

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.