International Tables for Crystallography (2006). Vol. G, Section 5.3.3.4, pp. 506–507.

the selected data name and hyperlinks to definitions of other data names referred to. Additionally, there is a small text-entry box allowing a specific definition to be retrieved and an ‘Index’ button to list all available definitions.

5.3.3.3.2. *Options*

As already mentioned, the user could modify the detailed mode of operation of the program. Any or all of the ‘initial’, ‘dictionary’ or ‘other’ checks could disabled.

The ‘dictionary’ checks could be modified by the user through the ‘Options’ button of the main control window. The CIF dictionary for validation could be specified; the dictionary itself had to be translated from a source file in DDL format to a Python data structure.

The types of dictionary-based validation supported by the program were:

(i) *List Status* (checking whether a data value should be included in a looped list),

(ii) *Limited Enumeration Options* (checking that a data value is one of the permitted codes where such a constraint exists),

(iii) *Incorrect Enumeration Case* [a special case of (ii), where a data value matches a permitted code except for incorrect alphanumeric case],

(iv) *Enumeration Range* (the data value falls outside the range permitted),

(v) *Value Type (numb or char)* (the data value has the wrong type),

(vi) *List Link Parent* (a data item is present within the data block, but its mandated parent item is not – for example, the data item **`_atom_site_aniso_label`** should not be present without its parent data item **`_atom_site_label`**),

(vii) *List Reference* (the required data name used to reference the loop in which the current data name appears is missing),

(viii) *Esd Allowable* (a data value appears to have a standard uncertainty value where one is not expected).

The user could also supply the program with a list of data names that do not appear in the validation dictionary but for which no warning message should be raised. The program normally flagged such nonstandard data names as possible errors and suggested the possible form of a standard data name that might have been intended. This was useful in catching misspellings of additional data items entered by hand.

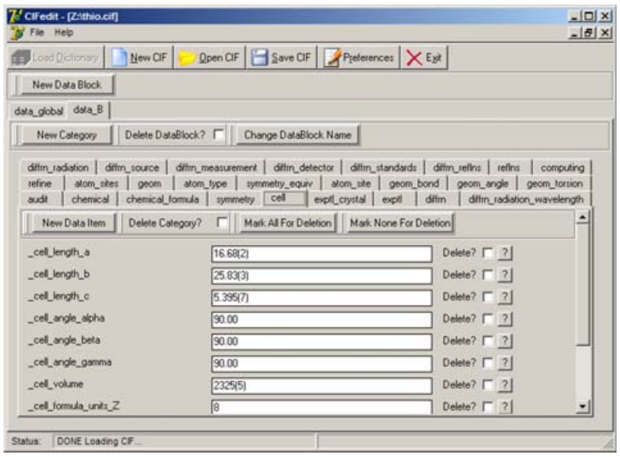
The program could also be run in a batch mode when the objective was to work through a large volume of CIF data and identify the data blocks that require attention. This mode of operation is particularly useful in databases or publishing houses. In this mode, input is from a named file or from the standard input channel; output is written to standard output or redirected to a results file. The operation of the program may be controlled by the application of various command-line flags.

5.3.3.4. Platform-specific editors

As well as the tools described earlier in this section, which are designed to run under a variety of common operating systems, there are some applications restricted to users of particular types of computer. Here we mention two that run in the popular Microsoft Windows environment on personal computers.

5.3.3.4.1. *beCIF*

The Windows program *beCIF* (Brown *et al.*, 2004) is still in prototype. It is a DDL1-dictionary-driven CIF manipulation tool that does not require detailed knowledge of CIF or dictionary structures. It provides a rather different view of the contents of a CIF from the applications discussed above through an interface that will be familiar to users of Microsoft Windows applications. When the application is opened, the user is prompted to provide the location of a CIF dictionary (at any one time, only a single dictionary file may be loaded). This dictionary is loaded into memory and used to validate CIFs upon input. As a data file is read, discrepancies from the types and value ranges permitted by the dictionary are listed in an information window.



_cell_length_a	16.68(2)	Delete? □	?
_cell_length_b	25.83(3)	Delete? □	?
_cell_length_c	5.395(7)	Delete? □	?
_cell_angle_alpha	90.00	Delete? □	?
_cell_angle_beta	90.00	Delete? □	?
_cell_angle_gamma	90.00	Delete? □	?
_cell_volume	2325(5)	Delete? □	?
_cell_formula_units_Z	8	Delete? □	?

Status: DONE Loading CIF...

Fig. 5.3.3.11. A category view in the *beCIF* editor of a CIF with navigation by tabs.

The file contents are presented in a number of panels, one per dictionary category, between which the user may navigate by selecting the tab with the desired category name (Fig. 5.3.3.11).

At the highest level, tabs allow the user to choose the data block of interest. Buttons are provided to delete a data block entirely, to rename it or to create a new data block.

Within each data block, the user may add new categories. Again, to help the novice user, when the button ‘New Category’ is selected, a list of only those categories described in the current dictionary but absent from the current data block is presented to the user. Each category present in the data file is accessed through its own tabbed display panel.

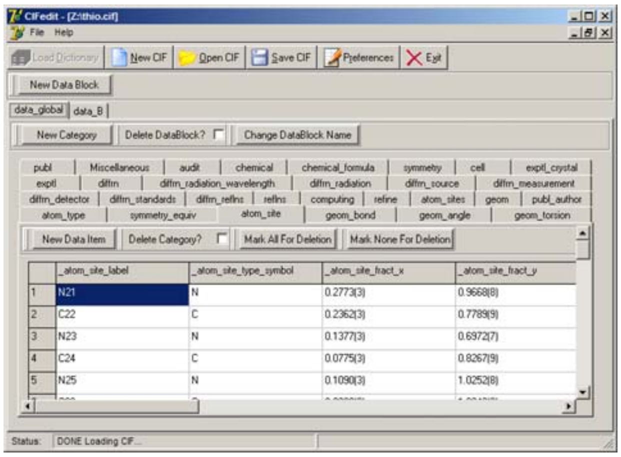
Where the category contains non-looped data items, values may be edited within individual text widgets; data items may be removed by selecting the adjacent check box; or new data items may be added by selecting the ‘New Data Item’ button to create a dialogue box offering a choice of the remaining data items in the dictionary category. Against each data item a button provides access to a pop-up window containing the relevant dictionary definition.

For a category with looped data, the contents are displayed in a spreadsheet-style representation, with columns headed by the matching data name and rows numbered for convenience (Fig. 5.3.3.12).

The changes requested to the CIF are only effected when the user selects the ‘Save CIF’ button. Unlike many other of the CIF editors previously discussed, this program does not make any effort to retain the initial ordering of the input data, nor does it preserve comments. The edited CIF may therefore be superficially very different from the input file; however, the only significant differences in content will be those introduced through use of the editing functions within the application.

5.3.3.4.2. *printCIF for Word*

The tools described so far emphasize the data content of a CIF. *printCIF for Word* (Westrip, 2004), on the other hand, was commissioned to help prospective authors of structure reports in the



	_atom_site_label	_atom_site_type_symbol	_atom_site_fract_x	_atom_site_fract_y
1	N21	N	0.2773(3)	0.9668(8)
2	C22	C	0.2362(3)	0.7789(9)
3	N23	N	0.1377(3)	0.6972(7)
4	C24	C	0.0775(3)	0.8267(9)
5	N25	N	0.1090(3)	1.0252(8)

Status: DONE Loading CIF...

Fig. 5.3.3.12. Representation by the *beCIF* editor of looped data within a category (here ATOM_SITE) in spreadsheet style.

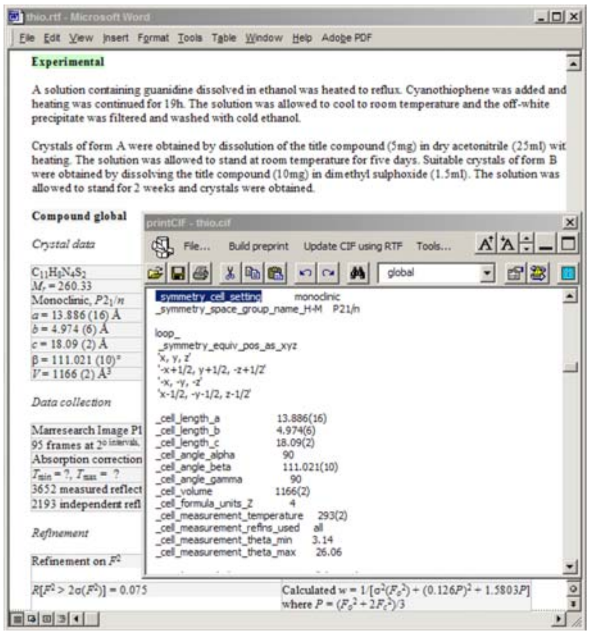
IUCr journals to visualize and prepare for publication complete papers submitted in CIF format. Chapter 5.7 describes the workflow and processing of such submissions. Here is given a brief description of the use of the *printCIF* software from an author's viewpoint.

This application also differs from others discussed in this chapter in that it is rather specific to a particular program environment, being written as Visual Basic macros embedded in a Microsoft *Word* template document. Efforts are under way to provide versions that can run with other word processors. Nevertheless, *Word* is currently sufficiently widespread that the utility is likely to be of use to a large community.

Typically the author begins by double-clicking on the icon associated with the printcif.dot template file. The initial macros are loaded and the author is prompted to provide the location of a CIF. As the CIF is imported into the application, the data items that will be used in the publication are extracted and converted into a rich-text format (RTF) representation. For extended text fields, this RTF content may be edited directly in the word-processing environment; this makes it easy for authors to compose and edit continuous text in a familiar way. Numeric and brief textual data items from the CIF are processed and presented in read-only fields in the manner in which they will appear in the journal, often as entries in a table or as a list of brief experimental details. These fields may not be edited within the RTF representation; if it is necessary to change these, the author must modify the data value in the CIF itself. To assist the author, the contents of the CIF are opened in a text-editor window alongside the formatted representation. The CIF and RTF representations are linked; if the author selects text in the RTF window, the corresponding CIF data item is highlighted within the text-editor window (Fig. 5.3.3.13).

The advantages to the author of editing in RTF format are that existing text may be cut and pasted from other applications, and formatting features, such as subscript or superscript text, Greek letters and other special symbols, may be entered through the word-processor's menu-driven interface, rather than by use of the rather unmemorable ASCII codings used in CIF.

The major disadvantage is the need to recognize that two versions of the file, both editable, are accessible at the same time; and care must therefore be taken to ensure that conflicting changes are not made, and that the author is aware of which version is currently the master. The function 'Update CIF using RTF' (in the toolbar of the CIF editing window) will reimport into the CIF all the editable content from the RTF window, replacing any existing data items.


thio.rtf - Microsoft Word
File Edit View Insert Format Tools Table Window Help Adobe PDF
printCIF - thio.cif
File...
Build preprint
Update CIF using RTF
Tools...
global

Experimental

A solution containing guanidine dissolved in ethanol was heated to reflux. Cyanothiophene was added and heating was continued for 19h. The solution was allowed to cool to room temperature and the off-white precipitate was filtered and washed with cold ethanol.

Crystals of form A were obtained by dissolution of the title compound (5mg) in dry acetonitrile (25ml) wit heating. The solution was allowed to stand at room temperature for five days. Suitable crystals of form B were obtained by dissolving the title compound (10mg) in dimethyl sulphoxide (1.5ml). The solution was allowed to stand for 2 weeks and crystals were obtained.

Compound global

Crystal data

$C_{11}H_8N_4S_2$
$M_r = 260.33$
Monoclinic, $P2_1/n$
$a = 13.886$ (16) Å
$b = 4.974$ (6) Å
$c = 18.09$ (2) Å
$\beta = 111.021$ (10)°
$V = 1166$ (2) Å^3

Data collection

Marresearch Image Pl
95 frames at 2° intervals,
Absorption correction
$T_{min} = ?$, $T_{max} = ?$
3652 measured reflect
2193 independent refl

Refinement

Refinement on F^2
$R[F^2 > 2\sigma(F^2)] = 0.075$

Calculated $w = 1/[\sigma^2(F_o^2) + (0.126P)^2 + 1.5803P]$
where $P = (F_o^2 + 2F_c^2)/3$

```
_symmetry_cell_setting          monoclinic
_symmetry_space_group_name_H-M   P21/n

loop_
 _symmetry_equiv_pos_as_xyz
 'x, y, z'
 '-x+1/2, y+1/2, -z+1/2'
 '-x, -y, -z'
 'x-1/2, -y-1/2, z-1/2'

_cell_length_a                  13.886(16)
_cell_length_b                  4.974(6)
_cell_length_c                  18.09(2)
_cell_angle_alpha                 90
_cell_angle_beta                  111.021(10)
_cell_angle_gamma                  90
_cell_volume                    1166(2)
_cell_formula_units_Z              4
_cell_measurement_temperature     293(2)
_cell_measurement_reflns_used     all
_cell_measurement_theta_min       3.14
_cell_measurement_theta_max        26.06
```

Fig. 5.3.3.13. The dual RTF/CIF editing windows in the *printCIF for Word* application. In this example, the author has selected the word 'Monoclinic' in the read-only table of crystal data; the corresponding CIF data item **`_symmetry_cell_setting`** is highlighted in the CIF window, where it may be edited.

The complementary function, 'Build preprint', creates a fresh copy of the preprint representation of the document in RTF format.

A number of options are available to modify the preprint that is generated (for example, by printing a complete list of the geometry included in the CIF rather than just the items flagged for publication; or listing the atomic coordinate data). The general style is that of *Acta Crystallographica Section C* and *Section E*; nevertheless, the application may be useful to users who do not intend to submit to these journals but who wish to produce an attractive representation of the content of their CIFs.

Utilities are provided to create tables in the RTF environment suitable for embedding in the CIF, to browse the contents of the CIF core dictionary and to validate the syntax of the CIF. The application is not dictionary-driven, however, and does not carry out detailed consistency checks. It is therefore best considered as an aid to publication, to be used alongside data-centric editors and validation tools such as *enCIFer*.

A particularly useful self-documenting feature of *printCIF for Word* is that the User Guide is automatically opened when the application is started, before a CIF is loaded.

5.3.4. Data-name validation

In a CIF, a data name (a character token beginning with an underscore character, _) is an essential handle on an item of data within a data block. Equipped only with knowledge of the data names appearing in a CIF, a user may extract, reorder or query the information content of the file. Such manipulations require no prior knowledge of the semantic content of the data. However, for most practical applications it is important to know the meaning attached to data names, and CIF dictionaries provide the mechanism for associating a data name with its intended meaning for an application. It is therefore valuable to be able to check whether data names in a CIF match those defined in a dictionary file. It is also valuable to check the consistency of the data names listed in the dictionary file itself; since this will be used by external applications to validate data names, it is essential that it be internally consistent.

references