items must comply with the STAR syntax specification of either a ‘contiguous single-line string’ bounded by blanks or blank-quotes, or a ‘text string’ bounded by semicolons as the first character of a line. Type ‘null’ identifies items which appear in the dictionary for data-definition and descriptive purposes. These items serve no function outside the dictionary files.

The data value must be one of the following:

numb numerically interpretable string
char character or text string
null for dictionary purposes only

_type_conditions

Codes defining conditions on the _type specification. ‘su’ permits a number string to contain an appended standard uncertainty number enclosed within parentheses, e.g. 4.37(5). ‘esd’ is a deprecated synonym for ‘su’, arising from the former use of the term ‘estimated standard deviation’ for ‘standard uncertainty’, and permitting a number string to contain an appended standard uncertainty 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 v1, v2, v3 ... signals that v1, v2, v3 etc. are alternative values. The sequence v1::v2 signals that v1 and v2 are the boundary values of a continuous range of values satisfying the requirements of _enumeration for the defined item. Combinations of alternative and range sequences are permitted.

May appear in list.

The data value must be one of the following:

none no extra conditions apply to the defined _type
esd synonym for ‘su’
seq data may be declared as a permitted sequence
su numbers may have su’s appended within parentheses

Where no value is given, the assumed value is ‘none’.

_type_construct

String of characters specifying the construction of the data value for the defined data item. The construction is composed of two entities: (1) data names; (2) construction characters. The rules of construction conform to the regular expression (REGEX) specifications detailed in IEEE (1991) and International Tables for Crystallography (2005), Volume G, Chapter 2.5.


Where no value is given, the assumed value is ‘*’.

Example: ‘{%year} - {%month} - {%day}’ (a typical construction for _date).

_units

A unique code which identifies the units of the defined data item. A description of the units is provided in _units_detail.

Examples: ‘K’ (kelvins), ‘°’ (degrees Celsius), ‘r’ (radians), ‘e’ (electrons), ‘V’ (volts), ‘dal’ (daltons), ‘m’ (metres), ‘kg’ (kilograms), ‘s’ (seconds).

_units_detail

A description of the numerical units applicable to the defined item and identified by the code _units.