5.2. STAR File utilities

BY N. SPADACCINI, S. R. HALL AND B. MCMAHON

5.2.1. Introduction

The STAR File, described in Chapter 2.1, has a simple format intended to allow the flexible and extensible representation of data without regard to specific data models. In crystallography and related disciplines, the restricted format chosen for the Crystallographic Information File (CIF, Chapter 2.2) and Crystallographic Binary File (CBF, Chapter 2.3) lends itself to rather flat data models. In particular, the relationships between data items enforced through DDL2 dictionaries in applications such as mmCIF (Chapter 3.6) are essentially equivalent to the data structures and relationships of a relational database. Of course, properly normalized relational tables can represent a hierarchy of structure, although this may not be an efficient representation.

There are other applications, such as the molecular information file (MIF, Chapter 2.4), that make use of additional features of the STAR File, such as multiple-level loop structures, global variable scoping and data-instance encapsulation in save frames. These applications may more efficiently represent certain hierarchical or object-oriented data models.

While particular applications require software tools tailored to their specific purposes, it is helpful to have programs or libraries capable of manipulating arbitrary STAR File data, relying solely on the syntax rules and format of the STAR File and taking no account of the semantic content of the included data.

In this chapter, the stand-alone program *Star_Base* is described in detail. This program uses a local query language to demonstrate the ability to retrieve or re-order data with their associated context. There is also a brief review of *Star.vim* and *StarMarkUp*, applications for editing and browsing STAR Files. The chapter concludes by reviewing a number of object classes and libraries for a variety of STAR and generalized CIF applications: prototypical approaches *OOSTAR* and *CIF++*, *CIFOBJ* and *starlib* used by major macromolecular data repositories, and the document-object model package *StarDOM*.

5.2.2. Data instances and context

In a STAR File, a data item consists of a *value*, which is a simple ASCII character string, and an associated identifier or *data name* which precedes the value, and is invariably an ASCII character string beginning with an underscore character and not including any white-space character, such as `_date` or `_chemical_formula_sum`. (The detailed and formal syntax rules for STAR Files are given in Chapter 2.1.)

5.2.2.1. Single and multiple values

A data item may have a single value, in which case the data name may immediately precede the data value, separated only by white space, *e.g.*

```
_chapter_title  'STAR File utilities'
```

Alternatively, a data item may occur multiple times, in a vector or a list. In such a case, the data identifiers appear in a loop header and the values follow in the order of presentation in the loop header. For the simple example of a tabular array, the loop header plays the role of column header, *e.g.*

```
loop_
   _chapter_number
   _chapter_title
       5.2  'STAR File utilities'
       5.3  'Syntactic utilities for CIF'
```

Here the instances of the data item identified by the data name `_chapter_number` have two values, `5.2` and `5.3`. Likewise the instances of the data item identified by `_chapter_title` have two values.

Note an important point: the example has been chosen to suggest to the reader a tabular relationship between the two data items, and in many STAR File applications such a relationship is intended and perhaps formalized through an external dictionary defining the relationships between these data names. However, *the existence of such a relationship is not mandated by the STAR File syntax*. It is legitimate for a generic STAR application to extract a single data item from such an aggregated loop without making any supposition about its relationship with other data items in the same loop. (It should be emphasized that in practice such physical juxtaposition of data items will almost invariably represent a real relationship, and that most application-specific programming will depend on this fact; but it is not an essential component of STAR in its most abstract form.)

It is also axiomatic that the ordering of the multiple values within a list structure has no intrinsic significance in the STAR paradigm. (Again, specific applications may override this by enforcing an ordering, but this is not fundamental to STAR.)

5.2.2.2. Loop packets and context within lists

Where multiple data names are declared in a loop header, STAR does however enforce the notion of a 'loop packet'. The loop packet is the data structure including all individual data values at a particular iteration through the loop. Hence, in the simple example above, `5.2` and `STAR File utilities` comprise the tuple of values in a single loop packet. For the single level of loop considered so far, the loop packet plays the role of a table row.

For nested loops, the situation is more complex. Consider Fig. 5.2.2.1, which is an example of quantum chemistry basis sets for hydrogen and lithium. (The examples in this chapter are derived from various test applications, and do not represent specific adopted exchange protocols in the selected subject areas.) For each element, a list of basis sets is presented, each containing a set of parameters and a table of functional values. At the outermost level of looping in this example, a loop packet comprises all the data associated with an individual atom type, for example hydrogen. At the next inner level of looping, a loop packet corresponds to an individual basis set (including its embedded table of

Affiliations: N. SPADACCINI, School of Computer Science and Software Engineering, University of Western Australia, 35 Stirling Highway, Crawley, Perth, WA 6009, Australia; SYDNEY R. HALL, School of Biomedical and Chemical Sciences, University of Western Australia, Crawley, Perth, WA 6009, Australia; BRIAN MCMAHON, International Union of Crystallography, 5 Abbey Square, Chester CH1 2HU, England.

references