STARDDL: A *STAR* DICTIONARY DEFINITION LANGUAGE

1. INTRODUCTION

Data dictionaries play an increasingly important role in the rapid and reliable exchange of electronic data. As data ontologies they provide for the precise, machine parseable, definition of data items and this is the basis for automatic data interpretation and validation. In the relatively short history of applying the *STAR File* format for universal exchange of data (Hall, 1991; Hall & Spadaccini, 1994), four different approaches to the definition of data items have evolved. Each advance has served to extend the scope and the precision of dictionary definitions.

The first approach used during the initial design of the *STAR File* was to define data items in a simple text file. The application of the *STAR* syntax to the development of the *crystallographic information file* (CIF) (Hall, Allen & Brown, 1991) and the *molecular information file* (MIF) (Allen, Barnard, Cook & Hall, 1995) led Cook to suggest that data definitions could be specified and machine parsed, as an electronic *STAR File*. The prototype *STAR* dictionary definition language (*DDL0*) (Cook, 1991) was subsequently refined and extended in the early nineties to become *DDL1*, which was, and still is, used for a number of discipline-specific dictionaries, such as *CoreCIF* and *PowderCIF* (see *www.iucr.org/iucr-top/cif/*). The specifications of *DDL1* have been published (Hall & Cook, 1995).

Following the specification of the *CoreCIF* and *PowderCIF* dictionaries, work started on the definition of data items used to describe macromolecular structure. This proved difficult using *DDL1* because of a need for stronger relational attributes, both between individual items and between categories of items. These requirements led to the development of a another version of the *STAR* dictionary definition language *DDL2* (Westbrook & Hall, 1995), which is the basis for the macromolecular CIF dictionary *mmCIF* (see *www.iucr.org/iucr-top/cif/mm/*).

There are strong similarities between the syntax and attributes of *DDL1* and *DDL2*, and these provide for a high level of compatibility between existing dictionaries. Most *STAR* parsing software is conversant with dictionaries in either format and existing dictionaries based on *DDL1* and *DDL2* have co-existed without too much inconvenience for the past five years.

The existence of two different definition approaches does, however, pose immediate and long-term problems for efficient data exchange and validation. In the first place, creators of new discipline dictionaries now need to choose between the two definition languages. Despite their similarity this is not a simple choice, in that *DDL2* identifies the relations between categories, but it has a more complicated syntax and definition structure because of this. On the other hand, *DDL1* has a smaller attribute set that is easier to understand, and this simplifies the creation of new definitions.

The recent development of a dictionary relational expression language, *dREL* (Spadaccini, Hall and Castleden, 2000) for use in *methods* attributes has been part of the creation of a new DDL, referred to as *StarDDL*. The *dREL* script language enables *STAR* data items to be used directly in relational expressions as object-oriented *variables*, and, in doing so, depends on the attributes of these items to be precisely defined in the dictionary. A StarDDL / *dREL* interpreter has been developed which parses a *STAR* dictionary to establish the type, functionality and inter-relationships of data items, in order that this information can be automatically applied to derived

data values. Such processes demand a higher level of objectivity, specificity and functionality than is currently available in the syntax of either *DDL1* or *DD2*.

It is the need for simpler more-accessible dictionaries that permit expression languages such as *dREL* that has led to the development of StarDDL. StarDDL provides a more hierarchical, concise and implicit approach to data definitions that should be easier to comprehend both by developers and browsers.

2. HOW DOES *STARDDL* COMPARE TO *DD1* AND *DDL2*?

2.1 GENERAL

In many respects the *StarDDL* attributes are equivalent to those for *DDL2*. *StarDDL* differs in that it has a small but richer vocabulary and an organisational framework that reduces definition redundancy. The richer definition capability has been achieved in part by making inheritance between data categories, in terms of the child relationships, implicit rather than needing to be defined explicitly.

2.2 IMPLICIT CHILD RELATIONSHIPS

The need to specify all parent-child relationships in DDL2 dictionaries leads to considerable redundancy; it is an undesirable overhead for dictionary creators, and it can obscure more important aspects of definitions. C*hild* relationships are not specified in *STARDDL* dictionaries, as these can be derived automatically on instatiation.

2.3 STRONGER DATA TYPING.

STARDDL provides attributes with much stronger data typing. In particular, items may be defined as the *container* TYPES: *Single*, *Multiple*, *Tables*, *Lists*, *Tuples*, *Vectors* and *Matrices*. In turn, the elements of these containers may be classified into a range of number and character types and there are other attributes which specify more detailed structuring of the containers; this being needed for manipulating data with the expression language *dREL*. In addition, *STARDDL* specifies measured data items explicitly as part of the typing so that associated uncertainty values can be handled more consistently (and in some cases automatically within the method expressions) than is possible with the rather *ad hoc* approaches in *DDL1* and *DDL2*.

2.4 CATEGORY INHERITANCE

In *STARDDL* dictionaries the hierarchical relationships between categories of data items is specified explicitly. Each category inherits the attributes of its parent category, excepted when these are re-specified locally.

2.5 DATA NAME STRUCTURES

For *STARDDL* dictionaries, the *name* of a category is commonly the lead string (i.e. the characters preceding the period) of an item name, *but it is not mandatory*. This flexibility in the data name contruction is possible because the *category name*, and the item *name extension* string (i.e. the characters following the period) are specified explicitly as separate attributes. Allowing category names and item names (within a category) to be independent has several advantages. It permits the names of *any* items in current use (some of which do not comply to the *DDL2* name rules), and it will also avoid the "progressive elongation" in names that occurs with deeply nested families of categories. Because the category name need not be an integral part of the data name in *StarDDL*, the embedded period in a name be used simply as the precursor to the "name extension"string. This construction has important implications for the methods script language *dREL* where the extension name is used as the unique "object"tag for items in the category.

3. DICTIONARY ORGANISATION

The file structure of a *STARDDL* dictionary conforms to the *Star File* syntax. Each data or category definition is contained within a *save_* frame, known as the "definition frame". Each frame contains a sequence of "attributes" represented as tag / value pairs. The definition frames are contained with a *data_* block, known as the "dictionary block".

3.1 DEFINITION FRAME

Each definition in a *STARDDL* dictionary is contained within a *save_* frame. That is, the definition starts with a save_*frame_name* statement and ends with a save_ statement. The "*frame_name*" usually matches the name of the particular category or item that is being defined within the frame. Here an example of a *category* definition of ATOM_SITE from the *CoreCIF* dictionary.

```
save_ATOM_SITE
    _definition.id              atom_site
    _definition.scope           Category
    _definition.class           Datum
    _definition.update          2000-11-03
    _description.text
;
     The CATEGORY of data items used to describe atomic site information
     used in crystallograhic structure studies.
;
    _description.compact       'AtomSiteList'
    _category.parent_id         structure
    _category_key.item_id     '_atom_site.label'
     save_
```

And here is an example of an *item* definition from the *CoreCIF* dictionary.

```
save_atom_site.description
    _definition.id             '_atom_site.description'
    _definition.update          2000-11-03
    _description.text
;
     A description of special aspects of this site. See also
     atom_site.refinement_flags.
;
    _description.compact       'AtomSiteDetails'
```

```
    _name.category_id              atom_site
    _name.attribute_id             description
    _type.container                Single
    _type.value                    Char
    _description_example.case     'Ag/Si disordered'
     save_
```

3.2 DICTIONARY BLOCK

Each *STAR* dictionary must start with a data block statement which identifies the dictionary. For example, in the *StarDDL* dictionary this statement is

```
data_DDL_STAR
```

whereas for the *CoreCIF* dictionary it is

```
data_CIF_CORE_STRUCTURAL
```

3.3 HEAD CATEGORY

Each dictionary contains a definition frame that specifies the attributes of the category that is the parent to all other categories in the dictionary. This is known as the "head" category, and its definition frame contains attributes that apply to the dictionary as a whole. These attributes are inherited by all other categories and items, *except where the same attribute in individual definitions override the inherited attribute value.* As an example, the head category for the *CoreCIF* dictionary follows.

```
save_CORE
    _dictionary.name               CORE_STRUCTURAL
    _dictionary.version            3.4.08
    _dictionary.date               2001-05-11
    _dictionary.uri               'www.crystal.uwa.edu.au/~syd/dic/core_3.dic'
    _definition.id                 core
    _definition.scope              Dictionary
    _definition.class              Registration
    _description.text
;
     CIF CORE DICTIONARY expressed in StarDDL

     This dictionary contains the definitions of data items that
     are considered CORE to structural studies in crystallography.
;
     save_
```

4. CATEGORY AND ITEM RELATIONSHIPS

4.1 CATEGORY HIERARCHY

As stated in section §3.3, each dictionary contains the definition of the head category and these attributes apply to all other definitions in the dictionary. The head category is at the top of a hierarchical tree of categories, and all other categories in the dictionary are, directly or indirectly, related to it. In a *STARDDL* dictionary, lower categories in the hierarchy inherit attribute values from their parent categories, except when these are superceded by the local specification of the same attribute.

Here is a sample of category names, and contained data item names, for data describing the crystal unit cell in the *CoreCIF* dictionary.

```
CRYSTAL
          CELL
         _cell.atomic_mass
         _cell.formula_units_Z
         _cell.metric_tensor
         _cell.orthogonal_matrix
         _cell.special_details
         _cell.volume
                   CELL_ANGLE
                  _cell_angle.alpha
                  _cell_angle.beta
                  _cell_angle.gamma
                   CELL_LENGTH
                  _cell_length.a
                  _cell_length.b
                  _cell_length.c
                   CELL_VECTOR
                  _cell_vector.a
                  _cell_vector.b
                  _cell_vector.c
                   CELL_MEASUREMENT
                  _cell_measurement.pressure
                  _cell_measurement.radiation
                  _cell_measurement.reflns_used
                  _cell_measurement.temperature
                  _cell_measurement.theta_max
                  _cell_measurement.theta_min
                  _cell_measurement.wavelength
                             CELL_MEASUREMENT_REFLN
                            _cell_measurement_refln.hkl
                            _cell_measurement_refln.theta
                            _cell_measurement_refln.index_h
                            _cell_measurement_refln.index_k
                            _cell_measurement_refln.index_l
```

The parent of each category is specified in category definition of with the attribute `_category.parent_id`. For example in the `CELL` category definition this attribute would have the value `'crystal'`; in the `CELL_ANGLE` category definition this attribute would have the value `'cell'`

As we shall see later, the ability to specify relationships between categories of items has important advantages in expressing the semantics of data.

4.2 List categories and Reference Keys

Items that appear more than once in a Star data file must be expressed as a looped list. Such items are usually grouped into a single category, known as a *list* category. For example, the coordinates of atomic sites are defined in the `ATOM_SITE` category (see the definition in 3.1 above).

```
loop_
_atom_site.label
_atom_site.fract_x
_atom_site.fract_y
_atom_site.fract_z
  o1  .5501(6)  .6371(6)  .1601(13)
  o2  .4012(6)  .5162(6)  .2290(12)
  o3  .2502(7)  .5705(7)  .6011(14)
  c1  .4170(8)  .6931(9)  .4965(18)
```

```
  c2  .3144(8)  .6702(9)  .6420(19)
  c3  .2789(9)  .7494(10)  .838(2)
```

This list represents a 2D table in which the data names are a *row of header tags* identifying *columns of values*. That is, the tag `_atom_site.label` refers to the column of values “o1”, “o2”,...,”c3”. In any such list at least one item must have unique values in order that other values in a row may be unambiguously accessed. These rows of values are often referred to as “packets”or “list instances”. The item that is designated to have “unique” values in a category is known as the “category key”. The *key* in the `ATOM_SITE` *category* above is `_atom_site.label` and each value of this item provides the access to the other three items in the packet. For example, “o3” points to the values “.2502(7)”, “.5705(7)” and “.6011(14)”, and no others! In a dictionary the category key is specified in the category definition with the `_category_key.item_id` attribute (see the `ATOM_SITE` definition in 3.1 above).

The attribute information in the `ATOM_SITE` *category* definition refers all `ATOM_SITE` items. That is, the properties of a category apply to all data items that are defined with the attribute `_name.category_id` set with the value `“atom_site”`.

The definition of the key item `_atom_site.label` is as follows.

```
save_atom_site.label
    _definition.id              '_atom_site.label'
    _definition.import_id       '_generic.atom_site_label'
    _description.compact         'AtomSiteLabel'
    _name.category_id             atom_site
    _name.attribute_id            label
     save_
```

Compared to the definition of `_atom_site.description` (see 3.1 above), the `_atom_site.label` definition is abbreviated in that the attribute `_definition.import_id` is used to import attributes from a generic definition `_generic.atom_site_label`. This definition shorthand is used here because atom label items appear in many different lists and each needs to be defined separately in a crystallographic dictionary. Because these labels are equivalent, in that they all originate from labels in the `ATOM_SITE` category, it is efficient to store common attributes in a *generic definition* and import these to the individual label definitions.

Here is the generic definition `_generic.atom_site_label`.

```
save_generic.atom_site_label
    _definition.id              '_generic.atom_site_label'
    _definition.class             Transient
    _definition.update            2000-11-03
    _description.text
;
     This label is a unique identifier for a particular site in the
     asymmetric unit of the crystal unit cell.
;
    _description.compact         'AtomSiteLabel'
    _name.category_id             generic
    _name.attribute_id            atom_site_label
    _name.linked_item_id        '_atom_site.label'
    _type.container               Single
    _type.value                   Uchar
    _type.purpose                 Label
     loop_
    _description_example.case    C12     Ca3g28     Fe3+17     H*251
                                 C_a_phe_83_a_0  Zn_Zn_301_A_0
```

```
     save_
```

The dictional “import protocol” assumes, as it does for inherited properties, that locally specified attributes take precedence over imported attributes.

4.3 JOINED CATEGORIES

As discussed in §4.2, the specification of a key item is essential for each *list* category. In this section we describe how, in special circumstances, separate list categories may be *joined*, at instantiation, into a single list. This is sometimes desirable if different categories have equivalent category keys and list structures, but, for reasons of data presentation and simplification, contained items are instantiated as separate lists. This separation may be efficient for lists in which there is significant disparity between the population of packets (i.e. row lengths). In such cases the densely and sparsely populated items are often defined as separate list categories, where one category is the parent of the other (see 4.1).

The ability to join equivalent categories is specified in the “child” category definition using the attribute `_category.join_set_id`. This specifies the name of the parent category into which the data items are merged at instantiation. The category keys of joined categories are interchangeable.

In the CoreCIF dictionary, an example of joined categories is `ATOM_SITE` and `ATOM_SITE_ANISO`. We have seen in 3.2 that the key item of the first is `_atom_site.label` and the definition below shows that the key item of the second is `_atom_site_aniso.label`. These are interchangeable.

```
save_ATOM_SITE_ANISO
    _definition.id               atom_site_aniso
    _definition.scope            Category
    _definition.class            Datum
    _definition.update           2000-11-03
    _description.text
;
     The CATEGORY of data items used to describe atomic site information
     used in crystallograhic structure studies.
;
    _description.compact        'AtomSiteAnisoList'
    _category.parent_id          atom_site
    _category.join_set_id        atom_site
    _category_key.item_id      '_atom_site_aniso.label'
     save_
```

Here is simple data instantiation showing items in these categories expressed as separate lists.

```
loop_
_atom_site.label
_atom_site.fract_x
_atom_site.fract_y
_atom_site.fract_z
  o1  .5501(6)  .6371(6)  .1601(13)
  o2  .4012(6)  .5162(6)  .2290(12)
  o3  .2502(7)  .5705(7)  .6011(14)

loop_
_atom_site_aniso.label
_atom_site_aniso.U_11
_atom_site_aniso.U_12
_atom_site_aniso.U_13
_atom_site_aniso.U_22
_atom_site_aniso.U_23
```

```
_atom_site_aniso.U_33
  o1   .035   .012   .003 .043   .001   .022
  o3   .048   .011   .021 .034   .009   .032
```

Because the categories ATOM_SITE and ATOM_SITE_ANISO are defined as joinable, these lists may be considered to be one list by parsers. That is, as follows.

```
loop_
_atom_site.label
_atom_site.fract_x
_atom_site.fract_y
_atom_site.fract_z
_atom_site_aniso.U_11
_atom_site_aniso.U_12
_atom_site_aniso.U_13
_atom_site_aniso.U_22
_atom_site_aniso.U_23
_atom_site_aniso.U_33
  o1   .5501(6) .6371(6) .1601(13) .035   .012   .003 .043   .001   .022
  o2   .4012(6) .5162(6) .2290(12) ? ? ? ? ? ?
  o3   .2502(7) .5705(7) .6011(14) .048   .011   .021 .034   .009   .032
```

The concept of a single joined list is important to the generality of data relational languages such as *dREL* used in *StarDDL* method expressions. Joinable lists are assumed to have equivalent key items, so that above, either of the items _atom_site.label and _atom_site_aniso.label may be used as the key to the joined list. This allows items in joined lists be addressed simply in terms of the extension names *of the parent category name*. That is, reference to ATOM_SITE.fract_x, and ATOM_SITE.U_11, point to the values for _atom_site.fract_x and _atom_site_aniso.U_11.

4.4 LINKED LIST KEYS

We know from §4.3 that category keys uniquely identify data packets in lists. If other items are dependent, or linked, to a list key this need to be specified in their definitions in order that they can also be used to reference list packets. In the example instance list below the item _atom_site.calc_attached_atom is the label of a calculated atom site in the same list.

```
loop_
_atom_site.label
_atom_site.type_symbol
_atom_site.fract_x
_atom_site.fract_y
_atom_site.fract_z
_atom_site.calc_attached_atom
  C1   C   .41520   .69430   .49560   .
  C2   C   .31850   .66960   .63180   H1
  C3   C   .27660   .75080   .84370   .
  H1   H   .41000   .72000   .47000   .

loop_
_atom_type.symbol
_atom_type_scat.dispersion_real
_atom_type_scat.dispersion_imag
_atom_type_scat.source
  O    .047   .032   'Int Tables Vol IV Tables 2.2B and 2.3.1'
  C    .017   .009   'Int Tables Vol IV Tables 2.2B and 2.3.1'
  H     0     0     'Stewart and Davidson'
```

This relationship is specified in the definition of `_atom_site.calc_attached_atom` with the attribute `_name.linked_item_id`.

```
save_atom_site.calc_attached_atom
    _definition.id              '_atom_site.calc_attached_atom'
    _definition.update           2000-11-03
    _description.text
;
     The _atom_site.label of the atom site to which the 'geometry-
     calculated' atom site is attached.
;
    _description.compact        'AtomSiteParentAtom'
    _name.category_id            atom_site
    _name.attribute_id           calc_attached_atom
    _name.linked_item_id        '_atom_site.label'
    _type.container              Single
    _type.value                  Uchar
    _type.purpose                Label
     save_
```

In the same instance list above the item `_atom_site.type_symbol` refers to the element symbols which are stored in `ATOM_TYPE` category list as the item `_atom_type.symbol`. Again, this is specified in the definition of `_atom_site.type_symbol` using the attribute `_name.linked_item_id`.

```
save_atom_site.type_symbol
    _definition.id              '_atom_site.type_symbol'
    _definition.update           2000-11-03
    _description.text
;
     A code to identify the atom specie(s) occupying this site.
     This code must match a corresponding _atom_type.symbol. The
     specification of this code is optional if component_0 of the
     _atom_site.label is used for this purpose. See _atom_type.symbol.
;
    _description.compact        'AtomSiteTypeSymbol'
    _name.category_id            atom_site
    _name.attribute_id           type_symbol
    _name.linked_item_id        '_atom_type.symbol'
    _type.container              Single
    _type.value                  Uchar
     save_
```

4.5 LINKED KEYS OF DERIVATIVE ITEMS

Data items fall into two origin classes. If a data value is measured or postulated, it is considered "primitive". Primitive data items cannot be derived from other data. All other data are classed as "derivative". Derivative data may be evaluated by knowing their relationship to other data, primitive and derivative. These distinctions are important in *StarDDL* dictionaries, because the definition of derivative data items may contain method expressions that enable the value, and indeed other attributes within the definition, to be expressed in terms of relationships to other items. Derivative data items may be instantiated singly or as entities in a list. In some cases the entire list category may be considered derivative.

In this section, we explain how category keys are related when one list category is derived from another. If a*n entire* list category is derivative, its list key will be dependent, or *linked*, to the key of list from which it was derived.

This is best understood from an example. We will use the dependence of a list of geometric bond distances between the atom sites in a molecule on the list of atom site coordinates. The distances form a "derivative list" because are derived from the coordinate list. Using the definitions in the *CoreCIF* dictionary, we see that the category GEOM_BOND items may be derived from category ATOM_SITE values. Here is an instance file containing these two lists.

```
loop_
_atom_site.label
_atom_site.fract_x
_atom_site.fract_y
_atom_site.fract_z
_atom_site.U_iso_or_equiv
_atom_site.adp_type
  c1  .41520  .69430  .49560  .03000  Uiso
  c2  .31850  .66960  .63180  .03000  Uiso
  c3  .27660  .75080  .84370  .03000  Uiso
  c4  .34400  .85470  .87960  .03000  Uiso
  c5  .44470  .87990  .74290  .03000  Uiso
  c6  .47570  .79210  .53701  .03000  Uiso
  c7  .45300  .61239  .27920  .03000  Uiso

loop_
_geom_bond.id
_geom_bond.distance
  [c1,c2]   1.3373(10)
  [c1,c6]   1.3134(10)
  [c1,c7]   1.4764(12)
  [c2,c3]   1.4707(10)
  [c3,c4]   1.4094(10)
  [c4,c5]   1.3786(11)
  [c5,c6]   1.4604(10)
```

These two lists are linked through their category keys. In the ATOM_SITE list, the defined key is _atom_site.label (see the definition in §3.1). In the GEOM_BOND list the key is _geom_bond.id which is a *tuple* made up of two ATOM_SITE keys identifying the "bonded" atoms in the site list. That is, the GEOM_BOND key is dependent on the existence of the ATOM_SITE keys.

We will now show how this is specified in a *StarDDL* dictionary. Here is the definition of the GEOM_BOND category indicating that _geom_bond.id is the category key. The definition of _geom_bond.id follows and here the relationship of the key to the individual atom labels _geom_bond.atom_site_label_1 and _geom_bond.atom_site_label_2 is specified in the _method.expression attribute.

```
save_GEOM_BOND
    _definition.id                 geom_bond
    _definition.scope              Category
    _definition.class              Datum
    _definition.update             2000-11-03
    _description.text
;
     The CATEGORY of data items used to specify the geometry bonds in the
     structural model as derived from the atomic sites.
;
    _category.parent_id            geom
    _category_key.item_id       '_geom_bond.id'
    save_
```

```
save_geom_bond.id
    _definition.id              '_geom_bond.id'
```

```
    _definition.update           2000-11-03
    _description.text
;
     Identity of bond distance in terms of the atom site labels and
     symmetry operators as pairs for each of the two "bonded" atom sites.
;
    _name.category_id            geom_bond
    _name.attribute_id           id
    _type.container              Tuple
    _type.value                  Uchar
    _type_array.dimension        [2]
    _method.expression
;
     With  a  as  geom_bond
         _geom_bond.id <- Tuple ([ a.atom_site_label_1, a.atom_site_label_2])
;
     save_
```

The following definition of _geom_bond.atom_site_label_1 shows how the dependency of this label on _atom_site.label is specified with the attribute _name.linked_item_id.

```
save_geom_bond.atom_site_label_1
    _definition.id              '_geom_bond.atom_site_label_1'
    _definition.update           2000-11-03
    _description.text
;
    Label of the first site in the geometry bond.
;
    _description.compact        'AtomSiteLabel'
    _name.category_id            geom_bond
    _name.attribute_id           atom_site_label_1
    _name.linked_item_id        '_atom_site.label'
    _type.container              Single
    _type.value                  Uchar
    _type.purpose                Label
     save_
```

That is, each value of _geom_bond.atom_site_label_1 (and _geom_bond.atom_site_label_2) must match a value of _atom_site.label, otherwise the file is in error. The link between these labels means that dictionary relational may access the coordinate packets in the ATOM_SITE list *directly* using _geom_bond.atom_site_label_1 or _geom_bond.atom_site_label_2 values in the GEOM_BOND list.

6. DESCRIPTION OF ATTRIBUTES IN A DDL DICTIONARY

The following items and categories are defined in the *STARDDL* dictionary. The category and item names are shown in upper and lower case letters, respectively.

```
ATTRIBUTES
          ALIAS
         _alias.id
         _alias.definition_id
         _alias.dictionary_uri
          CATEGORY
         _category.id
         _category.parent_id
         _category.join_set_id
```

```
                CATEGORY_KEY
               _category_key.id
               _category_key.item_id
               _category_key.relational
                CATEGORY_MANDATORY
               _category_mandatory.id
               _category_mandatory.item_id
 DEFINITION
_definition.class
_definition.id
_definition.import_id
_definition.scope
_definition.update
_definition.xref_code
 DESCRIPTION
_description.id
_description.abbreviated
_description.compact
_description.text
                DESCRIPTION_EXAMPLE
               _description_example.id
               _description_example.case
               _description_example.detail
 DICTIONARY
_dictionary.class
_dictionary.date
_dictionary.name
_dictionary.parent_name
_dictionary.parent_uri
_dictionary.uri
_dictionary.version
                DICTIONARY_AUDIT
               _dictionary_audit.date
               _dictionary_audit.version
               _dictionary_audit.revision
                DICTIONARY_VALID
               _dictionary_valid.attributes
               _dictionary_valid.scope
                DICTIONARY_XREF
               _dictionary_xref.date
               _dictionary_xref.format
               _dictionary_xref.name
 ENUMERATION
_enumeration.id
_enumeration.default
_enumeration.range
                ENUMERATION_SET
               _enumeration_set.id
               _enumeration_set.code
               _enumeration_set.construct
               _enumeration_set.detail
               _enumeration_set.xref_code
 LOOP
_loop.id
_loop.level
 METHOD
_method.id
_method.class
_method.expression
 NAME
_name.id
_name.attribute_id
_name.category_id
_name.linked_item_id
 TYPE
_type.id
_type.container
_type.value
_type.purpose
                TYPE_ARRAY
```

```
                    _type_array.id
                    _type_array.class
                    _type_array.dimension
                    _type_array.order
          UNITS
         _units.id
         _units.code
IMPORT_DDL
          GENERIC
         _generic.ddl_id
          CODES
         _codes.units_code
AUDIT_DDL
```

7. ALIAS ATTRIBUTES

The ALIAS data items are used as attributes in dictionary definitions to specify alias (i.e. alternative or substitutable) names. This is necessary when the equivalent data item has been defined in another dictionary or in the same dictionary with a different name.

```
save_ALIAS
     _definition.id                alias
    _definition.scope             Category
    _definition.class             Attribute
    _definition.update            2000-11-02
    _description.text
;
     The attributes used to specify the aliased names of definitions.
;
    _category.parent_id           attributes
    _category_key.item_id       '_alias.definition_id'
    _category_key.relational    '_alias.id'
     save_
```

The name of an aliased item in a dictionary definition is specified with the attribute _alias.definition_id.

```
save_alias.definition_id
    _definition.id              '_alias.definition_id'
    _definition.update            2000-11-02
    _description.text
;
     Identifier of an aliased definition.
;
    _name.category_id             alias
    _name.attribute_id            definition_id
    _type.container               Single
    _type.value                   Uchar
    _type.purpose                 Idname
     save_
```

The Universal Resource Identifier of the dictionary to which the aliased item in a dictionary definition belongs is specified with the attribute _alias.dictionary_uri.

```
save_alias.dictionary_uri
    _definition.id              '_alias.dictionary_uri'
    _definition.update            2000-11-02
    _description.text
;
```

```
     The URI of the dictionary to which the aliased name belongs.
;
    _name.category_id             alias
    _name.attribute_id            dictionary_uri
    _type.container               Single
    _type.value                   Char
    _type.purpose                 Uri
     save_
```

8. CATEGORY ATTRIBUTES

The CATEGORY attributes are used in dictionary definitions to specify the properties of a category of data items. In other words, values assigned to these items specify the properties of all data items in a category.

```
save_CATEGORY
    _definition.id                category
    _definition.scope             Category
    _definition.class             Attribute
    _definition.update            2000-11-02
    _description.text
;
     The attributes used to specify the properties of a
     "category" of data items.
;
    _category.parent_id           attributes
    _category_key.relational    '_category.id'
     save_
```

The attribute _category.parent_id specifies the *parent* (the next highest in the category hierarchy) of the category being defined (see also 4.1 above).

```
save_category.parent_id
    _definition.id              '_category.parent_id'
    _definition.update            2000-11-02
    _description.text
;
     The definition id of the category which is a higher member of the
     organisational hierarchy than the current category definition.
;
    _name.category_id             category
    _name.attribute_id            parent_id
    _type.container               Single
    _type.value                   Uchar
    _type.purpose                 Idname
     save_
```

The attribute _category.join_set_id specifies the parent category (the next highest in the category hierarchy) to which items in the defined category can be *joined* at instantiation time (see also 4.3 above).

```
save_category.join_set_id
    _definition.id              '_category.join_set_id'
    _definition.update            2000-11-02
    _description.text
;
     The definition id of the category to which the curently defined
```

```
     category may be joined or merged. This relationship presumes that
     the category keys are interchangeable and the two categories form
     a common basis set.
;
    _name.category_id            category
    _name.attribute_id           join_set_id
    _type.container              Single
    _type.value                  Uchar
    _type.purpose                Idname
     save_
```

8.1 CATEGORY_KEY Attributes

The category CATEGORY_KEY is a child category within the category CATEGORY. It contains the attributes that are used to specify the category key properties. This is the item that must have a unique value within a list in order to serve as a "reference key" to specific packets of item values within the list (see examples in 4.2).

```
save_CATEGORY_KEY
    _definition.id               category_key
    _definition.scope            Category
    _definition.class            Attribute
    _definition.update           2000-11-02
    _description.text
;
     The attributes used to specify the properties of a
     "category_key" of data items.
;
    _category.parent_id          category
    _category_key.item_id      '_category_key.id'
     save_
```

The attribute _category_key.item_id specifies the name of the item in the list category that is the category key.

```
save_category_key.item_id
    _definition.id             '_category_key.item_id'
    _definition.update           2000-11-02
    _description.text
;
     Name of the data item within the category which is a key reference
     item for accessing other items in the category. The value of this
     item must be unique in order to provide unambiguous access.
;
    _name.category_id            category_key
    _name.attribute_id           item_id
    _type.container              Single
    _type.value                  Uchar
    _type.purpose                Name
     save_
```

The attribute _category_key.relational specifies the name of the item category definitions that is used as a pseudo key in a relational database model.

```
save_category_key.relational
    _definition.id             '_category_key.relational'
    _definition.update           2000-11-06
    _description.text
;
     Name of the data item within the category which is the relational
```

```
     key to the category.
;
    _name.category_id            category_key
    _name.attribute_id           relational
    _type.container              Single
    _type.value                  Uchar
    _type.purpose                Name
     save_
```

8.2 CATEGORY_MANDATORY Attributes

The CATEGORY_MANDATORY category contains attributes used to describe items that *must* be present in a save frame or data block, if any other member of the same category are present.

```
save_CATEGORY_MANDATORY
    _definition.id               category_mandatory
    _definition.scope            Category
    _definition.class            Attribute
    _definition.update           2000-11-02
    _description.text
;
     The attributes used to specify the properties of a
     "category_mandatory" of data items.
;
    _category.parent_id          category
    _category_key.item_id      '_category_mandatory.item_id'
    _category_key.relational   '_category_mandatory.id'
     save_
```

The attribute _category_mandatory.item_id specifies the name of the item in the defined category that must be present in an instantiated document if any member of that category is present.

```
save_category_mandatory.item_id
    _definition.id             '_category_mandatory.item_id'
    _definition.update           2000-11-02
    _description.text
;
     The data name of an item which must exist if any item in the
     category appears in the save frame or data block.
;
    _name.category_id            category_mandatory
    _name.attribute_id           item_id
    _type.container              Single
    _type.value                  Uchar
    _type.purpose                Name
     save_
```

9. DEFINITION ATTRIBUTES

The DEFINITION category contains attributes that identify and type definitions in a dictionary.

```
save_DEFINITION
    _definition.id               definition
    _definition.scope            Category
    _definition.class            Attribute
    _definition.update           2000-11-02
```

```
    _description.text
;
     The attributes for classifying dictionary definitions.
;
    _category.parent_id          attributes
    _category_key.item_id       '_definition.id'
    _category_key.relational    '_definition.id'
    _category_mandatory.item_id '_definition.id'
     save_
```

The attribute _definition.class specifies the class of item being defined as enumeration states. Four states are permitted: *Registration*, *Attribute*, *Datum* and *Transient*.

```
save_definition.class
    _definition.id             '_definition.class'
    _definition.update          2000-11-10
    _description.text
;
     The nature and the function of a definition or definitions.
;
    _name.category_id           definition
    _name.attribute_id          class
    _type.container             Single
    _type.value                 Uchar
    _type.purpose               Code
     loop_
    _enumeration_set.code
    _enumeration_set.detail

     Registration
;            The definition of an item that is used to IDENTIFY and AUDIT
             the function and compilation of each data dictionary.
;
     Attribute
;            The definition of an item that is used to define the
             ATTRIBUTES (i.e. characteristics) of other data items.
             This class of definitions can only appear in a DDL dictionary.
;
     Datum
;            The definition of an item that describes the specific
             characteristic of a discipline-specific DATA ITEM.
;
     Transient
;            The specification of attributes and data which are used
             by other definitions in a dictionary. The definition names
             may have no function or existance outside of this dictionary.
;
    _enumeration.default        Datum
     save_
```

The attribute _definition.import_id specifies the name of a generically-defined item whose attributes must be imported into the definition. The rules of importation are that local attributes override imported attributes. See 4.2 above for an example use of this attribute.

```
save_definition.import_id
    _definition.id             '_definition.import_id'
    _definition.update          2000-11-13
    _description.text
;
     The identification of the save frame containing attributes to be
     imported into the defined data item. This identifier contains the
     _definition.id name and this may be preceded by a code name that
     can be mapped to an externally defined URI, or by the URI itself.
```

```
;
    _name.category_id             definition
    _name.attribute_id            import_id
    _type.container               Single
    _type.value                   Char
    _type.purpose                 Import
     loop_
    _description_example.case '_generic.atom_site_label'
                              'DIC5:_generic.atom_site_label'
                            'www.iugg.org/dic/period:_generic.atom_site_label'
     save_
```

The attribute `_definition.scope` specifies the *scope* of item being defined in terms of its inheritance. Three states are permitted: *Dictionary*, *Category* and *Item*

```
save_definition.scope
    _definition.id              '_definition.scope'
    _definition.update            2000-11-06
    _description.text
;
     The extent to which a definition affects other definitions.
;
    _name.category_id             definition
    _name.attribute_id            scope
    _type.container               Single
    _type.value                   Uchar
    _type.purpose                 Code
     loop_
    _enumeration_set.code
    _enumeration_set.detail
          Dictionary      "applies to all defined items in the dictionary"
          Category        "applies to all defined items in the category"
          Item            "applies to a single definition"
    _enumeration.default          Item
     save_
```

The attribute `_definition.update` specifies the calendar date (format "`yyyy-mm-dd`") that the defined item was last updated.

```
save_definition.update
    _definition.id              '_definition.update'
    _definition.update            2000-11-10
    _description.text
;
     The date that a definition was last changed.
;
    _name.category_id             definition
    _name.attribute_id            update
    _type.container               Single
    _type.value                   Char
    _type.purpose                 Date
     save_
```

The attribute `_definition.xref_code` specifies a code that identifies the same item defined in another dictionary identified by the `DICTIONARY_XREF` category of attributes.

```
save_definition.xref_code
    _definition.id              '_definition.xref_code'
    _definition.update            2000-11-02
    _description.text
;
     Code identifying the equivalent definition in the dictionary
```

```
     referenced by the DICTIONARY_XREF attributes.
;
    _name.category_id            definition
    _name.attribute_id           xref_code
    _type.container              Single
    _type.value                  Char
     save_
```

10. DESCRIPTION ATTRIBUTES

The DESCRIPTION category contains one data item which is the attribute _description.text for giving a grammatical description of the defined data item. The value of this item, i.e. the descriptive text, is not considered machine interpretable.

```
save_DESCRIPTION
    _definition.id               description
    _definition.scope            Category
    _definition.class            Attribute
    _definition.update           2000-11-02
    _description.text
;
     The attributes of descriptive (non-machine parseable) parts of
     definitions.
;
    _category.parent_id          attributes
    _category_key.relational   '_description.id'
     save_
```

The attribute _description.abbreviated specifies a sequence of blank delimited words that describe the defined item, and are suitable for a keyword search.

```
save_description.abbreviated
    _definition.id             '_description.abbreviated'
    _definition.update           2000-11-02
    _description.text
;
     Abbreviated description containing key-words of the item.
;
    _name.category_id            description
    _name.attribute_id           abbreviated
    _type.container              Single
    _type.value                  Char
     save_
```

The attribute _description.compact specifies the abbreviated name of the defined item.

```
save_description.compact
    _definition.id             '_description.compact'
    _definition.update           2000-11-02
    _description.text
;
     Compact concatenated description of the item.
;
    _name.category_id            description
    _name.attribute_id           compact
    _type.container              Single
    _type.value                  Char
```

```
     save_
```

The attribute _description.text specifies the text describing of the defined item.

```
save_description.text
    _definition.id              '_description.text'
    _definition.update           2000-11-02
    _description.text
;
     The text description of the defined item.
;
    _name.category_id            description
    _name.attribute_id           text
    _type.container              Single
    _type.value                  Char
     save_
```

10.1 DESCRIPTION_EXAMPLE Attributes

The DESCRIPTION_EXAMPLE category contains attributes that are used to specify descriptive examples in the defined item. The values of these attributes are not machine interpretable.

```
save_DESCRIPTION_EXAMPLE
    _definition.id               description_example
    _definition.scope            Category
    _definition.class            Attribute
    _definition.update           2000-11-02
    _description.text
;
     The attributes of descriptive (non-machine parseable) examples of
     values of the defined items.
;
    _category.parent_id          description
    _category_key.item_id      '_description_example.case'
    _category_key.relational   '_description_example.id'
     save_
```

The attribute _description_example.case specifies an example value for the defined item.

```
save_description_example.case
    _definition.id              '_description_example.case'
    _definition.update           2000-11-02
    _description.text
;
     An example case of the defined item.
;
    _name.category_id            description_example
    _name.attribute_id           case
    _type.container              Single
    _type.value                  Char
     save_
```

The attribute _description_example.detail specifies the text details of an example value for the defined item.

```
save_description_example.detail
    _definition.id              '_description_example.detail'
    _definition.update           2000-11-02
    _description.text
;
     A description of an example case for the defined item.
```

```
;
    _name.category_id              description_example
    _name.attribute_id             detail
    _type.container                Single
    _type.value                    Char
     save_
```

11. DICTIONARY ATTRIBUTES

The DICTIONARY category contains attributes that describe the aspects of the dictionary as a whole.

```
save_DICTIONARY
    _definition.id                 dictionary
    _definition.scope              Category
    _definition.class              Attribute
    _definition.update             2000-11-10
    _description.text
;
     Attributes for identifying and registering the dictionary. The items
     in this category are NOT used as attributes of INDIVIDUAL data items.
;
    _category.parent_id            attributes
     loop_
    _category_mandatory.item_id
                                  '_dictionary.name'
                                  '_dictionary.uri'
                                  '_dictionary.date'
                                  '_dictionary.version'
     save_
```

The attribute _dictionary.class specifies the nature of the items defined in the dictionary. Only two class states are permitted: *Ddl* (for dictionaries containing attribute definitions) and *Instance* (for dictionaries defining the items used in instance documents i.e. Star Files).

```
save_dictionary.class
    _definition.id               '_dictionary.class'
    _definition.update             2000-11-13
    _description.text
;
     The nature of data items defined in the dictionary.
;
    _name.category_id              dictionary
    _name.attribute_id             class
    _type.container                Single
    _type.value                    Uchar
    _type.purpose                  Code
     loop_
    _enumeration_set.code
    _enumeration_set.detail
              Ddl        'dictionaries specifying the definition language'
              Instance   'dictionaries specifying instance data items'
    _enumeration.default           Instance
     save_
```

The attribute _dictionary.date specifies the calendar date (format “yyyy-mm-dd”) that the dictionary was last updated.

```
save_dictionary.date
    _definition.id              '_dictionary.date'
    _definition.update            2000-11-10
    _description.text
;
     The date that the last dictionary revision took place.
;
    _name.category_id             dictionary
    _name.attribute_id            date
    _type.container               Single
    _type.value                   Char
    _type.purpose                 Date
     save_
```

The attribute _dictionary.name specifies the common name of the dictionary.

```
save_dictionary.name
    _definition.id              '_dictionary.name'
    _definition.update            2000-11-10
    _description.text
;
     The Common name of the dictionary.
;
    _name.category_id             dictionary
    _name.attribute_id            name
    _type.container               Single
    _type.value                   Char
    _type.purpose                 Idname
     save_
```

The attribute _dictionary.parent_name specifies the common name of a dictionary, or dictionaries, that need to used in association with the dictionary.

```
save_dictionary.parent_name
    _definition.id              '_dictionary.parent_name'
    _definition.update            2000-11-10
    _description.text
;
     The common name of the dictionary containing definitions that are
     referred to in the current dictionary.
;
    _name.category_id             dictionary
    _name.attribute_id            parent_name
    _type.container               Single
    _type.value                   Char
    _type.purpose                 Idname
     save_
```

The attribute _dictionary.parent_uri specifies the location of the dictionary, or dictionaries, that need to used in association with the dictionary.

```
save_dictionary.parent_uri
    _definition.id              '_dictionary.parent_uri'
    _definition.update            2000-11-10
    _description.text
;
     The universal resource indicator of the dictionary referred to in
     _dictionary.parent_name.
;
    _name.category_id             dictionary
    _name.attribute_id            parent_uri
```

```
    _type.container                 Single
    _type.value                     Char
    _type.purpose                   Uri
     save_
```

The attribute _dictionary.uri specifies the location of the current dictionary.

```
save_dictionary.uri
    _definition.id              '_dictionary.uri'
    _definition.update            2000-11-10
    _description.text
;
     The universal resource indicator of this dictionary.
;
    _name.category_id             dictionary
    _name.attribute_id            uri
    _type.container               Single
    _type.value                   Char
    _type.purpose                 Uri
     save_
```

The attribute _dictionary.version specifies the version code (nn.mm.ii) of the dictionary.

```
save_dictionary.version
    _definition.id              '_dictionary.version'
    _definition.update            2000-11-10
    _description.text
;
     A unique version identifier for the dictionary.
;
    _name.category_id             dictionary
    _name.attribute_id            version
    _type.container               Single
    _type.value                   Char
    _type.purpose                 Version
     save_
```

11.1 DICTIONARY AUDIT Attributes

The DICTIONARY_AUDIT data items describe the status and the origins of a dictionary. The category definition below shows that two of these items, _dictionary_audit.date and _dictionary_audit.version, are “mandatory” i.e. they must be present for the dictionary to be conformant.

```
save_DICTIONARY_AUDIT
    _definition.id                dictionary_audit
    _definition.scope             Category
    _definition.class             Attribute
    _definition.update            2000-11-02
    _description.text
;
     Attributes for identifying and registering the dictionary. The items
     in this category are NOT used as attributes of individual data items.
;
    _category.parent_id           dictionary
    _category_key.item_id        '_dictionary_audit.version'
    _category_mandatory.item_id '_dictionary_audit.date'
     save_
```

The attribute _dictionary_audit.date is used to specify the calendar date (format "yyyy-mm-dd") of the last revision of the dictionary.

```
save_dictionary_audit.date
    _definition.id              '_dictionary_audit.date'
    _definition.update           2000-11-10
    _description.text
;
     The date of each dictionary revision.
;
    _name.category_id            dictionary_audit
    _name.attribute_id           date
    _type.container              Single
    _type.value                  Char
    _type.purpose                Date
     save_
```

The attribute _dictionary_audit.revision is used to describe the revision applied.

```
save_dictionary_audit.revision
    _definition.id              '_dictionary_audit.revision'
    _definition.update           2000-11-02
    _description.text
;
     A description of the revision applied for the _dictionary_audit.version.
;
    _name.category_id            dictionary_audit
    _name.attribute_id           revision
    _type.container              Single
    _type.value                  Char
     save_
```

The attribute _dictionary_audit.version is used to specify the version code (nn.mm.ii) of a dictionary associated with a revision.

```
save_dictionary_audit.version
    _definition.id              '_dictionary_audit.version'
    _definition.update           2000-11-10
    _description.text
;
     A unique version identifier for each revision of the dictionary.
;
    _name.category_id            dictionary_audit
    _name.attribute_id           version
    _type.container              Single
    _type.value                  Char
    _type.purpose                Code
     save_
```

11.2 DICTIONARY_VALID Attributes

The DICTIONARY_VALID category contains attributes that are used to specify which attributes are essential, optional or prohibited in the definitions as a function of dictionary scope.

```
save_DICTIONARY_VALID
    _definition.id               dictionary_valid
    _definition.scope            Category
    _definition.class            Attribute
    _definition.update           2000-11-07
    _description.text
;
     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.parent_id          dictionary
    _category_key.item_id       '_dictionary_valid.scope'
    _category_mandatory.item_id '_dictionary_valid.attributes'
     save_
```

The attribute `_dictionary_valid.attributes` is used to list the names of attributes that are mandatory or prohibited.

```
save_dictionary_valid.attributes
    _definition.id             '_dictionary_valid.attributes'
    _definition.update          2000-11-07
    _description.text
;
     A list of the attribute names and the attribute categories that are
     either MANDATORY or PROHIBITED for the _definition.scope value
     specified in the corresponding _dictionary_valid.scope.  All
     unlisted attributes are considered optional.

     MANDATORY  attribute names are preceded by a "+" character.
     PROHIBITED attribute names are preceded by a "!" character.
;
    _name.category_id            dictionary_valid
    _name.attribute_id           attributes
    _type.container              Single
    _type.value                  Uchar
     save_
```

The attribute `_dictionary_valid.scope` is used to specify the dictionary scope associated with the `_dictionary_valid.attributes` lists.

```
save_dictionary_valid.scope
    _definition.id             '_dictionary_valid.scope'
    _definition.update          2000-11-07
    _description.text
;
     The _definition.scope code corresponding to the attribute list
     given in _dictionary_valid.attributes.
;
    _name.category_id            dictionary_valid
    _name.attribute_id           scope
    _name.linked_item_id        '_definition.scope'
    _type.container              Single
    _type.value                  Char
    _type.value                  Code
     save_
```

11.3 DICTIONARY_XREF Attributes

The `DICTIONARY_XREF` category contains attributes that specify the dictionary that items are cross-referenced to.

```
save_DICTIONARY_XREF
    _definition.id               dictionary_xref
    _definition.scope            Category
    _definition.class            Attribute
    _definition.update           2000-11-02
    _description.text
;
     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.parent_id          dictionary
     save_
```

The attribute _dictionary_xref.date is used to specify the calendar date (format "yyyy-mm-dd") of the cross-referenced dictionary.

```
save_dictionary_xref.date
    _definition.id             '_dictionary_xref.date'
    _definition.update           2000-11-02
    _description.text
;
     Date of the cross-referenced dictionary.
;
    _name.category_id            dictionary_xref
    _name.attribute_id           date
    _type.container              Single
    _type.value                  Char
    _type.purpose                Date
     save_
```

The attribute _dictionary_xref.format is used to specify the format of the cross-referenced dictionary.

```
save_dictionary_xref.format
    _definition.id             '_dictionary_xref.format'
    _definition.update           2000-11-02
    _description.text
;
     Format of the cross referenced dictionary.
;
    _name.category_id            dictionary_xref
    _name.attribute_id           format
    _type.container              Single
    _type.value                  Char
     save_
```

The attribute _dictionary_xref.name is used to specify the common name of the cross-referenced dictionary.

```
save_dictionary_xref.name
    _definition.id             '_dictionary_xref.name'
    _definition.update           2000-11-02
    _description.text
;
     Name of the cross referenced dictionary.
;
    _name.category_id            dictionary_xref
    _name.attribute_id           name
    _type.container              Single
    _type.value                  Char
     save_
```

12. ENUMERATION ATTRIBUTES

The ENUMERATION category contains attributes for specifying enumeration constraints for the values of defined data items.

```
save_ENUMERATION
    _definition.id                 enumeration
    _definition.scope              Category
    _definition.class              Attribute
    _definition.update             2000-11-02
    _description.text
;
     The attributes for restricting the values of defined data items.
;
    _category.parent_id            attributes
    _category_key.relational    '_enumeration.id'
     save_
```

The attribute _enumeration.default is used to set the value of a data item, which is implied if the value is not specified in the save frame or data block.

```
save_enumeration.default
    _definition.id              '_enumeration.default'
    _definition.update            2000-11-02
    _description.text
;
     The default value for the defined item if it is not specified explicitly.
;
    _name.category_id             enumeration
    _name.attribute_id            default
    _type.container               Implied
     save_
```

The attribute _enumeration.range is used to set the range of values permitted for a data item. The minimum and maximum values are separated by a colon ":" character.

```
save_enumeration.range
    _definition.id              '_enumeration.range'
    _definition.update            2000-11-02
    _description.text
;
     The inclusive range of values "from:to" allowed for the defined item.
;
    _name.category_id             enumeration
    _name.attribute_id            range
    _type.container               Single
    _type.value                   Char
    _type.purpose                 Range
     save_
```

12.1 ENUMERATION_SET Attributes

The ENUMERATION_SET category contains items that are used as the attributes for specifying the predetermined values of data items.

```
save_ENUMERATION_SET
    _definition.id                 enumeration_set
    _definition.scope              Category
    _definition.class              Attribute
    _definition.update             2000-11-02
    _description.text
;
```

```
     Attributes of data items with pre-determined values.
;
    _category.parent_id          enumeration
    _category_key.item_id      '_enumeration_set.code'
    _category_key.relational   '_enumeration_set.id'
     save_
```

The attribute _enumeration_set.code is used to specify permitted codes or “states” for a data item.

```
save_enumeration_set.code
    _definition.id             '_enumeration_set.code'
    _definition.update           2000-11-02
    _description.text
;
     A permitted code value for the defined item.
;
    _name.category_id            enumeration_set
    _name.attribute_id           code
    _type.container              Single
    _type.value                  Uchar
     save_
```

The attribute _enumeration_set.construct is used to specify the construction rules of permitted states for a data item, in terms regular expression (REGEX) rules.

```
save_enumeration_set.construct
    _definition.id             '_enumeration_set.construct'
    _definition.update           2000-11-02
    _description.text
;
     The construction rules of the value that the code describes.
     The construction rules conform to the regular expression
     (REGEX) specifications detailed in the IEEE document P1003.2
     Draft 11.2 Sept 1991 (ftp file '/doc/POSIX/1003.2/p121-140') with
     the *EXCEPTION* that they can include the '\n' and '\t' special
     characters.
;
    _name.category_id            enumeration_set
    _name.attribute_id           construct
    _type.container              Single
    _type.value                  Char
     save_
```

The attribute _enumeration_set.detail is used to describe a permitted state of a data item.

```
save_enumeration_set.detail
    _definition.id             '_enumeration_set.detail'
    _definition.update           2000-11-02
    _description.text
;
     The meaning of the code (identified by _enumeration_set.code)
     in terms of the value of the quantity it describes.
;
    _name.category_id            enumeration_set
    _name.attribute_id           detail
    _type.container              Single
    _type.value                  Char
     save_
```

The attribute _enumeration_set.xref_code is used to specify a cross-reference code for a permitted state of a data item, with respect to the codes used in the dictionary identified with the DICTIONARY_XREF category attributes.

```
save_enumeration_set.xref_code
    _definition.id              '_enumeration_set.xref_code'
    _definition.update            2000-11-02
    _description.text
;
     Code identifying the equivalent definition in the dictionary
     referenced by the DICTIONRY_XREF attributes.
;
    _name.category_id             enumeration_set
    _name.attribute_id            xref_code
    _type.container               Single
    _type.value                   Char
     save_
```

13. LOOP ATTRIBUTES

The LOOP category contains attributes for specifying the expected loop level of the defined data item. For CIF data this will always be 1, but for *STAR* data nested lists are permitted.

```
save_LOOP
    _definition.id                loop
    _definition.scope             Category
    _definition.class             Attribute
    _definition.update            2000-11-02
    _description.text
;
     Attributes for looped lists.
;
    _category.parent_id           attributes
    _category_key.relational    '_loop.id'
     save_
```

The attribute _loop.level is used to specify the loop level of the defined data item.

```
save_loop.level
    _definition.id              '_loop.level'
    _definition.update            2000-11-02
    _description.text
;
     Specifies the level of the loop structure in which a defined
     item must reside if it used in a looped list.
;
    _name.category_id             loop
    _name.attribute_id            level
    _type.container               numb
    _type.value                   Integer
    _type.purpose                 Index
    _enumeration.range            1:
    _enumeration.default          1
     save_
```

14. METHOD ATTRIBUTES

The METHOD category contains attributes for specifying methods for evaluating and validating the defined item in terms of other item values in the data file, or other method expressions in the dictionary.

```
save_METHOD
    _definition.id                 method
    _definition.scope              Category
    _definition.class              Attribute
    _definition.update             2000-11-02
    _description.text
;
     Methods used for evaluating and validating defined items.
;
    _category.parent_id            attributes
    _category_key.item_id        '_method.class'
    _category_key.relational     '_method.id'
     save_
```

The attribute _method.class is used to specify the purpose code of the method for the defined data item. Three method classes exist: *Evaluation*, *Definition* and *Validation*.

```
save_method.class
    _definition.id               '_method.class'
    _definition.update             2000-11-02
    _description.text
;
     The purpose and scope of the method expression.
;
    _name.category_id              method
    _name.attribute_id             class
    _type.container                Single
    _type.value                    Uchar
    _type.purpose                  Code
     loop_
    _enumeration_set.code
    _enumeration_set.detail
     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"
    _enumeration.default           Evaluation
     save_
```

The attribute _method.expression is used to specify the relational expression script, in the dREL language, of the method for the defined data item.

```
save_method.expression
    _definition.id               '_method.expression'
    _definition.update             2000-11-02
    _description.text
;
     The method expression for the defined item.
;
    _name.category_id              method
    _name.attribute_id             expression
    _type.container                Single
    _type.value                    Char
    _type.purpose                  Method
     save_
```

15. NAME ATTRIBUTES

The NAME category contains items that are used as the attributes for specifying the name constructs of defined data items.

```
save_NAME
    _definition.id                 name
    _definition.scope              Category
    _definition.class              Attribute
    _definition.update             2000-11-02
    _description.text
;
     Attributes for identifying items and item categories.
;
    _category.parent_id            attributes
    _category_key.relational    '_name.id'
     loop_
    _category_mandatory.item_id
                                '_name.attribute_id'
                                '_name.category_id'
     save_
```

The attribute _name.attribute_id is used to specify the “name attribute” of the defined data item. This is a unique name string in that category that follows the period in the data name.

```
save_name.attribute_id
    _definition.id              '_name.attribute_id'
    _definition.update            2000-11-02
    _description.text
;
     Identity of the attribute or extension of the defined data name
     which is unique within the category or mergable family of categories.
;
    _name.category_id              name
    _name.attribute_id             attribute_id
    _type.container                Single
    _type.value                    Uchar
    _type.purpose                  Idname
     save_
```

The attribute _name.category_id is used to specify the category name that the defined data item is a member of.

```
save_name.category_id
    _definition.id              '_name.category_id'
    _definition.update            2000-11-02
    _description.text
;
     Name of the category of the defined data item.
;
    _name.category_id              name
    _name.attribute_id             category_id
    _type.container                Single
    _type.value                    Uchar
    _type.purpose                  Idname
     save_
```

The attribute _name.linked_item_id is used to specify the data name of a category key that the defined data item is linked to. See 4.4 above.

```
save_name.linked_item_id
    _definition.id              '_name.linked_item_id'
    _definition.update           2000-11-02
    _description.text
;
     Name of a linked item in another category which has equivalent values.
;
    _name.category_id            name
    _name.attribute_id           linked_item_id
    _type.container              Single
    _type.value                  Uchar
    _type.purpose                Name
     save_
```

16. TYPE ATTRIBUTES

The TYPE category contains attributes for specifying the TYPE of the defined data item.

```
save_TYPE
    _definition.id               type
    _definition.scope            Category
    _definition.class            Attribute
    _definition.update           2000-11-02
    _description.text
;
    Attributes which specify the 'typing' of data items.
;
    _category.parent_id          attributes
    _category_key.relational   '_type.id'
     save_
```

The attribute _type.container is used to specify the container type of the defined data item. This is the simplest type description of the text string representing a value. Seven container types are recognised.

```
save_type.container
    _definition.id              '_type.container'
    _definition.update           2000-11-02
    _description.text
;
     The simplest description of the contents of the STAR text string.
;
    _name.category_id            type
    _name.attribute_id           container
    _type.container              Single
    _type.value                  Uchar
    _type.purpose                Code
     loop_
    _enumeration_set.code
    _enumeration_set.detail
        Single   'a single value'
        Multiple 'values related by boolean "|&!*" or range ":" ops'
        List     'an array of values'
        Vector   'an array of one dimension'
        Matrix   'an array of two dimensions or more'
```

```
        Table    'an immutable array of items with "key":value format'
        Implied  'implied by type.container of associated value'
    _enumeration.default         Single
     save_
```

The attribute _type.value is used to specify the implied structure of the defined data item.

```
save_type.value
    _definition.id             '_type.value'
    _definition.update           2000-11-02
    _description.text
;
     The specific 'typing' of individual data values in the defined object.
;
    _name.category_id            type
    _name.attribute_id           value
    _type.container              Single
    _type.value                  Uchar
    _type.purpose                Code
     loop_
    _enumeration_set.code
    _enumeration_set.detail
    _enumeration_set.construct

     Char          'a string of text (case-sensitive)'
                   '[][ \n\t(_,.;:"&<>/\{}'`~!@#$%?+=*A-Za-z0-9|^-]*'

     Uchar         'a string of text (case-insensitive)'
                   '[][ \n\t(_,.;:"&<>/\{}'`~!@#$%?+=*A-Za-z0-9|^-]*'

     Integer       'an integer number'
                   '-?[0-9]+'

     Float         'a floating-point real number'
          '-?(([0-9]+)|([0-9]*[.][0-9]+))([][0-9]+[])?([eE][+-]?[0-9]+)?'

     Real          'a floating-point real number'
          '-?(([0-9]+)|([0-9]*[.][0-9]+))([][0-9]+[])?([eE][+-]?[0-9]+)?'

     Imag          'a floating-point imaginary number "Real[jJ]"'
                   ' '

     Complex       'a complex number of the form "Real+Imag"'
                   ' '

     Binary        'a binary number preceded by "0b"'
                   ' '

     Hexadecimal   'a hexadecimal number preceded by "0x"'
                   ' '

     Octal         'a hexadecimal number preceded by "0o"'
                   ' '
    _enumeration.default         Char
     save_
```

The attribute _type.purpose is used to specify a special purpose of the defined data item.

```
save_type.purpose
    _definition.id             '_type.purpose'
    _definition.update           2000-11-02
    _description.text
;
     The structure and purpose of the data value.
;
```

```
    _name.category_id              type
    _name.attribute_id             purpose
    _type.container                Single
    _type.value                    Uchar
    _type.purpose                  Code
     loop_
    _enumeration_set.code
    _enumeration_set.detail
    _enumeration_set.construct

     Implied   'data value implied soley by the VALUE and CONTAINER typing'
               ' '

     Name      'a defined data item name (always case insensitive)'
               '_[^\t\n "]+'

     Idname    'an defined identifier or name component'
               '_A-Za-z0-9]+'

     Code      'value or values defined in an enumerated list'
               '_,.;:"&<>/\{}'`~!@#$%A-Za-z0-9*|+-]*'

     y/n       'the flag with values of "yes", "y", "no" or "n".'
               ' '

     Date      'ISO date format yyyy-mm-dd'
               '[0-9][0-9][0-9][0-9]-[0-9]?[0-9]-[0-9][0-9]'

     Index     'integer which is member of a defined sequential set'
               ' '

     Import
;               a string containing the name of a definition; a definition
                name preceded by "<code>:" where the code represents an
                externally defined URI; or a definition name preceded by
                "<URI>:"
;
     Uri        'an universal resource indicator string specifying a file'
                ' '

     Version  'version digit string of the form <major>.<version>.<update>'
               ' '

     Count     'positive or zero integer'
               ' '

     Dimension  'integer dimensions of an array in square brackets'
                ' '

     Range      'An inclusive range of numerical values min:max'
                ' '

     Method     'a method expression for evaluating the defined item'
                '[][ \n\t(_,.;:"&<>/\{}'`~!@#$%?+=*A-Za-z0-9|^-]*'

     Measurement
;               a number which can have a standard uncertainty value
                 either
                     1) appended as integers in parentheses at the
                        precision of the trailing digits, or
                     2) as a separate item with the same name as
                        defined item but with a trailing underscore su.
;
                  ' '
    _enumeration.default       Implied
     save_
```

16.1 TYPE_ARRAY Attributes

The TYPE_ARRAY category contains attributes for specifying details of data item defined as "arrays" (i.e. a List, Tuple, Vector, Matrix or Table).

```
save_TYPE_ARRAY
    _definition.id              type_array
    _definition.scope           Category
    _definition.class           Attribute
    _definition.update          2000-11-02
    _description.text
;
    The attributes of array lists.
;
    _category.parent_id         type
    _category_key.relational   '_type_array.id'
     save_
```

The attribute _type_array.class is used to specify the array class of the defined data item.

```
save_type_array.class
    _definition.id             '_type_array.class'
    _definition.update          2000-11-02
    _description.text
;
     The nature of the array.
;
    _name.category_id           type_array
    _name.attribute_id          class
    _type.container             Single
    _type.value                 Uchar
    _type.purpose               Code
     loop_
    _enumeration_set.code
    _enumeration_set.detail
     Asymmetric       'no symmetry constraints'
     Diagonal         'a diagonal matrix'
     Symmetric        'a symmetric matrix'
     Anti_Symmetric   'an anti-symmetric matrix'
     Upper_Triangular 'a matrix with lower triangular zero elements'
     Lower_Triangular 'a matrix with upper triangular zero elements'
    _enumeration.default        Asymmetric
     save_
```

The attribute _type_array.dimension is used to specify the array dimensions (number of elements in each dimension) of the defined data item.

```
save_type_array.dimension
    _definition.id             '_type_array.dimension'
    _definition.update          2000-11-02
    _description.text
;
     The dimensions of the list array bounded by square brackets.
     Each dimension may be expressed simply as an integer giving
     the maximum index permitted (the minimum is assumed to be 1).
     Alternately, each dimension may be entered in the form:
     <min_index>:<max_index>
;
    _name.category_id           type_array
    _name.attribute_id          dimension
    _type.container             Single
    _type.value                 Char
```

```
    _type.purpose                Dimension
     loop_
    _description_example.case
    _description_example.detail  "[3,3]"   '3x3 Matrix'
                                 "[6]"     '6 element Tuple'
                                 "[4[2]]"  'Tuple of four Tuples of 2 elements'
     save_
```

The attribute _type_array.order is used to specify the array order (the order of the elements in each dimension) of the defined data item.

```
save_type_array.order
    _definition.id             '_type_array.order'
    _definition.update           2000-11-02
    _description.text
;
     The number of elements in each order of the list array.
;
    _name.category_id            type_array
    _name.attribute_id           order
    _type.container              Single
    _type.value                  Uchar
    _type.purpose                Code
     loop_
    _enumeration_set.code
    _enumeration_set.detail Column_Major 'first dimension list is by column'
                            Row_Major    'first dimension list is by row'
    _enumeration.default    Row_Major
     save_
```

17. UNITS ATTRIBUTES

The UNITS category contains attributes for specifying the units of measurement for a defined data item.

```
save_UNITS
    _definition.id               units
    _definition.scope            Category
    _definition.class            Attribute
    _definition.update           2000-11-02
    _description.text
;
    The attributes for specifying units of measure.
;
    _category.parent_id          attributes
    _category_key.relational   '_units.id'
     save_
```

The attribute _units.code is used to specify the name of the units of measurement of the defined data item.

```
save_units.code
    _definition.id             '_units.code'
    _definition.update           2000-11-02
    _definition.import_id      'GLOBAL:_codes.units_code'
    _description.text
;
     A code which identifies the units of measurement.
```

```
;
    _name.category_id              units
    _name.attribute_id             code
    _type.container                Single
    _type.value                    Uchar
    _type.purpose                  Code
     save_
```

18. IMPORT_DDL ATTRIBUTES

The IMPORT_DDL category contains items which provide a generic definition set of attributes for importation to other definition. These items are not used outside the dictionary.

```
save_IMPORT_DDL
    _definition.id                 import_ddl
    _definition.scope              Category
    _definition.class              Transient
    _definition.update             2000-11-02
    _description.text
;
     The MAJOR CATEGORY of definitions used to specify attributes which are
     imported into other items defined in the DDL dictionary.
;
    _category.parent_id            ddl
     save_
```

18.1 GENERIC Attributes

The GENERIC category contains items which provide a generic definition set of attributes for importation to other definition. These items are not used outside the dictionary.

```
save_GENERIC

    _definition.id                 generic
    _definition.scope              Category
    _definition.class              Transient
    _definition.update             2000-11-02
    _description.text
;
     Definitions used to specify generic attributes which are imported into
     items defined in this dictionary.
;
    _category.parent_id            import_ddl
     save_
```

The item _generic.ddl_id is used as a generic definition for importation to the definition of "relational id" items in the DDL dictionary.

```
save_generic.ddl_id
    _definition.id              '_generic.ddl_id'
    _definition.class             Registration
    _description.text
;
     Identity name of the ddl category used to internally reference the
     items defined in a DDL category.
;
    _name.category_id              generic
```

```
    _name.attribute_id          ddl_id
    _type.container             Single
    _type.value                 Uchar
    _type.purpose               IdName
     save_
```

18.2 CODES Attributes

The CODES category contains items which form the enumeration list of _units.code values and are imported into the definition of this item. These lists are discipline dependent.

```
save_CODES
    _definition.id              codes
    _definition.scope           Category
    _definition.class           Transient
    _definition.update          2000-11-02
    _description.text
;
     Lists of enumeration codes which are imported into items
     defined in the dictionary.
;
    _category.parent_id         import_ddl
     save_
```

The item _codes.units_code is used as an enumeration list for importation to the definition of _units.code.

```
save_codes.units_code
    _definition.id             '_codes.units_code'
    _definition.update          2000-11-02
    _description.text
;
     Enumeration data imported into the _units.code attribute.
;
    _name.category_id           codes
    _name.attribute_id          units_code
     loop_
    _enumeration_set.code
    _enumeration_set.detail

   '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'"

   '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'"

   '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'"

   'electron_per_nanometres_cubed'
   "electron-density 'electrons per nanometres cubed (meters * 10^( -9))^3'"
   'electron_per_angstroms_cubed'
   "electron-density 'electrons per angstroms  cubed (meters * 10^(-10))^3'"
   'electron_per_picometres_cubed'
   "electron-density 'electrons per picometres cubed (meters * 10^(-12))^3'"

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

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

    save_
```

19. REFERENCES

Allen, F.H., Barnard, J.M., Cook, A.F.P. & Hall, S.R. (1995). *J. Chem. Inform. Comp. Sci*. **35**, 412-427.

Cook, A.F.P. (1991) “Dictionary Definition Language in STAR File Format.” *ORAC Report*.

Hall, S.R. *The STAR File:* (1991) “A New Format for Electronic Data Transfer and Archiving.” *J Chemical Information and Computer Science* **31**, 326-333.

Hall, S.R., Allen, F.H. & Brown, I.D. (1991) “The Crystallographic Information File (CIF): A New Standard Archive File for Crystallography.” *Acta Cryst*. **A47**, 655-685.

Hall, S.R. & Cook, A.P.F. (1995) “STAR Data Definition Language: Initial Specification.” *J Chemical Information and Computer Science* **35**, 819-825.

Hall, S.R. & Spadaccini, N. (1994) “The STAR File: Detailed Specifications.” *J Chemical Information and Computer Science* **34**, *505-508.*

Spadaccini, N., Hall, S.R., and Castleden, I.R. (2000) “Relational Expressions in STAR File Dictionaries.” *J Chemical Information and Computer Science* **40**, 1289-1301

Westbrook, J.D & Hall, S.R. (1995) “A Dictionary Description Language for Macromolecular Structure.” Draft DDL V 2.1.1 (draft from *ndbserver.rutgers.edu/mmcif/ddl/*)