

## 3. CIF DATA DEFINITION AND CLASSIFICATION

sible to write a program that can deduce the structure of a standard reference within an undifferentiated reference list (provided the journal guidelines have been followed by the author) to the extent that enough information can be extracted to add hyperlinks to references using a cross-publisher reference linking service such as CrossRef (CrossRef, 2004). Therefore, in practice, IUCr journals still ask the author of an article to supply their reference list in the `_publ_section_references` field, rather than using the apparently more useful `_citation_` fields. It remains to be seen whether this is the best strategy in the long term.

In more technical topic areas, the details of an experimental instrument could be described by a huge number of possible data names, ranging from the manufacturer's serial number to the colour of the instrument casing. However, many of these details are irrelevant to the analysis of the data generated by the instrument, so the characteristics of an instrument that are assigned individual data names are typically just those parameters that need to be entered in equations describing the calibration or interpretation of the data it generates.

## 3.1.7.2. Category 'special details' fields

When the specific items in a particular topic area that need to be recorded under their own data names have been decided, there is likely to be other information that could be recorded, but is felt to be irrelevant to the immediate purposes of the data collection and analysis. It is good practice to provide a place in the CIF for such additional information; it encourages an author to record the information and permits data mining at a later stage. Each category typically contains a data name with the suffix `_details` (or `_special_details`) which identifies a text field in which additional information relating to the category may be stored. This field often contains explanatory text qualifying the information recorded elsewhere in the same category, but it might contain additional specific items of information for which no data name is given and for which no obvious application is envisaged. This helps to guard against the loss of information that might be put to good use in the future. Of course, if a `*_details` field is regularly used to store some specific item of information *and* this information is seen to be valuable in the analysis or interpretation of data elsewhere in the file, there is a case for defining a new, separate tag for this information.

## 3.1.7.3. Construction of data names

Since a dictionary definition contains all the machine-readable attributes necessary for validating the contents of a data field, the data name itself may be an arbitrary tag, devoid of semantic content. However, while dictionary-driven access to a CIF is useful in many cases, there are circumstances where it is useful to browse the file. It is therefore helpful to construct a data name in a way that gives a good indication of the quantity described. From the beginning, CIF data names have been constructed from self-descriptive components in an order that reflects the hierarchical relationship of the component ideas, from highest (most general) level to lowest (most specific) level when read from left to right.

In a typical example from the core CIF dictionary, the data name `_atom_site_type_symbol` defines a code (`symbol`) indicating the chemical nature (`type`) of the occupant of a location in the crystal lattice (`atom_site`). The equivalent data name from the mmCIF dictionary, `_atom_site.type_symbol`, explicitly separates the category to which the data name belongs from its more specific qualifiers by using a full stop (.) instead of an underscore (\_). While this use of a full stop is mandated in DDL2 dictionaries, it should

<code>_database_code_CSD</code>	(a)	'VOBYUG'
<code>_database_2.database_id</code>		'PDB'
<code>_database_2.database_code</code>	(b)	'5HVP'

Fig. 3.1.7.1. Alternative quantities described (a) by data-name extension (core dictionary) or (b) by paired data names (mmCIF dictionary).

nevertheless be considered a convenience, since the category membership is explicitly listed in the dictionary definition frame for every data name.

However, it may not always be easy to establish the best order of components when constructing a new data name. In the JOURNAL category, there was initially some uncertainty about whether to associate the telephone numbers of different contact persons by appending codes such as `_coeditor` and `_techeditor` to a common base name. In the end, the order of components was reversed to give names like `_journal_coeditor_phone` and `_journal_techeditor_phone`. Examining the JOURNAL category in the core CIF dictionary will show why this was done. Similarly, the extension of geometry categories to include details of hydrogen bonding went through a stage of discussing adding new data names to the existing categories, but with suffixes indicating that the components were participating in hydrogen bonding, before it was decided that a completely new category for describing all elements of a hydrogen bond was justified. These examples show that the correct ordering of components within a data name is closely related to the perceived classification of data names by category and subcategory.

Sometimes it is useful to differentiate alternative data items by appending a suffix to a root data name. For example, the core dictionary defines several data names for recording the reference codes associated with a data block by different databases: `_database_code_CAS`, `_database_code_CSD` etc. This is convenient where there are two or three alternatives, but becomes unwieldy when the number of possibilities increases, because new data names need to be defined for each new alternative case. A better solution is to have a single base name and a companion data item that defines which of the available alternatives the base item refers to. The mmCIF dictionary follows this principle: the category DATABASE\_2 contains two data names, `_database_2.database_code` (the value of which is an assigned database code) and `_database_2.database_id` (the value of which identifies which of the possible databases assigned the code) (Fig. 3.1.7.1).

Note the distinction between a data name constructed with a suffix indicating a particular database, and a data name which incorporates a prefix registered for the private use of a database. The data name `_database_code_PDB` is a *public* data name specifying an entry in the Protein Data Bank, while `_pdb_database_code` is a *private* data name used for some internal purpose by the Protein Data Bank (see Section 3.1.8.2).

## 3.1.7.4. Parsable data values versus separate data names

An advantage of defining multiple data names for the individual components of a complicated quantity is that there is no ambiguity in resolving the separate components. Hence the Miller indices of a reflection in the list of diffraction measurements are specified in the core dictionary by the group of three data names `_diffrn_refl_index_h`, `_diffrn_refl_index_k` and `_diffrn_refl_index_l`. In principle, a single data name