International Tables for Crystallography (2006). Vol. G, Section 5.3.3, pp. 501–507.

5.3.2.1.3. *Limitations of vcif*

Because the program is testing certain properties of character strings within logical lines of a file, it stores a line at a time for further internal processing. If a line contains a null character (an ASCII character with integer value zero), this will be taken as the termination of the string currently being processed, according to the normal conventions in the C programming language for marking the end of a text string. In this case, subsequent error messages may not reflect the real problem. The null character, of course, is not allowed in a CIF.

vcif also interprets syntax rules literally, so a misplaced semicolon might mean that a large section of the file is regarded as a text field and too many or too few error messages are generated. This can make a correct interpretation of the causative errors difficult for a novice user.

5.3.3. Editors with graphical user interfaces

A useful class of editing tool is the graphical editor, where different types of access can be provided through icons, windows or frames, menus and other graphical representations. The availability of standard instructions through drop-down menus makes such tools particularly suitable for users who are not expert on the fine details of the file format. The ability within the program to restrict access to particular regions of the file makes it easier to modify the contents of a CIF without breaking the syntax rules. A small but growing number of such editors are becoming available, such as those described here.

5.3.3.1. *enCIFer*

The program *enCIFer* (Allen *et al.*, 2004) has been developed as a graphical utility designed to indicate clearly to a novice user where errors are present in a CIF, to permit interactive editing and revalidation of the file, and to allow visualization of three-dimensional structures described in the file. In its early releases, it was targeted at the community of small-molecule crystallographers interested in publishing structures or depositing them directly in a structure database. Version 1.0 depended on a compiled version of the CIF core dictionary, but subsequent versions allow external CIF dictionaries to be imported. At the time of publication (2005), development is concentrating on support for DDL1 dictionaries.

Given its target user base, the purpose of the program is to permit the following operations within single- or multi-block CIFs:

(i) Location and reporting of syntax and/or format violations using the current CIF dictionary.

(ii) Correction of these syntax and/or format violations.

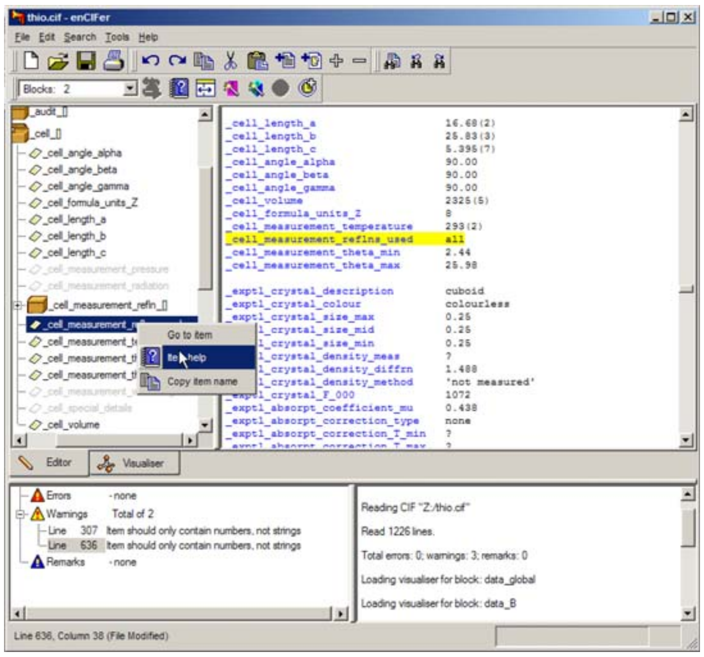
(iii) Editing of existing individual data items or looped data items.

(iv) Addition of new individual data items or looped data items.

(v) Addition of some standard additional information *via* two data-entry utilities prompting the user for required input (‘wizards’): the *publication wizard*, for entering the basic bibliographic information required by most journals and databases that accept CIFs for publication or deposition; and the *chemical and crystal data wizard*, for entering chemical and physical property information in a CIF for publication in a journal or deposition in a database.

(vi) Visualization of the structure(s) in the CIF.

In all cases where data are edited or added, *enCIFer* can be used to check the format integrity of the amended file.



```
_cell_length_a                    16.68(2)
_cell_length_b                    25.83(3)
_cell_length_c                    5.395(7)
_cell_angle_alpha                 90.00
_cell_angle_beta                  90.00
_cell_angle_gamma                 90.00
_cell_volume                      2325(5)
_cell_formula_units_Z             8
_cell_measurement_temperature     293(2)
_cell_measurement_reflns_used     all
_cell_measurement_theta_min       2.44
_cell_measurement_theta_max       25.98

_exptl_crystal_description        cuboid
_exptl_crystal_colour             colourless
_exptl_crystal_size_max           0.25
l_crystal_size_mid                0.25
l_crystal_size_min                0.25
l_crystal_density_meas            ?
l_crystal_density_diffrn          1.488
l_crystal_density_method          'not measured'
l_crystal_F_000                   1072
_exptl_absorpt_coefficient_mu     0.438
_exptl_absorpt_correction_type    none
_exptl_absorpt_correction_T_min   ?
```

```
Errors      - none
Warnings    Total of 2
  Line  307  Item should only contain numbers, not strings
  Line  636  Item should only contain numbers, not strings
Remarks     - none
```

```
Reading CIF "Z:/thio.cif"
Read 1226 lines.
Total errors: 0; warnings: 3; remarks: 0
Loading visualiser for block: data_global
Loading visualiser for block: data_B
```

```
Line 636, Column 38 (File Modified)
```

Fig. 5.3.3.1. The *enCIFer* graphical user interface.

5.3.3.1.1. *The main graphical window*

Fig. 5.3.3.1 is an example of the use of *enCIFer* to read and modify a CIF. The figure shows the components of the main window after a file has been opened. Beneath the standard toolbar that provides access to operating-system utilities and to the main functions of the program itself is a task bar (here split over two lines) providing rapid access to a subset of the program’s features. Under this are two large panes. The pane on the right is the editing window, where the content of the CIF is displayed and may be modified. The left-hand pane is a user-selectable view by category of the data names stored in the CIF dictionary against which the file is to be validated. At the bottom are two smaller panes. The one on the right logs the session activities and displays informational messages. The left-hand pane lists errors and warning notices generated by the validation system. Errors are labelled by line number, and selection of a specific message (by a mouse double-click) scrolls the content of the main text-editing window to that line number.

Tabs in the middle of the display allow the user to switch rapidly between the editing mode and a visualization of the three-dimensional structures described in the CIF.

These components are described more fully below, followed by a description of the other windows that may be created by a user: the help viewer, the loop editor and the data-entry wizards.

5.3.3.1.2. *The interface toolbar*

This toolbar provides menus labelled ‘File’, ‘Edit’, ‘Search’, ‘Tools’ and ‘Help’ that provide the expected functionality of graphical interfaces: the ability to open, close and save files, store a list of recently accessed files, spawn help and other windows, allow searching for strings within the document, allow the user to modify aspects of the behaviour or the look and feel of the program, and provide entry points for specific modes of operation. The most useful of these utilities can also be accessed from icons on the task bar. They are discussed in more detail in the following section.

This main menu is structured in a way familar to users of popular applications designed for the Microsoft Windows operating system, although the *enCIFer* program runs on a variety of different operating systems and machine hardware platforms. Nevertheless, the use of a common menu style makes the initial use of the program much easier for novice users and allows the program to be effectively used without detailed study of its documentation.

5.3.3.1.3. *The task bar*

The task bar allows rapid one-click access to the standard operations of creating a new document, opening, saving or printing the contents of the current file, copying, cutting and pasting text, searching for specific text within the document, and undoing or redoing previous edits.

Two buttons allow insertion of complete text files. One allows the user to select any file from local or network-mounted file systems. The other imports a specific file (the location of which may be specified by the user through the 'Preferences...' selection of the main 'Edit' menu). While this specific file may contain anything, it is intended to be a template CIF that a user will tailor to meet their own requirements. The default provided with the software is a standard template distributed by the IUCr for use in submitting articles to *Acta Crystallographica*. In either case, the file is imported at the current editing location and is not subject to validation upon input; the user must manually revalidate the file after import.

An icon on the task bar allows the user to run a validation procedure. This icon will be dimmed (indicating that the validation procedure may *not* be run) unless the user has modified the contents of the CIF. Other icons on the task bar behave in the same manner, allowing the procedures with which they are associated to be executed only under appropriate circumstances. Thus, for example, the looped list editor is not invoked unless the user clicks within the reserved word `loop_` in a list header.

Similarly, the 'help' icon in the task bar is dimmed unless the user has selected a data name in the CIF; when this is done, the icon is activated and clicking on it launches a help window containing the CIF dictionary definition of the data item.

The task bar also contains a drop-down menu listing all the data-block names in the current file. When the user selects one of the data-block names, the edit cursor is positioned at the head of the matching data block in the edit window. This is a rapid and efficient way of navigating within large and complex files.

The other buttons provided on the task bar allow the user to: reduce or increase the font size in the editing window; create a new looped list within the loop-editing window; invoke the publication and data-entry wizards; and hide or reveal the dictionary browse window pane.

Users may modify the appearance of the task bar to retain or conceal subsets of these icons, depending on which they find most useful.

5.3.3.1.4. *The main edit pane*

The main edit pane is a text-editing area where the user may directly modify the content of a CIF. Colours and font styles are used to indicate different syntactic elements. The details of the colours and styles may be modified to suit the user.

For the novice user, this is perhaps the most immediately helpful feature offered by this program. When a trailing semicolon is inadvertently lost from an extended text field, typical sequential parsers may interpret succeeding tokens as part of the quoted text and produce misleading error reports. Within the *enCIFer* edit window, all such text is marked up in a specific colour (green by default) so that the fault is much more obvious to the human eye and its source much easier to locate.

Two other typographic cues are used to help the user to trace errors, or to ensure that certain text has been input correctly. Subscripts and superscripts are represented in a smaller typeface (and in a different colour) so that missing delimiter characters are again obvious to the eye. Secondly, some special characters in the conventional CIF encoding (such as Greek letters) are displayed in an appropriate symbol font when the file is first loaded, so that for example the input string `\a` is rendered as $\backslash\alpha$. Note that the backslash character is retained, and that the symbol character is not generated as new text is input or edited. This scheme therefore has some potential for confusion, but is nevertheless helpful in checking that less obvious special codes have been entered correctly.

The user is free to enter arbitrary text in this pane, possibly breaking CIF syntax rules in the process. Only when the revalidation process is manually invoked will the file be rescanned and any errors reported.

5.3.3.1.5. *The dictionary browse pane*

The upper left-hand pane in Fig. 5.3.3.1 illustrates the dictionary browser, an optional graphical view of the contents of the CIF dictionary against which the file is being validated. (The presence or absence of this pane is toggled from an icon in the task bar.) Box icons represent the contents of categories, and the tree of category containers may be expanded or collapsed as desired to show individual items within categories.

A dictionary view is generated for each separate data block in the CIF. Within the dictionary view of an individual data block, those data items present in the data block are shown in bold; other items defined in the dictionary but absent from the current data block appear in a lighter colour.

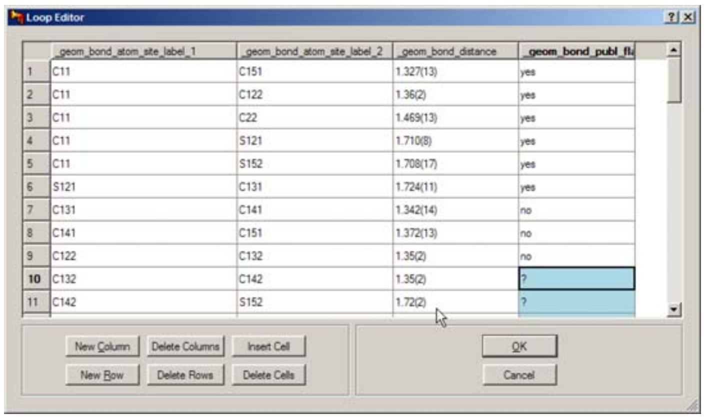
Within the dictionary browse pane, a user may select (with a click of the appropriate mouse button) a menu of three options which depend on whether the data name is present or absent in the data block. If present, one option positions the cursor in the editing window at the location of the selected data item. If the item is absent from the data block, the user is given the option to paste the data name into the editing window at the current insertion point. The other options (in both cases) are to copy the data name to the clipboard or to open the help window with the CIF dictionary definition of the selected item.

5.3.3.1.6. *The error notification pane and logging area*

The lower left-hand pane of Fig. 5.3.3.1 illustrates typical error notices generated by the parser when the validation process is invoked. At present, the classification of the severity of errors is guided by the editorial requirements of databases and journals, and does not necessarily match the formal errors dictated by the CIF specification. It is likely that this will change in future releases as validation is driven increasingly by the dictionaries rather than by hard-coded subroutines.

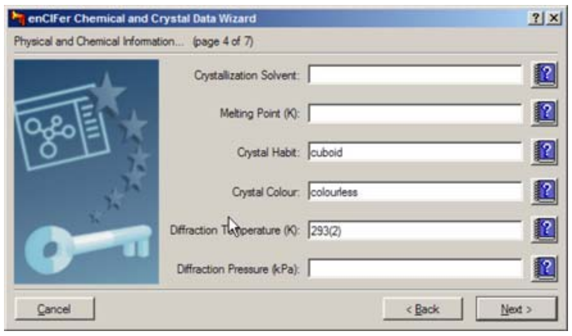
A convenient feature is that double-clicking on the line number in the error report relocates the cursor to that line in the editing pane. At present, error messages are listed by line only – they are not grouped by data block.

The user has a small number of options to control error notification. The choice of the maximum number of consecutive error lines to permit before error checking is abandoned is a useful way, especially for novices, to reduce the amount of output generated by severe syntax errors and to focus on repairing individual errors. The user may also specify a file that contains a set of CIF data names which are considered *mandatory* components of a particular file. Absence of any of these items from the current data



	_geom_bond_atom_site_label_1	_geom_bond_atom_site_label_2	_geom_bond_distance	_geom_bond_publ_fl
1	C11	C151	1.327(13)	yes
2	C11	C122	1.36(2)	yes
3	C11	C22	1.469(13)	yes
4	C11	S121	1.710(8)	yes
5	C11	S152	1.708(17)	yes
6	S121	C131	1.724(11)	yes
7	C131	C141	1.342(14)	no
8	C141	C151	1.372(13)	no
9	C122	C132	1.35(2)	no
10	C132	C142	1.35(2)	?
11	C142	S152	1.72(2)	?

Fig. 5.3.3.2. The *enCIFer* loop editor.



Physical and Chemical Information... (page 4 of 7)

Crystallization Solvent:	
Melting Point (K):	
Crystal Habit:	cuboid
Crystal Colour:	colourless
Diffraction Temperature (K):	293(2)
Diffraction Pressure (kPa):	

Fig. 5.3.3.4. The *enCIFer* chemical and crystal data wizard.

block is flagged as an error. The program log in the lower right-hand part of the program window records the history of the user's interactions with the file during the current editing session.

Information is written to the status bar (the lower margin of the window) to indicate the location by line and column number of the editing cursor.

5.3.3.1.7. *The loop editor*

The program has a useful spreadsheet-style editor for looped lists (Fig. 5.3.3.2). A particular benefit of this style of display is that the spreadsheet cells are arranged in a rectangular grid, so that visual scans can often detect deviations from a pattern of values within a column, thus making it easy to identify placement errors where values have been omitted or inadvertently conjoined. Such errors are not always obvious by direct visual inspection of a CIF, where the layout of a looped list need not follow any regular pattern.

The buttons to add or delete columns allow for the straightforward addition or deletion of data items from the loop. If the user selects the 'New Column' button, a small pop-up window helpfully provides a view of the associated dictionary (in the same hierarchical category-based tree view of the dictionary browser pane) to help the user select the required new data name. The 'Insert Cell' and 'Delete Cells' buttons are convenient tools for the realignment of rows and columns where values have been omitted or misplaced.

The loop editor is invoked from one of two buttons in the task bar, allowing either the creation of a new looped list or the modification of an existing one. As with the application as a whole, there is no dynamic validation of input; the new list must be saved and the entire CIF then manually revalidated.

5.3.3.1.8. *The publication and chemical and crystal data wizards*

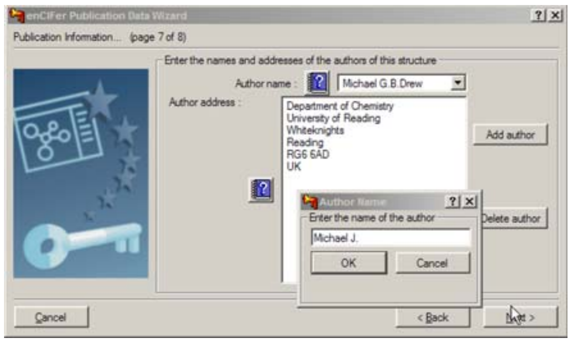
The user may invoke data-entry 'wizards', subordinate programs that prompt for particular data items useful for the publication of a crystal structure report or for the deposition of a crystal structure in a database. This is the kind of information that might be requested in the *Notes for authors* for a journal, and it is helpful if the information is routinely requested from inexperienced authors during normal use of the software. The data-entry tools are known as 'wizards' because they will utilize information already in the file.

Hence, as shown in Fig. 5.3.3.3, details of an article's contact author are retrieved from the CIF and used to seed a list of contributing authors. As the address for each author is entered, the program makes each new address available as a stored record for easier input of additional information.

Fig. 5.3.3.4 demonstrates the same approach to encouraging authors to supplement information already in the CIF with related chemical (or crystal) data not usually provided by the CIF generators embedded in crystallographic structure determination programs.

5.3.3.1.9. *The visualization window*

A final useful feature of *enCIFer* is its ability to visualize the three-dimensional structure of molecules described in the data blocks of a CIF. Fig. 5.3.3.5 demonstrates crystal packing with



Publication Information... (page 7 of 8)

Enter the names and addresses of the authors of this structure

Author name : Michael G.B.Drew

Author address : Department of Chemistry
University of Reading
Whiteknights
Reading
RG6 6AD
UK

Enter the name of the author

Michael J.

Fig. 5.3.3.3. The *enCIFer* publication data wizard. Information about the title and authors of an article to be submitted for publication is requested through a sequence of linked dialogue boxes.

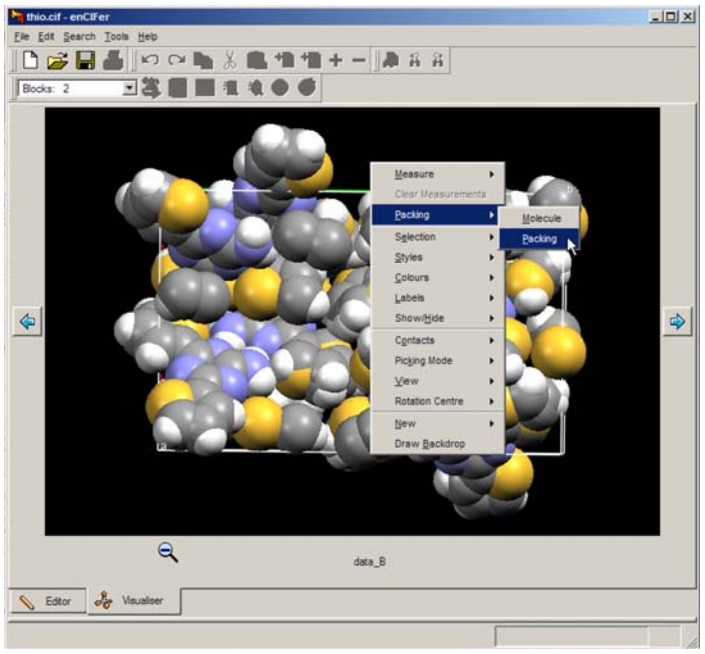


Fig. 5.3.3.5. Visualization of a molecular and crystal three-dimensional structure with *enCIFer*.

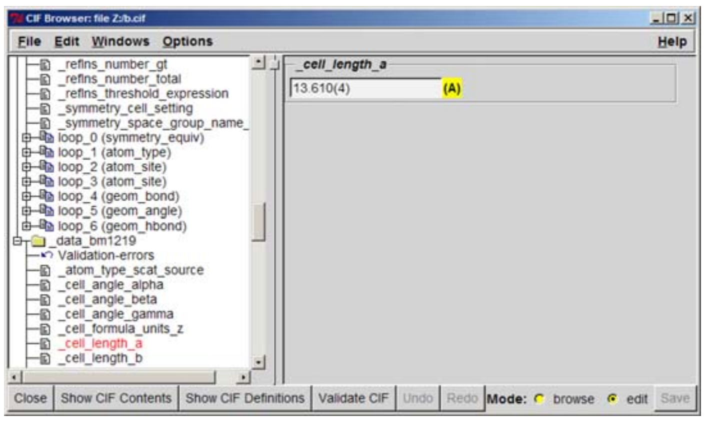
a space-filling molecular representation, and the drop-down menu indicates some of the options available to modify the appearance of the graphics. The molecular-graphics library used by the program is part of the larger database interface software package developed at the Cambridge Crystallographic Data Centre. In the present version, the visualizer is run only upon initial parsing of the input CIF, and therefore does not provide an ability to track visually the molecular changes associated with direct modification of the contents of the file.

5.3.3.2. *CIFEDIT*

The *CIFEDIT* program (Toby, 2003) is written in Tcl/Tk (Ousterhout, 1994) and provides an application for viewing and editing CIFs. The code is written in such a way that it can be embedded into larger programs to provide a CIF-editing interface within larger application suites.

The current version of the program is able to validate CIFs against both DDL1 and DDL2 dictionaries, although the DDL2 validation is currently less complete than for DDL1. For example, numeric values are checked against permitted enumeration ranges only for DDL1. Dictionaries are accessed through index files, each of which contains Tcl data structures that point to the location of the definitions in the dictionary file itself and store information such as units and enumeration ranges that can be used for data validation. A utility provided with the program allows a user to generate new index files when new versions of the dictionaries become available. It is intended that dictionary indexing will be incorporated within the main application in the next program release, so that interactive dictionary selection will be possible.

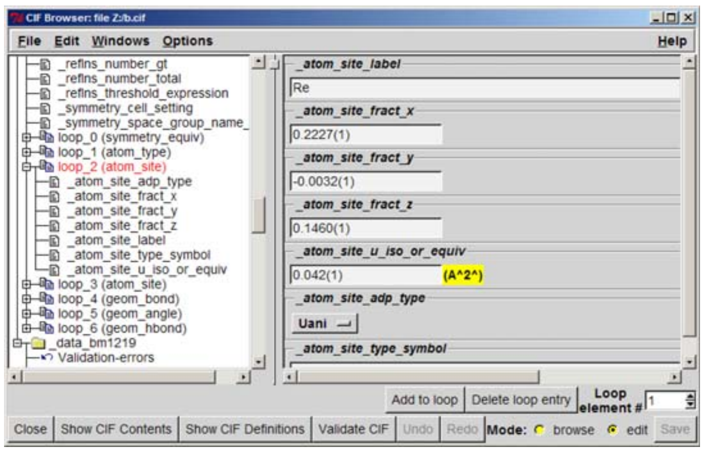
When a CIF is opened, the contents are parsed and validated against one or more user-selected dictionaries. Errors are displayed in a pop-up window and may be written to a file or viewed within the application. The main program window displays the contents of the CIF in two primary panes (Fig. 5.3.3.6). In the left-hand pane, a tree structure shows the data blocks in the file and the data names present in each block. The data blocks may be expanded or collapsed by the user, to present an overview or a detailed view of the data structure of the file. Underneath the icon representing the data block, non-looped data items are listed alphabetically. The figure demonstrates how a single value may be selected in the left-hand pane (`_cell_length_a`) and displayed in the main window. Physical units for the selected quantity are extracted from the corresponding dictionary definition and presented alongside the numeric value. The dictionary definition may also be displayed in a separate pop-up window using the 'Show CIF Definitions' button.



_cell_length_a: 13.610(4) (A)

Mode: browse [✓ edit]

Fig. 5.3.3.6. The use of *CIFEDIT* to display and alter the contents of a CIF; here a non-looped data item is shown.



_atom_site_label: Re
_atom_site_fract_x: 0.2227(1)
_atom_site_fract_y: -0.0032(1)
_atom_site_fract_z: 0.1460(1)
_atom_site_u_iso_or_equiv: 0.042(1) (A^2^)
_atom_site_adp_type: Uani
_atom_site_type_symbol

Loop element #: 1

Mode: browse [✓ edit]

Fig. 5.3.3.7. Row-based loop editing with *CIFEDIT*; here loop_2 (comprising the ATOM_SITE category) has been selected by the user; the editing cursor begins at row 1 of the loop.

The program may be run in two modes: a 'browse' mode, where the selected value is displayed in the main pane, but may not be altered; and an 'edit' mode (as in the example) where the value appears in an editable text widget.

Data loops in the CIF are displayed after the alphabetical list of non-looped items. The loops are numbered sequentially from zero and an indication of the loop category is given in parentheses in the tree-view window. The loop 'branches' of the tree may be expanded or collapsed as the user wishes.

Loops may be viewed and edited in two ways: by row or by column. If the user selects the loop title node in the hierarchical view pane, the loop is presented by row, starting in sequence at row 1 (Fig. 5.3.3.7). Other rows may be selected by using the address box in the lower-right-hand part of the window. Alternatively, if the user selects an individual data name within the loop representation in the hierarchical view, all instances of that data item within the loop are displayed in the main pane. (In practice the number of values shown is constrained to a maximum number that the user may choose, so that the application does not run out of memory if there are very large loops.)

For items with a restricted set of permitted values in the dictionary, the editing function allows the user to select only one of the permitted options *via* a drop-down menu.

While the application is intended to be used in this structured and itemized mode, there is an option to open the entire CIF in a text-editing window if there are errors that cannot be handled in the normal mode. This is not recommended, but is occasionally convenient. While this free-text editing mode is in operation, the ability to modify the file through the structured editing pane is suspended to avoid conflicting changes.

After any change has been made, the user may revalidate the file. This is strongly recommended after making changes in the free-text editing mode.

5.3.3.3. *HICCuP*

The program *HICCuP* (Edgington, 1997) was an early graphical utility developed at the Cambridge Crystallographic Data Centre for interactive editing and validation of a CIF. It is no longer supported, having been replaced by *enCIFer* (Section 5.3.3.1). Nevertheless, it contained some interesting features and is of potential interest to developers using multiple-platform scripting languages. It was implemented in the Python language (van Rossum, 1991) and required that Tcl/Tk (Ousterhout, 1994) be also available on the host computer. The name of the program is an acronym for 'High-Integrity CIF Checking using Python'.

HICCuP was designed to allow users of the Cambridge Structural Database (Allen, 2002) to check structures intended for deposition in the database and therefore included a range of additional content checks specific to this purpose. These could, however, be disabled by the user.

5.3.3.3.1. *Interactive use of the program*

5.3.3.3.1.1. *The control window*

Because *HICCuP* was designed as an interactive tool, upon invocation it presented to the user a *control window* from which CIFs could be selected for analysis and in which summary results of the program's operations were logged. Fig. 5.3.3.8 shows an example of the control window after a single CIF has been loaded.

In the large frame below the file-entry field are listed the data blocks found by the program. The names are highlighted in various colours according to the highest level of severity of errors found within the corresponding data block.

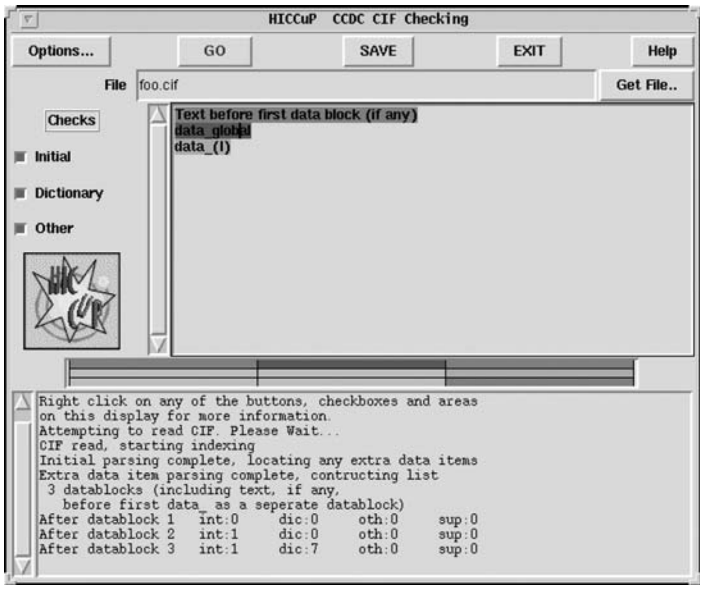
Because the utility was designed for processing large amounts of CIF data for structural databases, it was considered useful to supply a compact visual indicator of the progress of the program through a large file. This takes the form of a grid of rectangular cells, one column for each data block present. Each column contains three cells, which monitor the performance of checks on the file syntax, conformance against a CIF dictionary, and other checks specific to the requirements of the Cambridge Crystallographic Data Centre. As each data block was checked, the corresponding cells were coloured according to the types of error found. Different colours were used to indicate: no errors; structure errors in the initial syntax tests; dictionary errors; or a deviation from certain conventions used by journals and databases in naming datablocks.

The large frame at the bottom of the control window provides a text summary of the same information, listing the number of errors found.

Check boxes and an 'Options…' button allowed some configurability of checks by the user.

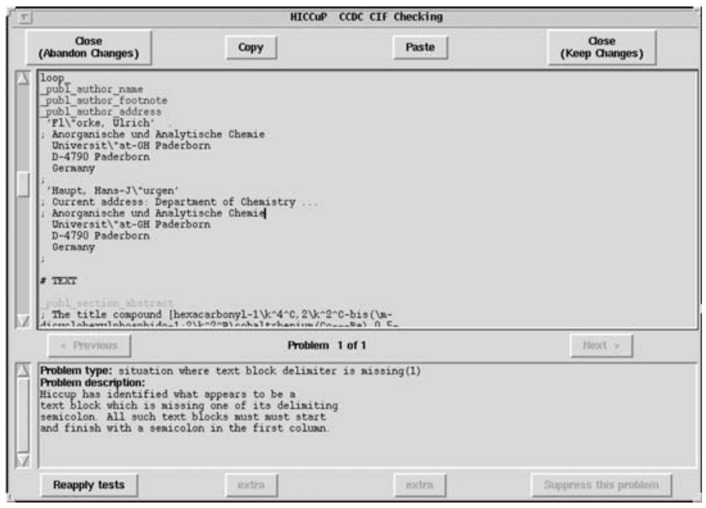
5.3.3.3.1.2. *The report frame and edit window*

The user could get more details of the reported errors by clicking on the name of the data block of interest in the control window. The text of the CIF would appear in a new window positioned at the point where the program has detected the first error and a terse statement of the type of error, with a longer explanation of its nature and possible cause, would be given.



```
Right click on any of the buttons, checkboxes and areas
on this display for more information.
Attempting to read CIF. Please Wait...
CIF read, starting indexing
Initial parsing complete, locating any extra data items
Extra data item parsing complete, contructing list
 3 datablocks (including text, if any,
   before first data_ as a seperate datablock)
After datablock 1   int:0     dic:0     oth:0     sup:0
After datablock 2   int:1     dic:0     oth:0     sup:0
After datablock 3   int:1     dic:7     oth:0     sup:0
```

Fig. 5.3.3.8. Control window of the *HICCuP* application.



```
loop_
_publ_author_name
_publ_author_footnote
_publ_author_address
 'Fl\"orke, Ulrich'
; Anorganische und Analytische Chemie
  Universit\"at-GH Paderborn
  D-4790 Paderborn
  Germany
;
 'Haupt, Hans-J\"urgen'
; Current address: Department of Chemistry ...
; Anorganische und Analytische Chemie
  Universit\"at-GH Paderborn
  D-4790 Paderborn
  Germany
;

# TEXT

_publ_section_abstract
; The title compound [hexacarbonyl-1\k^4^C,2\k^2^C-bis(\m-
```

Problem type: situation where text block delimiter is missing(1)
Problem description:
Hiccup has identified what appears to be a
text block which is missing one of its delimiting
semicolon. All such text blocks must must start
and finish with a semicolon in the first column.

Fig. 5.3.3.9. *HICCuP* edit window and error description.

In the example of Fig. 5.3.3.9, the program has detected that there is a missing text delimiter (a semicolon character), and positions the text in the upper frame at the likely location of the error. The program has attempted to localize the region where the error may have occurred. Because a text field might contain arbitrary contents, including extracts of CIF content, it is impossible to be sure on purely syntactic grounds of the nature of the error. Nonetheless, some heuristic rules serve to identify the author's likely intent in the majority of cases. So, in this example, the user may scan the file contents in the vicinity of the line highlighted by the program and find the error within a few lines (in this example an incorrectly terminated **`_publ_author_footnote`** entry beginning 'Current address:').

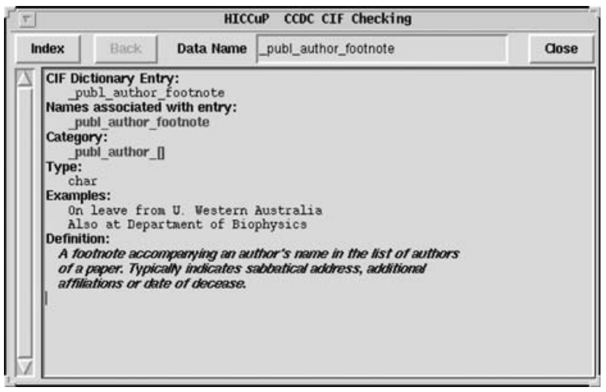
For this example, the more literal *vcif* error analysis provides only the message

```
ERROR: Text field at end of file does not terminate
```

The upper frame in this window is an editable window, so that the user could modify the text and revalidate the current data block. Only when a satisfactorily 'clean' data block was obtained were the changes saved, and the modified data block written back into the original file.

5.3.3.3.1.3. *Dictionary browsing*

An additional useful feature of the program was its interactive link to a CIF dictionary file (Fig. 5.3.3.10). The browser window contains the definition section of the dictionary referring to



CIF Dictionary Entry:
    _publ_author_footnote
Names associated with entry:
    _publ_author_footnote
Category:
    _publ_author_[]
Type:
    char
Examples:
    On leave from U. Western Australia
    Also at Department of Biophysics
Definition:
  A footnote accompanying an author's name in the list of authors of a paper. Typically indicates sabbatical address, additional affiliations or date of decease.

Fig. 5.3.3.10. *HICCuP* dictionary browser window.

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.

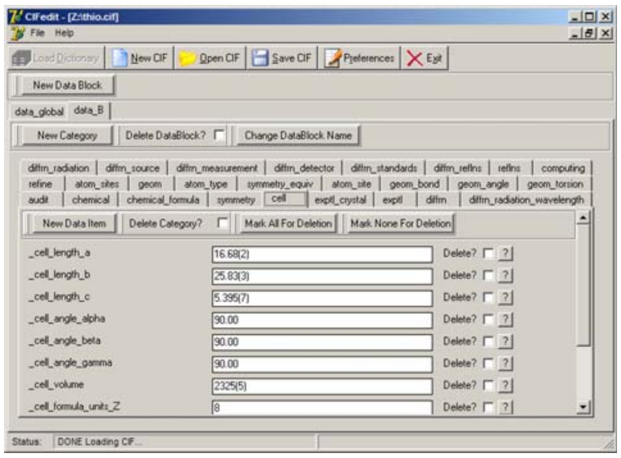
The program could also be run in a batch mode when the objective was to work through a large volume of CIF data and identify the data blocks that require attention. This mode of operation is particularly useful in databases or publishing houses. In this mode, input is from a named file or from the standard 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



```
_cell_length_a          16.68(2)
_cell_length_b          25.83(3)
_cell_length_c          5.395(7)
_cell_angle_alpha       90.00
_cell_angle_beta        90.00
_cell_angle_gamma       90.00
_cell_volume            2325(5)
_cell_formula_units_Z   8
Status: DONE Loading CIF...
```

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

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.

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.

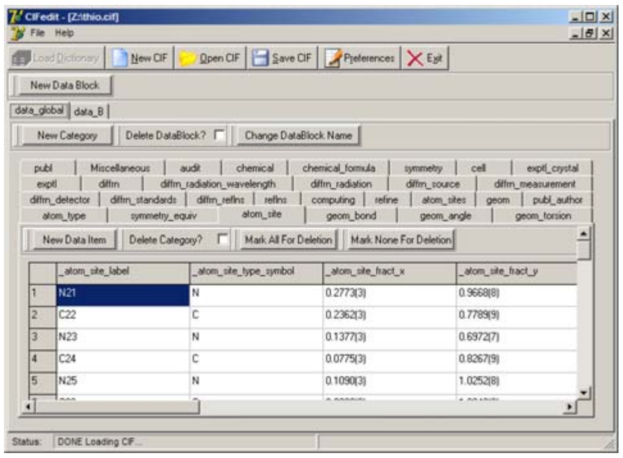
Where the category contains non-looped data items, values may be edited within individual text widgets; data items may be removed by selecting the adjacent check box; or new data items may be added by selecting the 'New Data Item' button to create a dialogue box offering a choice of the remaining data 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.

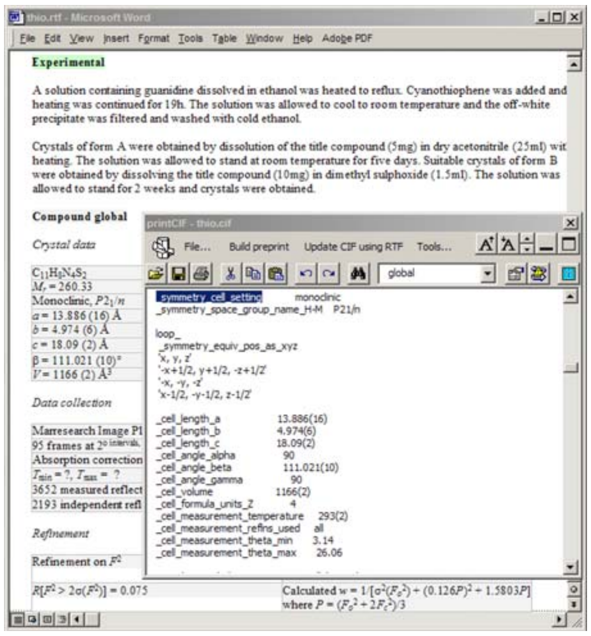
5.3.3.4.2. *printCIF for Word*

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)

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


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.

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. 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.