

5.3. SYNTACTIC UTILITIES FOR CIF

```

_cell_formula_units_Z      2
_cell_length_a             8.79(2)
_refine_ls_extinction_coef .347e4(5)
_chemical_name_common      'copper sulphate'

```

Fig. 5.3.5.4. Sample CIF data input to *ciftex*.

```

\cellz{2}
\nobreak\cella{8.79 (2)}
\extcoeffLarson{0.347 (5) $\times$ $10^{4}$}
\chemcom{copper sulfate}

```

Fig. 5.3.5.5. Output from *ciftex* run on the data of Fig. 5.3.5.4.

5.3.5.3.1.1. Non-looped data

For portions of the CIF that are not contained in looped lists, the transformations are trivial. A (*data name*, *data value*) pair is transformed to a T_EX macro and its argument. The macro name is determined from an external ‘map’ file which the program reads at run time; this file associates CIF data names and the corresponding T_EX macros through a simple lookup table.

A CIF data value is in most cases passed as the argument to the corresponding T_EX macro with few modifications. If the data value is a character string beginning with an integer, full point, hyphen or plus character, it is assumed to be of type ‘numb’. A space is introduced ahead of an embedded open parenthesis (to separate a standard uncertainty from its parent value). A leading zero is printed before any bare decimal point. An embedded ϵ is taken to indicate exponential notation and the format of the number is accordingly modified.

If the input data value is of type ‘char’ (*i.e.* is a single token beginning with characters other than those recognized as the leading characters for numerical data; or contains multiple tokens delimited by quote marks or semicolons), the program will search the map file for key values exactly matching each token, and if found will substitute the token by its replacement word or text. If no replacement is specified in the map file, the token is passed unchanged to the standard output channel. This facility was found to be useful in making global substitutions of individual words during file processing, but must be used with care since the substitutions are unconditional, without any reference to context.

Some small examples of typical non-looped data items are shown in Fig. 5.3.5.4 and the corresponding *ciftex* translation based on a map file used for typesetting *Acta Crystallographica Section C* is shown in Fig. 5.3.5.5.

Note the transformations of the numerical arguments and the translation of ‘sulphate’ to ‘sulfate’.

5.3.5.3.1.2. Looped data

If the input token is a `loop_` keyword, the program enters a different mode of operation. Looped data may be represented in print either as repetitive lists or in tabular format. There is no indication in a CIF dictionary of the appropriate representation (nor should there be, for what is essentially a matter of presentation) and the choice is made based on a flag associated with each data name in the map file. For non-tabular lists, the structure

```

loop_
  _dataname_1
  _dataname_2
  value_1     value_2
  value_3     value_4

```

```

\settabs 5 \columns
\+ \relax & $x$ & $y$ & $z$ & $U_{\rm eq}$ & \cr
\+Re &0.222 (1) &0.003 (1) &0.146 (1) &0.042 (1) & \cr
\+Co &0.234 (1) &0.139 (1) &0.299 (1) &0.046 (1) & \cr
\+P1 &0.358 (1) &0.222 (1) &0.197 (1) &0.044 (1) & \cr
\+P2 &0.106 (2) &0.051 (1) &0.289 (1) &0.046 (1) & \cr
\+C1 &0.308 (6) &0.029 (6) &0.034 (4) &0.057 (4) & \cr
\+O1 &0.356 (5) &0.044 (5) &0.030 (3) &0.079 (3) & \cr
\+C2 &0.066 (6) &0.039 (6) &0.111 (4) &0.056 (4) & \cr

```

Fig. 5.3.5.6. T_EX markup for typesetting a table of atomic coordinates.

is translated to a sequence of T_EX codes of the form

```

\macro_one(value_1)
\macro_two(value_2)
\macro_one(value_3)
\macro_two(value_4)

```

In the case of tabulated data, the `loop_` header is translated into a set of table headings and typographic codes are introduced to lay out in columnar format the values in the body of the list. The number of different data names in the loop header is counted and the data values are identified by their position in the loop modulo the total number of data names in the header (in effect, by their ‘phase’ in the loop). In the simplest case, a T_EX command is emitted that builds a table with *n* columns, where *n* is the number of different data names. Then the data values are counted as they are processed. After every *n*th data value, a T_EX code is emitted indicating ‘end of table row’ and a further code is emitted before the next value (if there is one) that means ‘beginning of new table row’. In all other cases, a code is emitted signifying ‘move to next column’.

Fig. 5.3.5.6 is a simplified extract from a table of atomic coordinates derived from the `_atom_site_` loop in a CIF.

5.3.5.3.1.3. The ancillary map file

The translation between a CIF data name and its replacement text in the T_EX output file is defined in the external map file. The format of the translation is very simple, as illustrated in Fig. 5.3.5.7.

Each line starts with a CIF data name, which is terminated by a space character. The next character is either ‘T’ or ‘N’ to indicate whether the output should be tabulated or not. The next character is an arbitrary character from the ASCII character set, and is chosen to collect together data that will appear in the same logical section of the output file. This locator character may be associated, in another ancillary file described below, with additional text for output. The remainder of the line is the replacement text.

In the example supplied, the cell-length parameters map to the T_EX macros `\cella`, `\cellb` and `\cellc` (each preceded by a standard T_EX macro forbidding a page break immediately before the contents are printed). The details of the publication authors are described by a set of T_EX macros that will occur in two different locations in the output file (the authors’ names and addresses may be looped together in the location labelled by the character *a*; any explanatory footnotes and email addresses will be printed elsewhere in the paper, at the location labelled *x*). The anisotropic displacement parameters U^{ij} will be printed in a table and the replacement text consists of the T_EX codes that will be printed at the head of each column in the table.

The initial text on the line need not be a CIF data name; it may be any other single word. In this case, every occurrence of that word in the input CIF will be replaced by the replacement text.

```

_cell_length_a Ng\nobreak\cella
_cell_length_b Ng\nobreak\cellb
_cell_length_c Ng\nobreak\cellc

_publ_author_name Na\author
_publ_author_address Na\address
_publ_author_footnote NX\aufootnote
_publ_contact_author_email NX\email

_atom_site_aniso_label TU\relax
_atom_site_aniso_U_11 TU{\hfill $U^{11}$ \hfill}
_atom_site_aniso_U_12 TU{\hfill $U^{12}$ \hfill}
_atom_site_aniso_U_13 TU{\hfill $U^{13}$ \hfill}
_atom_site_aniso_U_22 TU{\hfill $U^{22}$ \hfill}
_atom_site_aniso_U_23 TU{\hfill $U^{23}$ \hfill}
_atom_site_aniso_U_33 TU{\hfill $U^{33}$ \hfill}

```

Fig. 5.3.5.7. Example map file for use with *ciftex*.

If the initial *character* of the line is a hash mark #, the line is treated as a comment and discarded.

5.3.5.3.1.4. The ancillary format file

Because a printed paper may be more verbose than its parent CIF data file, it is necessary to add text to the output from *ciftex* to represent section headings, line spaces or other formatting instructions. The program reads an ancillary file, known as the format file, for such additional text.

Each line in the format file begins with a hash mark #, a single ASCII character and a colon. The second character is chosen to match the corresponding locator character associated with data names in the map file. The rest of the line is text to be output. When the locator character associated with the data name currently being processed differs from the previous one, the output text from all lines in the format file with the new locator character are output.

The special strings #[: and #] : indicate text to be emitted at the beginning and end of the output stream, respectively.

Fig. 5.3.5.8 is an example of a simplified format file. The first line is printed at the start of the output \TeX file; the second line at the end. The next line will be printed on the first occurrence of a data name flagged with the locator code *a* in the map file. In this example, that will be the name or address of an author of the paper; some typographic directives are emitted immediately before the authors' names and addresses, including the introduction of a blank line ('vertical skip', or 'vskip') of height 10 typographic points.

The lines beginning #g: are emitted immediately before the first data name in the group that is associated with locator code *g*. In this example, the effect is to output a heading and subheading before printing the cell-length parameters and to switch to double-column format. The line containing *only* the characters #g: provides for the introduction of a blank line into the \TeX file, with the sole purpose of making the file more readable by human editors.

The lines beginning #U: are emitted at the beginning of the table of anisotropic *U* values.

The mechanism looks complicated at first sight, but addresses the need to generate headings at standard locations in a printed paper when the exact content of the paper is not known in advance.

The different format for directives in the map and format files means that the same file can be used for both purposes, if required. In practice it is often easier to maintain different files: the same mapping between CIF data names and \TeX macros might be common to different journals, while each journal uses its own format file.

```

#[:\newif\ifproof \prooftrue
#]:\iftwocol\vfill\enddoublecolumns\fi
#a:\pretolerance1000\parskip0pt\tolerance5000
#a:\vskip10pt
#g:
#g:%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
#g:\iftwocol\enddoublecolumns\twocolfalse\fi
#g:\tenbf Experimental
#g:\noindent\ninebf Compound \datablock\vskip2pt
#g:\noindent\nineit Crystal data\par
#g:\vskip2pt\begindoublecolumns\twocoltrue\defaultfont
#U:%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
#U:\iftwocol\enddoublecolumns\twocolfalse\fi
#U:\rm Table \tableno. \it Anisotropic displacement
#U:parameters \rm (AA$^2$) for \datablock
#U:\vskip 6pt

```

Fig. 5.3.5.8. Example format file for *ciftex*.

5.3.5.3.2. Invocation of the program

The program reads a CIF on the standard input channel and outputs \TeX code on standard output. There is no provision to specify file names. It is therefore invoked within a Unix-style operating system by a command such as

```
ciftex < infile > outfile
```

where *infile* and *outfile* are the input and output files respectively; or it may be called as part of a pipeline of procedures:

```
program 1 < infile | ciftex | program 2...
```

A number of command-line options may be supplied to modify the operation of the program. Other than the specification of the map and format files, they are largely relevant to differing house styles for IUCr journals.

The options *-map mapfile* and *-format formatfile* specify the names of the ancillary map and format files. If not specified, they are sought in default locations on the user's file system (different values may be defined when the program is compiled) or as specified in the environment variables $\$CIFTEX_MAP$ and $\$CIFTEX_FORMAT$, respectively.

The options *-H* and *-N* specify, respectively, whether or not hydrogen atoms in coordinate tables should be printed. The hydrogen-atom lines in the table are in fact always emitted on standard output, but in the case of the *-N* option are prefixed by a % (\TeX comment) character and so ignored by \TeX .

Options *-c* and *-F* specify the printing of centred decimal points or commas for decimal points, respectively. Finally, the option *-d* modifies certain assumptions that *ciftex* makes when typesetting CIF dictionaries. The details are of interest only to a specialist.

5.3.5.3.3. Some general comments

Although *ciftex* is available for public use and redistribution within the academic community, it is clearly of most interest to users who need to generate typeset representations of the contents of CIFs. Nevertheless, some elements of its design are relevant to other applications that perform on-the-fly file transformations on a strictly syntactic basis.

First, the functionality is very simple, essentially tokenizing the input data stream and exchanging tokens for replacement text as directed. An immediate consequence of this is the need for additional utilities to manipulate the input file if, for example, the data need to be presented in a particular order. In the journals production process, *QUASAR* is used to reorder an input file before passing it to *ciftex*.