1.1. How to navigate this volume

BY JAMES R. HESTER AND BRIAN MCMAHON

1.1.1. Introduction

This volume of *International Tables for Crystallography* gives a comprehensive account of the Crystallographic Information Framework (CIF) and its applications and related standards, and caters for a wide range of interested readers. To help in finding the information of most use to different classes of reader, we present the following suggested reading strategies. In all cases, a good starting point will be the overview Chapter 2.1. The 'Executive summary' introducing that chapter (Section 2.1.1) is a concise account of essential information about CIF that provides a convenient *aide-memoire* for established users. The remainder of Chapter 2.1 provides a more detailed account of the CIF standards that may be helpful to the reader with little previous knowledge of the subject.

1.1.2. The general reader

The introductory Chapter 1.2 establishes the need for data exchange standards, and relates this to best practice for scientific data management. The history of how the International Union of Crystallography (IUCr) commissioned the CIF project to establish such standards is described in Chapter 8.1.

Chapter 4.1, on the management and use of CIF dictionaries, explains the significance of the data definitions that establish an interoperable ontology. If the general reader is interested in details of the concrete format that is used within crystallography both for data files and dictionaries, Section 2.1.3 is probably sufficient. If further detail on the syntax is desired, it may be found in Chapter 2.2.

Parts 6 and 7 discuss how CIF is used to facilitate publishing of structure report papers and deposition of structures in curated crystallographic databases, respectively.

1.1.3. The structural scientist

The typical researcher who views CIF as a necessary tool for submitting articles or depositing structures in databases might also wish to begin with the introductory Chapter 1.2 to gain a suitable perspective of the benefits of fully-featured data definitions and exchange standards.

Most researchers need not concern themselves unduly with the details of the CIF format, since this is usually catered for by the refinement software they are using, or by post-refinement end-user applications (Chapter 5.6) for editing or visualizing the CIFs they have produced. If they do need to modify CIFs by hand, the specifications in Chapter 2.2 may be consulted.

They may also have limited interest in the dictionary definitions, since again most refinement packages will select the appropriate data names to tag the data that they output. However, if users need to assign values to data items without the benefit of prompts from interactive programs, they may consult individual definitions in the dictionary listings that form Part 4 of this volume; the index of data names at the end of the volume may also be a useful starting point. Where they do need a better understanding of the significance of data names, especially in relation to other data names that might appear in a data file, the explanatory chapters for each dictionary in Part 3 will provide that information.

Part 6 discusses how CIF is used in the publication of chemical (Chapter 6.1) and macromolecular (Chapter 6.2) structures, and in articles describing raw data sets (Chapter 6.3). These chapters are particularly useful for understanding the validation criteria that may be applied to different types of structures submitted for publication.

The handling of structures deposited in chemical (Chapter 7.1) and macromolecular (Chapter 7.2) structural databases will also be of interest.

1.1.4. The software developer

Programmers wishing to write software that handles crystallographic data in CIF format should begin with the discussion of general principles in Chapter 5.1. They should also read the relevant format specification in Chapters 2.2 (CIF versions 1.1 and 2.0) and 2.3 (imgCIF/CBF).

1.1.4.1. Refinement or data processing packages

Many developers of crystallographic software are focused on data processing and analysis problems, and are interested in CIF only as an input/output channel for passing data to and from an application. They may find that one of the standard libraries discussed in Chapter 5.3 will provide all the support needed for this purpose. Otherwise, they may find it helpful to read the description of the CIF application programming interface (Chapter 5.2), either to use directly the reference implementation (although this is not optimized for performance), or to design their own API on similar principles.

Writing native code to output CIF data is relatively straightforward, since the programmer may choose the order in which data may be written, and has few layout constraints. Attention to the details of the format specification should ensure that syntactically correct CIFs are formed. The programmer must of course take care to ensure that the correct data names are used, and that their associated values conform to the restrictions detailed in the relevant dictionary. The dictionary listings in Part 4 should provide enough information for this, though the matching commentary chapter in Part 3 should also be read to ensure correct usage. If the programmer wishes to reduce the burden of conformance to dictionary attribute constraints, it is possible to write routines to validate directly against the dictionaries. In this case, an understanding of the DDL in which the relevant dictionary is written can be gained from Chapter 2.4. To implement any dictionary methods written in the dictionary relational language (dREL), one should also read Chapter 2.5.

If it is wished to archive in a CIF file some data that are not characterized by existing dictionary definitions, new data names may be created, provided they are differentiated from the existing

Affiliations: JAMES R. HESTER, Australian Nuclear Science and Technology Organisation, Locked Bag 2001, Kirrawee DC, NSW 2232, Australia; BRIAN MCMAHON, International Union of Crystallography, 5 Abbey Square, Chester CH1 2HU, UK.

references