

2. CONCEPTS AND SPECIFICATIONS

Example 2.3.4.1. A multiple-block CBF with several images.

```
###CBF: VERSION 1.0
# CBF file written by cbflib v0.6

# A comment cannot appear before the file identifier,
# but can appear anywhere else, except within the
# binary sections.

# Here the first data block starts
data_xxx

### ... various CIF tags and values here
###   but none that define array data items

# The "data_" identifier finishes the first data
# block and starts the second
data_yyy

### ... various CIF tags and values here including
###   ones that define array data items

loop_
  _array_data.array_id
  _array_data.binary_id
  _array_data.data

image_1 1
;
--CIF-BINARY-FORMAT-SECTION--
Content-Type: application/octet-stream;
  conversions="x-CBF_PACKED"
Content-Transfer-Encoding: BINARY
X-Binary-Size: 3745758
X-Binary-ID: 1
X-Binary-Element-Type: "signed 32-bit integer"
Content-MD5: 1zsJjWPfol2GY12V+QsXrw==

START_OF_BIN
<D5>^P<B8>P^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@ ...
[This is where the raw binary data would be - we can't print them here]

--CIF-BINARY-FORMAT-SECTION----
;
```

2.3.4. A complex example

In Example 2.3.4.1 only the ARRAY_DATA category CIF identifiers are shown. The other CIF data items are not shown. This shows a possible structuring of a more complicated example. There are two header sections; the first contains two data blocks and defines three binary sections. CIF comment lines, starting with a hash mark '#', are used to clarify the structure.

2.3.5. imgCIF encodings

For an imgCIF, there are several alternative encodings for binary image data as ASCII text. Each binary image may use a different encoding in the same imgCIF data set or even in the same data block. The choice of encoding is specified in the 'Content-Transfer-Encoding' MIME header.

If the Transfer Encoding is X-BASE8, X-BASE10 or X-BASE16, the data are presented as octal, decimal or hexadecimal data, respectively, organized into lines or words. Each word is created by composing octets of data in fixed groups of 2, 3, 4, 6 or 8 octets, either in the order ...4321 ('big-endian') or 1234... ('little-endian'). If there are fewer than the specified number of octets to fill the last word, then the missing octets are presented as '=' for each missing octet. Exactly two equal signs are used for each missing octet even for octal and decimal encoding. The format of lines is

```
rnd xxxxxxx xxxxxxx xxxxxxx
```

where *r* is H, O or D for hexadecimal, octal or decimal, *n* is the number of octets per word, and *d* is '<' or '>' for the '...4321' and

Example 2.3.4.1. (cont.)

```
# Following the "end of binary" identifier the file
# is pseudo-ASCII again, so comments are valid
# up to the next "start of binary" identifier. Note
# that we have increased the binary ID by one.

image_1 2
;
--CIF-BINARY-FORMAT-SECTION--
Content-Type: application/octet-stream;
  conversions="x-CBF_PACKED"
Content-Transfer-Encoding: BINARY
X-Binary-Size: 3745758
X-Binary-ID: 2
X-Binary-Element-Type: "signed 32-bit integer"
Content-MD5: xR5kxiOetd9T/Nr5vMfAmA==

START_OF_BIN
<D5>^P<B8>P^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@^@ ...
[This is where the raw binary data would be - we can't print them here]

--CIF-BINARY-FORMAT-SECTION----
;

# Third binary section; note that we have a new
# array ID.

image_2 3
;
--CIF-BINARY-FORMAT-SECTION--
Content-Type: application/octet-stream;
  conversions="x-CBF_PACKED"
Content-Transfer-Encoding: BINARY
X-Binary-ID: 3
Content-MD5: yS5kxiOetd9T/NrqTLfAmA==

START_OF_BIN
*****'g*****'***** ...
[This is where the raw binary data would be - we can't print them here]

--CIF-BINARY-FORMAT-SECTION----
;

# Second header section
data_zzz

### ... various CIF tags and values here
###   including ones that define array
###   data items

# Since we only have one block left, we won't
# use a loop

  _array_data.array_id  image
  _array_data.binary_id 1
  _array_data.data

# Note that we can put a comment here
;
--CIF-BINARY-FORMAT-SECTION--
Content-Type: application/octet-stream;
  conversions="x-CBF_PACKED"
Content-Transfer-Encoding: BINARY
X-Binary-ID: 1
Content-MD5: f00xiOetd9T/serNufAmA==

START_OF_BIN
*****'g*****'***** ...
[This is where the raw binary data would be - we can't print them here]

--CIF-BINARY-FORMAT-SECTION----
;

### END_OF_CBF
```

'1234...' octet orderings, respectively. The '=' padding for the last word should be on the appropriate side to correspond to the missing octets, e.g.

```
H4< FFFFFFFF FFFFFFFF 07FFFFFFF ====0000
```