Chart 1

Dictionary name: **ddl_core.dic**

Dictionary last updated: 1995-05-16

_category (*char*)

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).

[category]

_definition (*char*)

The text description of the defined item.

[definition]

_dictionary_history (*char*)

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]

_dictionary_name (*char*)

The name string which identifies the generic identity of 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.

Example(s): 'ddl_core.dic', 'cif_mm_core.dic' [dictionary]

_dictionary_update (*char*)

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

Permitted values may be constructed as the regular expression (_chronology_year)-(_chronology_month)-(_chronology_day) [dictionary]

_dictionary_version (*numb*)

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.

[dictionary]

_enumeration (*char*)

Permitted value(s) for the defined item.

May appear in list as essential element of loop structure. [enumeration]

_enumeration_default (*char*)

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_default]

_enumeration_detail (*char*)

A description of a permitted value(s) for the defined item, as identified by _enumeration.

May appear in listcontaining _enumeration. [enumeration]

_enumeration_range (*char*)

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 alphabetic characters). If 'max' is omitted then the item can have any permitted value greater than or equal to 'min'.

Permitted values may be constructed as the regular expression (_sequence_minimum):((_sequence_maximum)?)

Example(s): '-4:10', 'a:z', 'B:R', '0:' [enumeration_range]

_example (*char*)

An example value of the defined item.

May appear in list as essential element of loop structure. [example]

_example_detail (*char*)

A description of an example value for the defined item.

May appear in listcontaining _example. [example]

_list (*char*)

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

yes	can only be declared in a looped list
no	cannot be declared in a looped list
both	declaration in a looped list optional

Where no value is given, the assumed value is 'no'. [list]

_list_level (*numb*)

Specifies the level of the loop structure in which a defined item, with the attribute _list 'yes' or 'both', must be declared.

Where no value is given, the assumed value is '1'. The permitted range is 1→. [list]

_list_link_child (*char*)

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.

May appear in list [list_link_child]

_list_link_parent (*char*)

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.

May appear in list [list_link_parent]

_list_mandatory (*char*)

Signals if 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'.

yes	required item in this category of looped list
no	optional item in this category of looped list

Where no value is given, the assumed value is 'no'. [list]

_list_reference (*char*)

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

May appear in list [list_reference]

_list_uniqueness (*char*)

Identifies data items which, collectively, must have a unique values for the loop structure of the designated _category items to be deemed valid. This attribute is specified in the definition of a data item th _list_mandatory set to 'yes'.

May appear in list [list_uniqueness]

_name (*char*)

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

Example(s): '_atom_site_label', '_atom_attach_all _atom_attach_ring', '_index_h _index_k _index_l', '_matrix_11 _matrix_12 _matrix_21 _matrix_22' [name]

_related_item (*char*)

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. [related]

_related_function (*char*)

Specifies the relationship between the defined item and the item specified by _related_item. The following classifications are recognised. ’alternate’ signals that the item referred to in _related_item has attributes that permit it to be used alternately to the defined item for validation purposes. ’convention’ 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.

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

Appears in list containing _related_item. [related]

_type (*char*)

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 .42E+2 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 semi-colons as 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 of the dictionary files.

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

[type]

_type_conditions (*char*)

Codes defining conditions on the _type specification. ’esd’ permits a number string to contain an appended standard deviation number enclosed within parentheses. E.g. 4.37(5) ’seq’ permits data to be declared as a sequence of values separated by a comma <,> or a colon <:>. * The sequence 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 alternate and range sequences are permitted.

none	no extra conditions apply to the defined _type
esd	numbers *may* have esd’s appended within ()
seq	data may be declared as a permitted sequence

May appear in list [type_conditions]

_type_construct (*char*)

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 the regular expression (REGEX) specificatiopns detailed in the IEEE document P1003.2 Draft 11.2 Sept 1991 (ftp file ’/doc/POSIX/1003.2/p121–140’).

Example(s): ’(_year)-(_month)-(_day)’(a typical construction for _date) [type_construct]

_units (*char*)

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

Example(s): ’K’, ’C’, ’e’, ’V’, ’Dal’, ’*mm*’, ’cm’, ’*mm*-1’, ’cm-1’, ’eA-3’, ’ep-3’, ’en-3’, ’A’, ’pm’, ’nm’, ’A2’, ’pm2’, ’nm2’, ’A3’, ’pm3’, ’nm3’, ’A-1’, ’pm-1’, ’nm-1’, ’kPa’, ’GPa’, ’sec’, ’min’, ’hr’ [units]

_units_detail (*char*)

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

[units]