CIF APPLICATIONS

Authors of any software that reads, writes or validates CIF data are invited to contribute to this series. Authors should state clearly when submitting a manuscript to a Co-editor that the paper should be included as part of the CIF Applications series. An appropriate series number will be assigned by the Editorial Office.

J. Appl. Cryst. (1998). **31**, 278–281

CIF Applications. VII. *CYCLOPS2*: extending the validation of CIF data names†

HERBERT J. BERNSTEIN[a]* AND SYDNEY R. HALL[b] *at* [a]*Bernstein + Sons, 5 Brewster Lane, Bellport, NY 11713-2803, USA, and* [b]*Crystallographic Centre, University of Western Australia, Nedlands 6009, Australia. E-mail: yaya@bernstein-plus-sons.com*

(*Received 13 March 1997; accepted 13 May 1997*)

Abstract

CYCLOPS2 is a major revision of the program *CYCLOPS* [Hall (1993). *J. Appl. Cryst.* **26**, 482–494] which is used, in conjunction with Crystallographic Information File (CIF) dictionaries, to validate names in an ASCII file. The validated files may contain CIF or non-CIF data, text documents or a program source. *CYCLOPS2* is able to handle both DDL1 and DDL2 dictionaries, the longer mmCIF data names and can accommodate multiple dictionaries. This version is written using the *CIFtbx2* [Hall & Bernstein (1996). *J. Appl. Cryst.* **29**, 598–603] library of Fortran functions and is portable to a variety of platforms.

1. Introduction

This paper is part of the continuing series on Crystallographic Information File (CIF) applications. The first papers in the series described the *CIFtbx* Fortran subroutine library for programmers developing CIF applications (Hall, 1993) and *CYCLOPS*, a program for validating CIF data names (Hall, 1993). *CYCLOPS* is used primarily as a spelling checker of CIF data names in documents and program sources and is an essential tool in self-checking CIF dictionaries for consistent definitions and cross-references.

Since the first version of *CYCLOPS* was written in 1992, CIF definitions have been expanded to include a much richer and broader range of data types, particularly in the field of macromolecular structure. The mmCIF dictionary (Fitzgerald *et al.*, 1996; Bourne *et al.*, 1997) defines the structural parameters used in macromolecular studies and is based on an extended dictionary definition language, referred to as DDL2 (Westbrook & Hall, 1995). DDL2 employs stronger relational attributes than the DDL1 (Hall & Cook, 1995) version of the dictionary language used for the Core and other dictionaries. *CYCLOPS2* has been designed to process dictionaries based on either DDL1 or DDL2. It also accommodates the longer data names often used in mmCIF definitions, whereas the earlier *CYCLOPS* is limited to dictionaries using DDL1 attributes and 32-character names.

CYCLOPS2 has been rewritten using the new *CIFtbx2* library functions (Hall & Bernstein, 1996) to perform the dictionary-reading and CIF-parsing operations.

Because of the rapidly increasing number of CIF definitions (the mmCIF dictionary alone has over 1500 new data names), the validation information output by *CYCLOPS2* has been redesigned to list the data names encountered in the validated file and the dictionary files as three separate categories. The data names which are unrecognized (*i.e.* encountered in the validated file but not in the dictionary) are output first. Second come the names present in both the validated file and the dictionaries and finally, if requested, are listed the names in the entered dictionaries but not present in the validated file. Aliased data names are also included.

A primary impetus for this work was to support one of us (HJB) in the use of mmCIF data-sets derived from Protein Data Bank (Bernstein *et al.*, 1977) entries. Code for conversion of Protein Data Bank entries to mmCIF format was developed in the form of an awk script, *pdb2cif* (Bernstein, Bernstein & Bourne, 1998). *CYCLOPS2* has been used to validate data names in *pdb2cif* and in the CIFs produced by *pdb2cif*.

2. *CYCLOPS2* overview

Files are read and written by *CYCLOPS2* as follows.

File *a*. The input text file to be validated is entered from the standard input device (normally device 5). For Unix operating systems this is the file `stdin`; on other systems *CYCLOPS2* uses the file `STARTEXT`.

File *b*. The input text file, named `STARDICT`, contains either dictionary definitions or a list of dictionary filenames. That is, this file is either a DDL-conformant dictionary, or, if it begins with the characters '`#DICT`', it contains a list of dictionary filenames to be entered. For Unix operating systems, this information may be provided directly on the command line.

File *c*. The validation report is output to the standard output device (normally device 6). For Unix operating systems this is the file `stdout`; on other systems *CYCLOPS2* uses the file `STARCHECK`.

File *d*. Messages are output to the standard error device (normally device 0). For Unix operating systems this is the file `stderr`; on other systems *CYCLOPS2* uses device 6 and the file `STARCHECK`.

The following procedure is used by *CYCLOPS2* to check data names.

† This paper is one of a series on CIF applications. Offprints are available from The Managing Editor, International Union of Crystallography, 5 Abbey Square, Chester CH1 2HU, England. See text of paper for availability of program(s) by e-mail.

(i) Read STARDICT (file *b*) and if the first line contains '#DICT' make a list of the dictionaries to be loaded. On Unix systems, if the command line contains -d dictionary use the command line dictionaries instead of reading STARDICT.

(ii) Load the dictionary definitions and store all data names. Any definition processing errors are reported to the error output file (file *d*).

(iii) Read the text file to be validated (file *a*), parsing it line by line for identifiable data names (*i.e.* text strings starting with the underscore character '_'). Data names are recognized provided the name begins with an underscore and the name is: preceded by one of the characters ⟨*blank*⟩ ⟨*tab*⟩, .([{⟨/\|"':* or the beginning of a line, and followed by one of the characters ⟨*blank*⟩ ⟨*tab*⟩, .)]}⟩/\|"'-=?!;:. or the end of a line. All alphabetic characters are converted to lower case. The scan stops at the comment character, '#', and goes to the next line.

(iv) On encountering a data name in step (iii), search the stored dictionary names for a match. A match is identified in one of three ways. (*a*) If the data name is not preceded by the asterisk character '*' and it does not end with the underscore character '_', then search for an identical match. (*b*) If the data name ends with the underscore character '_', then search for a match in the dictionary where the leading characters in the dictionary name are the same as all the characters in the data name found in the text. For example, the text _atom_site.label_ would match the mmCIF dictionary entry _atom_site.label_alt_id. (*c*) If the data name is preceded by the asterisk character '*', then search for a match in the dictionary where the trailing characters in the dictionary name are the same as all the characters in the data name found in the text. The first match found in the dictionary is accepted. For example, the text *_alt_id would match _atom_site.label_alt_id, or, if that name had not been in the dictionary, _struct_conn.ptnrl_label_alt_id. If one of the searches succeeds, add the line number of the data name to a list attached to the dictionary name. Up to 19 line numbers are retained for each dictionary name (the first ten matches and the last nine).

```
                CYCLOPS Check List
                ------------------


          Dictionary data names  = 2244
          New data names in text =    4
          [1]  Dictionary cif_core.dic 2.0.1 data names =   624
          [2]  Dictionary cif_mm.dic 0.9.0 data names =  1620

Data names NOT in Dictionary                          Line Numbers

_blat1  . . . . . . . . . . . . . . . . . . . . . .     9    11    94    96
                                                      181   183   290   296   302   308
                                                      314
_blat2  . . . . . . . . . . . . . . . . . . . . . .    13    15    98   100
                                                      185   187   287   293   299   305
                                                      311
_dummy_test  . . . . . . . . . . . . . . . . . . .     5     7    90    92
                                                      177   179   201
_rubbish_here . . . . . . . . . . . . . . . . . .     431


[1]  Dictionary cif_core_2.0.1.dic
[2]  Dictionary cif_mm.dic
                                                      Line Numbers

[2] _atom_site.calc_attached_atom  . . . . . . . .    413
[1] = _atom_site_calc_attached_atom                   412
[2] _atom_site.calc_flag  . . . . . . . . . . . .     410
[1] = _atom_site_calc_flag                            409
[2] _atom_site.fract_x  . . . . . . . . . . . . .      38    44    50   390
[1] = _atom_site_fract_x                              389
[2] _atom_site.fract_y  . . . . . . . . . . . . .      39    45    51   394
[1] = _atom_site_fract_y                              393
[2] _atom_site.fract_z  . . . . . . . . . . . . .      40    46    52   398
[1] = _atom_site_fract_z                              397
[2] _atom_site.id  . . . . . . . . . . . . . . . .     37    43    49   386
[1] = _atom_site_label                                385
[2] _atom_site.thermal_displace_type  . . . . . .     406
[1] = _atom_site_thermal_displace_type                405
[2] _atom_site.type_symbol  . . . . . . . . . . .     416   420   424   428
                                                      434   438   442   450   454   458
                                                      462
[1] = _atom_site_type_symbol                          415   419   423   427
                                                      433   437   441   449   453   457
                                                      461
```

Fig. 1. Sample output at the start of a validation output file. Note that the mmCIF dictionary, cif_mm.dic, defines aliases for data names in the core dictionary, cif_core.dic.

(v) If no match is found, the unmatched data name is added to the list of unmatched names, along with the appropriate line number. If a data name has been misspelled it will be caught at this step.

(vi) When the text file has been processed, output the validation report file (file *c*) containing the alphabetically sorted list of unmatched names and line numbers, followed by the sorted list names from all dictionaries that are used within the text. If requested, this is followed by the sorted list of names from all dictionaries that are not used within the text in the file. If a data name has an alias defined in the dictionaries, a warning about the existence of the alias is given. If more than one dictionary has been used, the source dictionary is identified for each data name. Example extracts from a validation output file are shown in Tables 1 and 2. A command line option is provided to suppress all but the list of unmatched names.

In a Unix-like environment, the program may be run as

cyclops [**-i** input_text] [**-o** validation_output] [**-d** dictionary_1 [**-d** dictionary_2 [**-d** dictionary_3 [...]]]] [**-p** priority] [**-v** verbose] [**-s** short].

The value, input_text, of the **-i** option specifies the input text file, file *a*. The value, validation_output, of the **-o** option specifies the validation output file, file *c*. The portions of the command line shown in brackets are optional. Multiple dictionaries may be specified and the relative priority of dictionaries is controlled by the value for the **-p** option: rst, nal or nodup. The default is rst, which implies that the first of duplicate definitions takes priority. The value nal implies that the last of duplicate definitions takes priority. The value of nodup implies that duplicate definitions are a fatal error. The values for the **-v** and **-s** may be t, 1 or y for the option to be selected, or f, 0 or n for the option not to be selected. For example, **cyclops -d** cif_mm.dic **-s** t **-i** text processes the file named text against the dictionary cif_mm.dic producing a short list of names not found in the dictionary. Because a dictionary is specified on the command line, STARDICT (file *b*) is not processed.

3. Error message glossary

In addition to the error messages reported by the *CIFtbx2* library routines when processing dictionaries, *CYCLOPS2* can output the following error messages.

Data name in text is > NUMCHAR chars <*string*>

A nonfatal warning issued if the length of a data name in the validated file exceeds the preset value of NUMCHAR. Processing continues with a truncated name. If needed, the

```
[1] _symmetry_cell_setting . . . . . 319
[2] = _symmetry.cell_setting 320
[1] _symmetry_space_group_name_H-M . . . . . 323
[2] = _symmetry.space_group_name_H-M 324
[1] _symmetry_space_group_name_Hall . . . . . 327   445
[2] = _symmetry.space_group_name_Hall 328   446


[1]  Dictionary cif_core_2.0.1.dic
[2]  Dictionary cif_mm.dic
                                        Names Not Referenced

[2] _atom_site.aniso_B[1][1]
[2] _atom_site.aniso_B[1][1]_esd
[2] _atom_site.aniso_B[1][2]
[2] _atom_site.aniso_B[1][2]_esd

 [... portion of output omitted ...]

[2] _atom_site.aniso_U[2][3]_esd
[2] _atom_site.aniso_U[3][3]
[2] _atom_site.aniso_U[3][3]_esd
[2] _atom_site.attached_hydrogens
[1] = _atom_site_attached_hydrogens
[2] _atom_site.auth_asym_id
[2] _atom_site.auth_atom_id
[2] _atom_site.auth_comp_id
[2] _atom_site.auth_seq_id
[2] _atom_site.B_equiv_geom_mean
[1] = _atom_site_B_equiv_geom_mean
[2] _atom_site.B_equiv_geom_mean_esd
[2] _atom_site.B_iso_or_equiv
[1] = _atom_site_B_iso_or_equiv
[2] _atom_site.B_iso_or_equiv_esd
[2] _atom_site.Cartn_x
[1] = _atom_site_Cartn_x
[2] _atom_site.Cartn_x_esd
[2] _atom_site.Cartn_y
[1] = _atom_site_Cartn_y
[2] _atom_site.Cartn_y_esd

 [... remainder of output omitted ...]
```

Fig. 2. Sample output later in a validation output file, showing the transition to unreferenced data names.

value of NUMCHAR may be changed in ciftbx.sys, and ciftbx.f recompiled.

Dictionary list empty

The file STARDICT does not contain either definitions or a list of dictionary filenames and none were specified on the command line. See file *b* description above.

Too many dictionaries

More than 99 dictionaries have been loaded. This is probably due to an error in constructing the file STARDICT. If many small dictionaries must be loaded, they should be merged into composite files until there are less than 100 dictionaries.

<dictionary name> **not found**

A dictionary file listed in STARDICT or on the command line could not be opened.

Data name table exceeded (current max is NUMDICT)

The combined number of data names in the dictionaries and the text is greater than the parameter NUMDICT defined in ciftbx.sys. Recompile with a larger value of NUMDICT.

4. Implementation

CYCLOPS2 is a Fortran program. The dialect of Fortran used is the same modestly extended variant of Fortran77 used in *CIFtbx2* which is accepted on most current platforms. In particular, ENDDO and INCLUDE statements are used. Some variable names are longer than six characters and some variable names contain an underscore. Translation to pure Fortran77 can be done but would detract from the readability of the code.

The command line interface depends on having the routine iargc, which returns a count of the command line arguments, and getarg, which returns a string for the selected command line argument. The routines are standard under Unix operating systems and are available for many non-Unix platforms, but replacements are provided in the comments of *CYCLOPS2* for systems which do not provide command line access.

The filename defaults used by *CYCLOPS2* can be overridden under Unix operating systems by setting environment variables CYCLOPS_INPUT_TEXT, CYCLOPS_VALIDATION_OUT and CYCLOPS_CHECK_DICTIONARY. This feature is implemented by calling the routine getenv. For systems where this routine is not available, a simple replacement is needed.

The original *CYCLOPS* performed its own simple dictionary parsing. *CYCLOPS2* uses *CIFtbx2* to parse both DDL- and DDL2-based dictionaries. Rudimentary code for the handling of global sections was added to *CIFtbx2* to allow *CYCLOPS2* to accept all currently released dictionaries. The only areas for which *CYCLOPS2* is responsible for the dictionary processing is to load the dictionaries in the order specified by the command line option -p and to sort names into alphabetic order within dictionaries.

5. Distribution

The latest version of this software is available as part of the *CIFtbx2* distribution at any of the following WWW servers:

http://ndbserver.rutgers.edu/NDB/mmcif/software
http://www.ebi.ac.uk/NDB/mmcif/software
http://ndbserver.nibh.go.jp/NDB/mmcif/software
http://www.crystal.uwa.edu.au/cc_star.html
http://www.iucr.org/iucr-top/cif/software/ciftbx

or from the anonymous ftp site ftp.crystal.uwa.edu.au (130.95.232.12).

CYCLOPS2 was distributed with *CIFtbx2* within the file ciftbx.cshar through release 2.5.2 of *CIFtbx2* and release 2.1.2 of *CYCLOPS2*. Starting with the distribution of release 2.5.3 of *CIFtbx2* and release 2.1.3 of *CYCLOPS2*, *CYCLOPS2* is distributed as the file cyclops.cshar, and the *CYCLOPS2* files have been removed from ciftbx.cshar. The structure of the cshar files permits automatic unpacking in Unix systems having the C shell, csh, but, unlike the more commonly used shar format, also admits unpacking with a text editor. A cyclops.shar version is also available.

A number of dictionaries are available for use with the validation. In general, the lastest versions can be traced by starting at the IUCr CIF web page (http://www.iucr.org/cif/).

As of this writing, the direct URLs for the major CIF dictionaries in crystallography are, for the revised CIF dictionary (Core 1997): ftp://ftp.iucr.org/pub/cif_core_2.0.1; for the DDL (dictionary definition language) dictionary: ftp://ftp.iucr.org/pub/ddldic.c95; for the enhanced DDL (DDL2) used in the mmCIF dictionary: ftp://ftp.iucr.org/pub/ddl2.c96; for the mmCIF dictionary: http://ndbserver.rutgers.edu/NDB/mmcif/dictionary/cif_mm.dic; for the powder dictionary: ftp://ftp.iucr.org/pub/cifdic.p96. Please note that the direct URLs may change or be replaced by URLs for later versions. The IUCr CIF web page should be consulted for current URLs. For further information, send e-mail to syd@crystal.uwa.edu or yaya@bernstein-plus-sons.com.

We thank Frances C. Bernstein, Daniel J. Bernstein and Anne Hall for helpful comments and suggestions. Please note that Unix is a trademark of X/Open. This work was supported in part by the IUCr (for HJB).

References

Bernstein, H. J., Bernstein, F. C. & Bourne, P. E. (1998). *J. Appl. Cryst.* **31**, 282–295

Bernstein, F. C., Koetzle, T. F., Williams, G. J. B., Meyer, E. F. Jr, Brice, M. D., Rodgers, J. R., Kennard, O., Shimanouchi, T. & Tasumi, M. (1977). *J. Mol. Biol.* **112**, 535–543.

Bourne, P. E., Berman, H. M., McMahon, B., Watenpaugh, K. D., Westbrook, J. & Fitzgerald, P. M. D. (1997). *Methods Enzymol.* **277**, 571–590.

Fitzgerald, P. M., Berman, H. M., Bourne, P. E., McMahon, B., Watenpaugh, K. D. & Westbrook, J. (1996). 17th IUCr Congress and General Assembly, Seattle, Washington, USA, 8–17 August 1996, Abstract E1226.

Hall, S. R. (1993). *J. Appl. Cryst.* **26**, 482–494.

Hall, S. R. & Bernstein, H. J. (1996). *J. Appl. Cryst.* **29**, 598–603.

Hall, S. R. & Cook, A. P. C. (1995). *J. Chem. Inf. Comput. Sci.* **35**, 819–825.

Westbrook, J. & Hall, S. R. (1995). *A Dictionary Description Language for Macromolecular Structure, Draft DDL V 2.1.0*, IUCr COMCIFS, Chester, England.