4.6. imgCIF draft version 1.4

BY A. P. HAMMERSLEY, H. J. BERNSTEIN AND J. D. WESTBROOK

WARNING: this is a draft for discusssion subject to change without notice. Versions will be posted as cif_img_1.4_*DDMMMYY*_draft.html. Send comments to imgcif-l@iucr.org citing the version.

Category groups

array_data_group	Categories that describe array data.
axis_group	Categories that describe axes.
diffrn_group	Categories that describe details of the diffraction experiment.

ARRAY_DATA

Data items in the ARRAY_DATA category are the containers for the array data items described in the category ARRAY_STRUCTURE.

Category group(s): **inclusive_group**
array_data_group
Category key(s): **_array_data.array_id**
_array_data.binary_id

Example 1 -
This example shows two binary data blocks. The first one was compressed by the CBF_CANONICAL compression algorithm and is presented as hexadecimal data. The first character 'H' on the data lines means hexadecimal. It could have been 'O' for octal or 'D' for decimal. The second character on the line shows the number of bytes in each word (in this case '4'), which then requires eight hexadecimal digits per word. The third character gives the order of octets within a word, in this case '¡' for the ordering 4321 (i.e. 'big-endian'). Alternatively, the character '¿' could have been used for the ordering 1234 (i.e. 'little-endian'). The block has a 'message digest' to check the integrity of the data.
The second block is similar, but uses CBF_PACKED compression and BASE64 encoding. Note that the size and the digest are different.

```
    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_CANONICAL"
    Content-Transfer-Encoding: X-BASE16
    X-Binary-Size: 3927126
    X-Binary-ID: 1
    Content-MD5: u2sTJEovAHkmkDjPi+gWsg==

    # Hexadecimal encoding, byte 0, byte order ...21
    #
    H4< 0050B810 00000000 00000000 00000000 000F423F 00000000
00000000 ...
    ....
    --CIF-BINARY-FORMAT-SECTION----
    ;
    image_2 2
    ;
    --CIF-BINARY-FORMAT-SECTION--
    Content-Type: application/octet-stream;
         conversions="x-CBF-PACKED"
    Content-Transfer-Encoding: BASE64
    X-Binary-Size: 3745758
    X-Binary-ID: 2
    Content-MD5: 1zsJjWPfol2GYl2V+QSXrw==

    ELhQAAAAAAAA...
    ...
    --CIF-BINARY-FORMAT-SECTION----
    ;
```

*_array_data.array_id (*code*)

This item is a pointer to **_array_structure.id** in the ARRAY_STRUCTURE category.

[array_data]

(*)_array_data.binary_id (*int*)

This item is an integer identifier which, along with **_array_data.array_id**, should uniquely identify the particular block of array data. If **_array_data.binary_id** is not explicitly given, it defaults to 1. The value of **_array_data.binary_id** distinguishes among multiple sets of data with the same array structure. If the MIME header of the data array specifies a value for X-Binary-ID, the value of **_array_data.binary_id** should be equal to the value given for X-Binary-ID.

The following item(s) have an equivalent role in their respective categories:
_diffrn_data_frame.binary_id,
_array_intensities.binary_id.

The permitted range is $[1, \infty)$. Where no value is given, the assumed value is '1'.

[array_data]

*_array_data.data (*binary*)

The value of **_array_data.data** contains the array data encapsulated in a STAR string. The representation used is a variant on the Multipurpose Internet Mail Extensions (MIME) specified in RFC 2045-2049 by N. Freed *et al.* The boundary delimiter used in writing an imgCIF or CBF is **--CIF-BINARY-FORMAT-SECTION--** (including the required initial **--**).

The Content-Type may be any of the discrete types permitted in RFC 2045; 'application/octet-stream' is recommended. If an octet stream was compressed, the compression should be specified by the parameter **conversions="x-CBF_PACKED"** or the parameter **conversions="x-CBF_CANONICAL"**.

The Content-Transfer-Encoding may be 'BASE64', 'Quoted-Printable', 'X-BASE8', 'X-BASE10', 'X-BASE16' or 'X-BASE32K', for an imgCIF or 'BINARY' for a CBF. The octal, decimal and hexadecimal transfer encodings are provided for convenience in debugging and are not recommended for archiving and data interchange.

In a CIF, one of the parameters 'charset=us-ascii', 'charset=utf-8' or 'charset=utf-16' may be used on the Content-Transfer-Encoding to specify the character set used for the external presentation of the encoded data. If no charset parameter is given, the character set of the enclosing CIF is assumed. In any case, if a BOM flag is detected (FE FF for big-endian UTF-16, FF FE for little-endian UTF-16 or EF BB BF for UTF-8) is detected, the indicated charset will be assumed until the end of the encoded data or the detection of a different BOM. The charset of the Content-Transfer-Encoding is not the character set of the encoded data, only the character set of the presentation of the encoded data and should be respecified for each distinct STAR string.

In an imgCIF file, the encoded binary data begins after the empty line terminating the header. In an imgCIF file, the encoded binary data ends with the terminating boundary delimiter '**\n--CIF-BINARY-FORMAT-SECTION----**' in the currently effective charset or with the '; ' that terminates the STAR string.

In a CBF, the raw binary data begins after an empty line terminating the header and after the sequence:

Octet	Hexadecimal	Decimal	Purpose
0	0C	12	Ctrl-L: page break
1	1A	26	Ctrl-Z: stop listings in MS-DOS
2	04	04	Ctrl-D: stop listings in UNIX
3	D5	213	binary section begins

None of these octets are included in the calculation of the message size or in the calculation of the message digest.

The X-Binary-Size header specifies the size of the equivalent binary data in octets. If compression was used, this size is the size after compression, including any book-keeping fields. An adjustment is made for the deprecated binary formats in which eight bytes of binary header are used for the compression type. In this case, the eight bytes used for the compression type are subtracted from the size, so that the same size will be reported if the compression type is supplied in the MIME header. Use of the MIME header is the recommended way to supply the compression type. In general, no portion of the binary header is included in the calculation of the size.

The X-Binary-Element-Type header specifies the type of binary data in the octets, using the same descriptive phrases as in `_array_structure.encoding_type`. The default value is 'unsigned 32-bit integer'.

An MD5 message digest may, optionally, be used. The *RSA Data Security, Inc. MD5 Message-Digest Algorithm* should be used. No portion of the header is included in the calculation of the message digest.

If the Transfer Encoding is 'X-BASE8', 'X-BASE10' or 'X-BASE16', the data are presented as octal, decimal or hexadecimal data 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 xxxxxx xxxxxx xxxxxx
```

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 '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 07FFFFFF ====0000
```

or

```
H3> FF0700 00====
```

For these hexadecimal, octal and decimal formats only, comments beginning with '`#`' are permitted to improve readability.

BASE64 encoding follows MIME conventions. Octets are in groups of three: c1, c2, c3. The resulting 24 bits are broken into four six-bit quantities, starting with the high-order six bits ($c1 \gg 2$) of the first octet, then the low-order two bits of the first octet followed by the high-order four bits of the second octet [$(c1 \& 3) \ll 4 \mid (c2 \gg 4)$], then the bottom four bits of the second octet followed by the high-order two bits of the last octet [$(c2 \& 15) \ll 2 \mid (c3 \gg 6)$], then the bottom six bits of the last octet ($c3 \& 63$). Each of these four quantities is translated into an ASCII character using the mapping

```
0         1         2         3
0123456789012345678901234567890123456789
|         |         |         |
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmn


4         5         6
012345678901234567890123
|         |         |
opqrstuvwxyz0123456789+/
```

with short groups of octets padded on the right with one '`=`' if c3 is missing, and with '`==`' if both c2 and c3 are missing.

X-BASE32K encoding is similar to BASE64 encoding, except that sets of 15 octets are encoded as sets of 8 16-bit unicode characters, by breaking the 120 bits into 8 15-bit quantities. 256 is added to each 15 bit quantity to bring it into a printable uncode range. When encoding, zero padding is used to fill out the last 15 bit quantity. If 8 or more bits of padding are used, a single equals sign (hexadecimal 003D) is appended. Embedded whitespace and newlines are introduced to produce lines of no more than 80 characters each. On decoding, all printable ascii characters and ascii whitespace characters are ignored except for any trailing equals signs. The number of trailing equals signs indicated the number of trailing octets to be trimmed from the end of the decoded data. (see Georgi Darakev, Vassil Litchev, Kostadin Z. Mitev, Herbert J. Bernstein, 'Efficient Support of Binary Data in the XML Implementation of the NeXus File Format',absract W0165, ACA Summer Meeting, Honolulu, HI, July 2006).

QUOTED-PRINTABLE encoding also follows MIME conventions, copying octets without translation if their ASCII values are 32...8, 42, 48...57, 59, 60, 62, 64...126 and the octet is not a ';' in column 1. All other characters are translated to =*nn*, where *nn* is the hexadecimal encoding of the octet. All lines are 'wrapped' with a terminating `=` (*i.e.* the MIME conventions for an implicit line terminator are never used).

[array_data]

ARRAY_ELEMENT_SIZE

Data items in the ARRAY_ELEMENT_SIZE category record the physical size of array elements along each array dimension.

Category group(s): `inclusive_group`
`array_data_group`
Category key(s): `_array_element_size.array_id`
`_array_element_size.index`

Example 1 – A regular 2D array with a uniform element dimension of 1220 nanometres.

```
    loop_
   _array_element_size.array_id
   _array_element_size.index
   _array_element_size.size
    image_1   1     1.22e-6
    image_1   2     1.22e-6
```

***_array_element_size.array_id** (*code*)

This item is a pointer to `_array_structure.id` in the ARRAY_STRUCTURE category.

[array_element_size]

***_array_element_size.index** (*code*)

This item is a pointer to `_array_structure_list.index` in the ARRAY_STRUCTURE_LIST category.

[array_element_size]

***_array_element_size.size** (*float*)

The size in metres of an image element in this dimension. This supposes that the elements are arranged on a regular grid.

The permitted range is $[0.0, \infty)$. **[array_element_size]**

ARRAY_INTENSITIES

Data items in the ARRAY_INTENSITIES category record the information required to recover the intensity data from the set of data values stored in the ARRAY_DATA category. The detector may have a complex relationship between the raw intensity values and the number of incident photons. In most cases, the number stored in the final array will have a simple linear relationship to the actual number of incident photons, given by `_array_intensities.gain`. If raw, uncorrected values are presented (*e.g.* for calibration experiments), the value of `_array_intensities.linearity` will be 'raw' and `_array_intensities.gain` will not be used.

Category group(s): `inclusive_group`
`array_data_group`
Category key(s): `_array_intensities.array_id`
`_array_intensities.binary_id`

Example 1

```
loop_
_array_intensities.array_id
_array_intensities.linearity
_array_intensities.gain
_array_intensities.overload
_array_intensities.undefined_value
_array_intensities.pixel_fast_bin_size
_array_intensities.pixel_slow_bin_size
_array_intensities.pixel_binning_method
image_1   linear  1.2    655535   0   2   2    hardware
```

***_array_intensities.array_id** (*code*)

This item is a pointer to **_array_structure.id** in the ARRAY_STRUCTURE category.

[array_intensities]

(*)_array_intensities.binary_id (*int*)

This item is a pointer to **_array_data.binary_id** in the ARRAY_DATA category.

[array_intensities]

***_array_intensities.gain** (*float*)

Detector 'gain'. The factor by which linearized intensity count values should be divided to produce true photon counts.

The permitted range is $[0.0, \infty)$.

Related item: **_array_intensities.gain_esd** (associated value).

[array_intensities]

***_array_intensities.gain_esd** (*float*)

The estimated standard deviation in detector 'gain'.

The permitted range is $[0.0, \infty)$.

Related item: **_array_intensities.gain** (associated esd).

[array_intensities]

***_array_intensities.linearity** (*code*)

The intensity linearity scaling method used to convert from the raw intensity to the stored element value: 'linear' is linear. 'offset' means that the value defined by **_array_intensities.offset** should be added to each element value. 'scaling' means that the value defined by **_array_intensities.scaling** should be multiplied with each element value. 'scaling_offset' is the combination of the two previous cases, with the scale factor applied before the offset value. 'sqrt_scaled' means that the square root of raw intensities multiplied by **_array_intensities.scaling** is calculated and stored, perhaps rounded to the nearest integer. Thus, linearization involves dividing the stored values by **_array_intensities.scaling** and squaring the result. 'logarithmic_scaled' means that the logarithm base 10 of raw intensities multiplied by **_array_intensities.scaling** is calculated and stored, perhaps rounded to the nearest integer. Thus, linearization involves dividing the stored values by **_array_intensities.scaling** and calculating 10 to the power of this number. 'raw' means that the data are a set of raw values straight from the detector.

The data value must be one of the following:

`linear`

.

`offset`

The value defined by **_array_intensities.offset** should be added to each element value.

`scaling`

The value defined by **_array_intensities.scaling** should be multiplied with each element value.

`scaling_offset`

The combination of the scaling and offset with the scale factor applied before the offset value.

`sqrt_scaled`

The square root of raw intensities multiplied by **_array_intensities.scaling** is calculated and stored, perhaps rounded to the nearest integer. Thus, linearization involves dividing the stored values by **_array_intensities.scaling** and squaring the result.

`logarithmic_scaled`

The logarithm base 10 of raw intensities multiplied by **_array_intensities.scaling** is calculated and stored, perhaps rounded to the nearest integer. Thus, linearization involves dividing the stored values by **_array_intensities.scaling** and calculating 10 to the power of this number.

`raw`

The array consists of raw values to which no corrections have been applied. While the handling of the data is similar to that given for 'linear' data with no offset, the meaning of the data differs in that the number of incident photons is not necessarily linearly related to the number of counts reported. This value is intended for use either in calibration experiments or to allow for handling more complex data-fitting algorithms than are allowed for by this data item.

[array_intensities]

_array_intensities.offset (*float*)

Offset value to add to array element values in the manner described by the item **_array_intensities.linearity**.

[array_intensities]

_array_intensities.overload (*float*)

The saturation intensity level for this data array.

[array_intensities]

(*)_array_intensities.pixel_binning_method (*code*)

The value of **_array_intensities.pixel_binning_method** specifies the method used to derive array elements from multiple pixels.

The data value must be one of the following:

`hardware`

The element intensities were derived from the raw data of one or more pixels by used of hardware in the detector, e.g. by use of shift registers in a CCD to combine pixels into super-pixels.

`software`

The element intensities were derived from the raw data of more than one pixel by use of software.

`combined`

The element intensities were derived from the raw data of more than one pixel by use of both hardware and software, as when hardware binning is used in one direction and software in the other.

`none`

When the value of **_array_intensities.pixel_binning_method** is 'none' the values of **_array_intensities.pixel_fast_bin_size** and **_array_intensities.pixel_slow_bin_size** both must be 1.

`unspecified`

The method used to derive element intensities is not specified.

Where no value is given, the assumed value is '`unspecified`'.

[array_intensities]

(*)_array_intensities.pixel_fast_bin_size (*float*)

The value of **_array_intensities.pixel_fast_bin_size** specifies the number of pixels that compose one element in the direction of the most rapidly varying array dimension. Typical values are 1, 2, 4 or 8. When there is 1 pixel per array element in both directions, the value given for **_array_intensities.pixel_binning_method** normally should be 'none'. It is specified as a float to allow for binning algorithms that create array elements that are not integer multiples of the detector pixel size.

The permitted range is $[0.0, \infty)$. Where no value is given, the assumed value is '`1.`'.

[array_intensities]

(*)_array_intensities.pixel_slow_bin_size (*float*)

The value of **_array_intensities.pixel_slow_bin_size** specifies the number of pixels that compose one element in the direction of the second most rapidly varying array dimension. Typical values are 1, 2, 4 or 8. When there is 1 pixel per array element in both directions, the value given for **_array_intensities.pixel_binning_method** normally should be 'none'. It is specified as a float to allow for binning algorithms that create array elements that are not integer multiples of the detector pixel size.

The permitted range is $[0.0, \infty)$. Where no value is given, the assumed value is '`1.`'.

[array_intensities]

_array_intensities.scaling (*float*)

Multiplicative scaling value to be applied to array data in the manner described by item **_array_intensities.linearity**.

[array_intensities]

_array_intensities.undefined_value (*float*)

A value to be substituted for undefined values in the data array.

[array_intensities]

ARRAY_STRUCTURE

Data items in the ARRAY_STRUCTURE category record the organization and encoding of array data that may be stored in the ARRAY_DATA category.

Category group(s): **inclusive_group**
array_data_group

Category key(s): **_array_structure.id**

Example 1 -

```
 loop_
_array_structure.id
_array_structure.encoding_type
_array_structure.compression_type
_array_structure.byte_order
 image_1       "unsigned 16-bit integer"  none  little_endian
```

***_array_structure.byte_order** (*code*)

The order of bytes for integer values which require more than 1 byte. (IBM-PC's and compatibles and DEC VAXs use low-byte-first ordered integers, whereas Hewlett Packard 700 series, Sun-4 and Silicon Graphics use high-byte-first ordered integers. DEC Alphas can produce/use either depending on a compiler switch.)

The data value must be one of the following:

`big_endian`
: The first byte in the byte stream of the bytes which make up an integer value is the most significant byte of an integer.

`little_endian`
: The last byte in the byte stream of the bytes which make up an integer value is the most significant byte of an integer.

[array_structure]

_array_structure.compression_type (*code*)

Type of data-compression method used to compress the array data.

The data value must be one of the following:

`none`
: Data are stored in normal format as defined by **_array_structure.encoding_type** and **_array_structure.byte_order**.

`packed`
: Using the 'packed' compression scheme, a CCP4-style packing (International Tables for Crystallography Volume G, Section 5.6.3.2)

`canonical`
: Using the 'canonical' compression scheme (International Tables for Crystallography Volume G, Section 5.6.3.1)

Where no value is given, the assumed value is '`none`'.

[array_structure]

***_array_structure.encoding_type** (*uline*)

Data encoding of a single element of array data. In several cases, the IEEE format is referenced. See IEEE Standard 754-1985 (IEEE, 1985).

Reference: IEEE (1985). IEEE Standard for Binary Floating-Point Arithmetic. ANSI/IEEE Std 754-1985. New York: Institute of Electrical and Electronics Engineers.

The data value must be one of the following:

`'unsigned 8-bit integer'`
`'signed 8-bit integer'`
`'unsigned 16-bit integer'`
`'signed 16-bit integer'`
`'unsigned 32-bit integer'`
`'signed 32-bit integer'`
`'signed 32-bit real IEEE'`
`'signed 64-bit real IEEE'`
`'signed 32-bit complex IEEE'`

[array_structure]

***_array_structure.id** (*code*)

The value of **_array_structure.id** must uniquely identify each item of array data.

The following item(s) have an equivalent role in their respective categories:
_array_data.array_id,
_array_structure_list.array_id,
_array_intensities.array_id,
_diffrn_data_frame.array_id.

[array_structure]

ARRAY_STRUCTURE_LIST

Data items in the ARRAY_STRUCTURE_LIST category record the size and organization of each array dimension. The relationship to physical axes may be given.

Category group(s): **inclusive_group**
array_data_group

Category key(s): **_array_structure_list.array_id**
_array_structure_list.index

Example 1 – An image array of 1300 x 1200 elements. The raster order of the image is left to right (increasing) in the first dimension and bottom to top (decreasing) in the second dimension.

```
    loop_
   _array_structure_list.array_id
   _array_structure_list.index
   _array_structure_list.dimension
   _array_structure_list.precedence
   _array_structure_list.direction
   _array_structure_list.axis_set_id
    image_1   1    1300    1     increasing  ELEMENT_X
    image_1   2    1200    2     decreasing  ELEMENY_Y
```

***_array_structure_list.array_id** (*code*)

This item is a pointer to **_array_structure.id** in the ARRAY_STRUCTURE category.

[array_structure_list]

***_array_structure_list.axis_set_id** (*code*)

This is a descriptor for the physical axis or set of axes corresponding to an array index. This data item is related to the axes of the detector itself given in DIFFRN_DETECTOR_AXIS, but usually differs in that the axes in this category are the axes of the coordinate system of reported data points, while the axes in DIFFRN_DETECTOR_AXIS are the physical axes of the detector describing the 'poise' of the detector as an overall physical object. If there is only one axis in the set, the identifier of that axis should be used as the identifier of the set.

The following item(s) have an equivalent role in their respective categories:
_array_structure_list_axis.axis_set_id.

[array_structure_list]

***_array_structure_list.dimension** (*int*)

The number of elements stored in the array structure in this dimension.

The permitted range is $[1, \infty)$.

[array_structure_list]

***_array_structure_list.direction** (*code*)

Identifies the direction in which this array index changes.

The data value must be one of the following:

`increasing`
: Indicates the index changes from 1 to the maximum dimension.

`decreasing`
: Indicates the index changes from the maximum dimension to 1.

[array_structure_list]

***_array_structure_list.index** (*int*)

Identifies the one-based index of the row or column in the array structure.

The following item(s) have an equivalent role in their respective categories:
_array_element_size.index.

The permitted range is $[1, \infty)$.

[array_structure_list]

***_array_structure_list.precedence** (*int*)

Identifies the rank order in which this array index changes with respect to other array indices. The precedence of 1 indicates the index which changes fastest.

The permitted range is $[1, \infty)$.

[array_structure_list]

ARRAY_STRUCTURE_LIST_AXIS

Data items in the ARRAY_STRUCTURE_LIST_AXIS category describe the physical settings of sets of axes for the centres of pixels that correspond to data points described in the ARRAY_STRUCTURE_LIST category. In the simplest cases, the physical increments of a single axis correspond to the increments of a single array index. More complex organizations, *e.g.* spiral scans, may require coupled motions along multiple axes. Note that a spiral scan uses two coupled axes: one for the angular direction and one for the radial direction. This differs from a cylindrical scan for which the two axes are not coupled into one set.

Category group(s): **inclusive_group**
array_data_group
Category key(s): **_array_structure_list_axis.axis_set_id**
_array_structure_list_axis.axis_id

_array_structure_list_axis.angle (*float*)
The setting of the specified axis in degrees for the first data point of the array index with the corresponding value of **_array_structure_list.axis_set_id**. If the index is specified as 'increasing', this will be the centre of the pixel with index value 1. If the index is specified as 'decreasing', this will be the centre of the pixel with maximum index value.
Where no value is given, the assumed value is '0.0'.
[array_structure_list_axis]

_array_structure_list_axis.angle_increment (*float*)
The pixel-centre-to-pixel-centre increment in the angular setting of the specified axis in degrees. This is not meaningful in the case of 'constant velocity' spiral scans and should not be specified for this case. See **_array_structure_list_axis.angular_pitch**.
Where no value is given, the assumed value is '0.0'.
[array_structure_list_axis]

_array_structure_list_axis.angular_pitch (*float*)
The pixel-centre-to-pixel-centre distance for a one-step change in the setting of the specified axis in millimetres. This is meaningful only for 'constant velocity' spiral scans or for uncoupled angular scans at a constant radius (cylindrical scans) and should not be specified for cases in which the angle between pixels (rather than the distance between pixels) is uniform. See **_array_structure_list_axis.angle_increment**.
Where no value is given, the assumed value is '0.0'.
[array_structure_list_axis]

***_array_structure_list_axis.axis_id** (*code*)
The value of this data item is the identifier of one of the axes in the set of axes for which settings are being specified. Multiple axes may be specified for the same value of **_array_structure_list_axis.axis_set_id**. This item is a pointer to **_axis.id** in the AXIS category.
[array_structure_list_axis]

(*)**_array_structure_list_axis.axis_set_id** (*code*)
The value of this data item is the identifier of the set of axes for which axis settings are being specified. Multiple axes may be specified for the same value of **_array_structure_list_axis.axis_set_id**. This item is a pointer to **_array_structure_list.axis_set_id** in the ARRAY_STRUCTURE_LIST category. If this item is not specified, it defaults to the corresponding axis identifier.
[array_structure_list_axis]

_array_structure_list_axis.displacement (*float*)
The setting of the specified axis in millimetres for the first data point of the array index with the corresponding value of **_array_structure_list.axis_set_id**. If the index is specified as 'increasing', this will be the centre of the pixel with index value 1. If the index is specified as 'decreasing', this will be the centre of the pixel with maximum index value.
Where no value is given, the assumed value is '0.0'.
[array_structure_list_axis]

_array_structure_list_axis.displacement_increme
(*float*)
The pixel-centre-to-pixel-centre increment for the displacement setting of the specified axis in millimetres.
Where no value is given, the assumed value is '0.0'.
[array_structure_list_axis]

_array_structure_list_axis.radial_pitch (*float*)
The radial distance from one 'cylinder' of pixels to the next in millimetres. If the scan is a 'constant velocity' scan with differing angular displacements between pixels, the value of this item may differ significantly from the value of **_array_structure_list_axis.displacement_increment**.
Where no value is given, the assumed value is '0.0'.
[array_structure_list_axis]

AXIS

Data items in the AXIS category record the information required to describe the various goniometer, detector, source and other axes needed to specify a data collection. The location of each axis is specified by two vectors: the axis itself, given as a unit vector, and an offset to the base of the unit vector. These vectors are referenced to a right-handed laboratory coordinate system with its origin in the sample or specimen:

```
         | Y (to complete right-handed system)
         |
         |
         |
         |
         |
         |________________X
        /       principal goniometer axis
       /
      /
     /
    /
   /Z (to source)
```

Axis 1 (X): The *X* axis is aligned to the mechanical axis pointing from the sample or specimen along the principal axis of the goniometer. Axis 2 (Y): The *Y* axis completes an orthogonal right-handed system defined by the *X* axis and the *Z* axis (see below). Axis 3 (Z): The *Z* axis is derived from the source axis which goes from the sample to the source. The *Z* axis is the component of the source axis in the direction of the source orthogonal to the *X* axis in the plane defined by the *X* axis and the source axis. These axes are based on the goniometer, not on the orientation of the detector, gravity *etc.* The vectors necessary to specify all other axes are given by sets of three components in the order (X, Y, *Z*). If the axis involved is a rotation axis, it is right-handed, *i.e.* as one views the object to be rotated from the origin (the tail) of the unit vector, the rotation is clockwise. If a translation axis is specified, the direction of the unit vector specifies the sense of positive translation. Note: This choice of coordinate system is similar to but significantly different from the choice in *MOSFLM* (Leslie & Powell, 2004). In *MOSFLM*, X is along the X-ray beam (the CBF/imgCIF *Z* axis) and *Z* is along the rotation axis. All rotations are given in degrees and all translations are given in mm.

AXIS

Axes may be dependent on one another. The *X* axis is the only goniometer axis the direction of which is strictly connected to the hardware. All other axes are specified by the positions they would assume when the axes upon which they depend are at their zero points. When specifying detector axes, the axis is given to the beam centre. The location of the beam centre on the detector should be given in the DIFFRN_DETECTOR category in distortion-corrected millimetres from the (0,0) corner of the detector. It should be noted that many different origins arise in the definition of an experiment. In particular, as noted above, it is necessary to specify the location of the beam centre on the detector in terms of the origin of the detector, which is, of course, not coincident with the centre of the sample.

Reference: Leslie, A. G. W. & Powell, H. (2004). *MOSFLM* v6.11. MRC Laboratory of Molecular Biology, Hills Road, Cambridge, England. http://www.CCP4.ac.uk/dist/x-windows/Mosflm/.

Category group(s): **inclusive_group**
axis_group
diffrn_group
Category key(s): **_axis.id**
_axis.equipment

Example 1 -
This example shows the axis specification of the axes of a kappa- geometry goniometer [see Stout, G. H. & Jensen, L. H. (1989). X-ray structure determination. A practical guide, 2nd ed. p. 134. New York: Wiley Interscience].
There are three axes specified, and no offsets. The outermost axis, omega, is pointed along the X axis. The next innermost axis, kappa, is at a 50 degree angle to the X axis, pointed away from the source. The innermost axis, phi, aligns with the X axis when omega and phi are at their zero points. If T-omega, T-kappa and T-phi are the transformation matrices derived from the axis settings, the complete transformation would be: x' = (T-omega) (T-kappa) (T-phi) x

```
     loop_
    _axis.id
    _axis.type
    _axis.equipment
    _axis.depends_on
    _axis.vector[1] _axis.vector[2] _axis.vector[3]
    omega rotation goniometer     .    1        0        0
    kappa rotation goniometer omega    -.64279  0
-.76604
    phi   rotation goniometer kappa    1        0        0
```

Example 2 -
This example show the axis specification of the axes of a detector, source and gravity. The order has been changed as a reminder that the ordering of presentation of tokens is not significant. The centre of rotation of the detector has been taken to be 68 millimetres in the direction away from the source.

```
    loop_
    _axis.id
    _axis.type
    _axis.equipment
    _axis.depends_on
    _axis.vector[1] _axis.vector[2] _axis.vector[3]
    _axis.offset[1] _axis.offset[2] _axis.offset[3]
    source       .          source     .        0     0     1
. . .
    gravity      .          gravity    .        0    -1     0
. . .
    tranz     translation detector rotz     0     0     1
0 0 -68
    twotheta  rotation    detector   .        1     0     0
. . .
    roty      rotation    detector twotheta  0     1     0
0 0 -68
    rotz      rotation    detector roty      0     0     1
0 0 -68
```

_axis.depends_on

The value of **_axis.depends_on** specifies the next outermost axis upon which this axis depends. This item is a pointer to **_axis.id** in the same category.

[axis]

_axis.equipment (*ucode*)

The value of **_axis.equipment** specifies the type of equipment using the axis: 'goniometer', 'detector', 'gravity', 'source' or 'general'.

The data value must be one of the following:

`goniometer`
: equipment used to orient or position samples

`detector`
: equipment used to detect reflections

`general`
: equipment used for general purposes

`gravity`
: axis specifying the downward direction

`source`
: axis specifying the direction sample to source

Where no value is given, the assumed value is '`general`'.

[axis]

*_axis.id (*code*)

The value of **_axis.id** must uniquely identify each axis relevant to the experiment. Note that multiple pieces of equipment may share the same axis (*e.g.* a twotheta arm), so the category key for AXIS also includes the equipment.

The following item(s) have an equivalent role in their respective categories:
_axis.depends_on,
_array_structure_list_axis.axis_id,
_diffrn_detector_axis.axis_id,
_diffrn_measurement_axis.axis_id,
_diffrn_scan_axis.axis_id,
_diffrn_scan_frame_axis.axis_id. **[axis]**

_axis.offset[1] (*float*)

The [1] element of the three-element vector used to specify the offset to the base of a rotation or translation axis. The vector is specified in millimetres.

Where no value is given, the assumed value is '`0.0`'. **[axis]**

_axis.offset[2] (*float*)

The [2] element of the three-element vector used to specify the offset to the base of a rotation or translation axis. The vector is specified in millimetres.

Where no value is given, the assumed value is '`0.0`'. **[axis]**

_axis.offset[3] (*float*)

The [3] element of the three-element vector used to specify the offset to the base of a rotation or translation axis. The vector is specified in millimetres.

Where no value is given, the assumed value is '`0.0`'. **[axis]**

_axis.type (*ucode*)

The value of **_axis.type** specifies the type of axis: 'rotation' or 'translation' (or 'general' when the type is not relevant, as for gravity).

The data value must be one of the following:

`rotation`
: right-handed axis of rotation

`translation`
: translation in the direction of the axis

`general`
: axis for which the type is not relevant

Where no value is given, the assumed value is '`general`'.

[axis]

_axis.vector[1] (*float*)

The [1] element of the three-element vector used to specify the direction of a rotation or translation axis. The vector should be normalized to be a unit vector and is dimensionless.

Where no value is given, the assumed value is '`0.0`'. **[axis]**

_axis.vector[2] (*float*)

The [2] element of the three-element vector used to specify the direction of a rotation or translation axis. The vector should be normalized to be a unit vector and is dimensionless.

Where no value is given, the assumed value is '`0.0`'. **[axis]**

_axis.vector[3] (*float*)

The [3] element of the three-element vector used to specify the direction of a rotation or translation axis. The vector should be normalized to be a unit vector and is dimensionless.

Where no value is given, the assumed value is '0.0'. **[axis]**

DIFFRN_DATA_FRAME

Data items in the DIFFRN_DATA_FRAME category record the details about each frame of data. The items in this category were previously in a DIFFRN_FRAME_DATA category, which is now deprecated. The items from the old category are provided as aliases but should not be used for new work.

Category group(s): **inclusive_group**
array_data_group

Category key(s): **_diffrn_data_frame.id**
_diffrn_data_frame.detector_element_id

Example 1 – A frame containing data from 4 frame elements. Each frame element has a common array configuration 'array_1' described in ARRAY_STRUCTURE and related categories. The data for each detector element are stored in four groups of binary data in the ARRAY_DATA category, linked by the array_id and binary_id.

```
loop_
_diffrn_data_frame.id
_diffrn_data_frame.detector_element_id
_diffrn_data_frame.array_id
_diffrn_data_frame.binary_id
frame_1   d1_ccd_1  array_1  1
frame_1   d1_ccd_2  array_1  2
frame_1   d1_ccd_3  array_1  3
frame_1   d1_ccd_4  array_1  4
```

*__diffrn_data_frame.array_id** (*code*)

_diffrn_frame_data.array_id (*cif_img.dic 1.0*)

This item is a pointer to **_array_structure.id** in the ARRAY_STRUCTURE category.

[diffrn_data_frame]

(*)**_diffrn_data_frame.binary_id** (*int*)

_diffrn_frame_data.binary_id (*cif_img.dic 1.0*)

This item is a pointer to **_array_data.binary_id** in the ARRAY_DATA category.

[diffrn_data_frame]

_diffrn_data_frame.details (*text*)

_diffrn_frame_data.details (*cif_img.dic 1.4*)

The value of **_diffrn_data_frame.details** should give a description of special aspects of each frame of data. This is an appropriate location in which to record information from vendor headers as presented in those headers, but it should never be used as a substitute for providing the fully parsed information within the appropriate imgCIF/CBF categories.

Example:

```
;
  HEADER_BYTES = 512;
  DIM = 2;
  BYTE_ORDER = big_endian;
  TYPE = unsigned_short;
  SIZE1 = 3072;
  SIZE2 = 3072;
  PIXEL_SIZE = 0.102588;
  BIN = 2x2;
  DETECTOR_SN = 901;
  TIME = 29.945155;
  DISTANCE = 200.000000;
  PHI = 85.000000;
  OSC_START = 85.000000;
  OSC_RANGE = 1.000000;
  WAVELENGTH = 0.979381;
  BEAM_CENTER_X = 157.500000;
  BEAM_CENTER_Y = 157.500000;
  PIXEL SIZE = 0.102588;
  OSCILLATION RANGE = 1;
  EXPOSURE TIME = 29.9452;
  TWO THETA = 0;
  BEAM CENTRE = 157.5 157.5;
;
```

(Example of header information extracted from an ADSC Quantum 315 detector header by CBFlib_0.7.6. Image provided by Chris Nielsen of ADSC from a data collection at SSRL beamline 1-5.)

[diffrn_data_frame]

*__diffrn_data_frame.detector_element_id** (*code*)

_diffrn_frame_data.detector_element_id (*cif_img.dic 1.0*)

This item is a pointer to **_diffrn_detector_element.id** in the DIFFRN_DETECTOR_ELEMENT category.

[diffrn_data_frame]

*__diffrn_data_frame.id** (*code*)

_diffrn_frame_data.id (*cif_img.dic 1.0*)

The value of **_diffrn_data_frame.id** must uniquely identify each complete frame of data.

The following item(s) have an equivalent role in their respective categories:

_diffrn_refln.frame_id,
_diffrn_scan.frame_id_start,
_diffrn_scan.frame_id_end,
_diffrn_scan_frame.frame_id,
_diffrn_scan_frame_axis.frame_id. **[diffrn_data_frame]**

DIFFRN_DETECTOR

Data items in the DIFFRN_DETECTOR category describe the detector used to measure the scattered radiation, including any analyser and post-sample collimation.

Category group(s): **inclusive_group**
diffrn_group

Category key(s): **_diffrn_detector.diffrn_id**
_diffrn_detector.id

Example 1 – based on PDB entry 5HVP and laboratory records for the structure corresponding to PDB entry 5HVP.

```
_diffrn_detector.diffrn_id             'd1'
_diffrn_detector.detector              'multiwire'
_diffrn_detector.type                  'Siemens'
```

_diffrn_detector.details (*text*)

_diffrn_detector_details (*cif_core.dic 2.0.1*)

A description of special aspects of the radiation detector.

Example: 'slow mode'. **[diffrn_detector]**

_diffrn_detector.detector (*text*)

_diffrn_radiation_detector (*cifdic.c91 1.0*)

_diffrn_detector (*cif_core.dic 2.0*)

The general class of the radiation detector.

Examples: 'photographic film', 'scintillation counter', 'CCD plate', 'BF~3~ counter'. **[diffrn_detector]**

*__diffrn_detector.diffrn_id** (*code*)

This data item is a pointer to **_diffrn.id** in the DIFFRN category. The value of **_diffrn.id** uniquely defines a set of diffraction data.

_diffrn_detector.dtime (*float*)

_diffrn_radiation_detector_dtime (*cifdic.c91 1.0*)

_diffrn_detector_dtime (*cif_core.dic 2.0*)

The deadtime in microseconds of the detector(s) used to measure the diffraction intensities.

The permitted range is $[0.0, \infty)$. **[diffrn_detector]**

(*)**_diffrn_detector.id** (*code*)

The value of **_diffrn_detector.id** must uniquely identify each detector used to collect each diffraction data set. If the value of **_diffrn_detector.id** is not given, it is implicitly equal to the value of **_diffrn_detector.diffrn_id**.

The following item(s) have an equivalent role in their respective categories:

_diffrn_detector_axis.detector_id. **[diffrn_detector]**

_diffrn_detector.number_of_axes (*int*)

The value of **_diffrn_detector.number_of_axes** gives the number of axes of the positioner for the detector identified by **_diffrn_detector.id**. The word 'positioner' is a general term used in instrumentation design for devices that are used to change the positions of portions of apparatus by linear translation, rotation or combinations of such motions. Axes which are used to provide a coordinate system for the face of an area detetctor should not be counted for this data item. The description of each axis should be provided by entries in DIFFRN_DETECTOR_AXIS.

The permitted range is $[1, \infty)$. **[diffrn_detector]**

_diffrn_detector.type (*text*)
_diffrn_detector_type (*cif_core.dic 2.0.1*)
The make, model or name of the detector device used.
[diffrn_detector]

DIFFRN_DETECTOR_AXIS

Data items in the DIFFRN_DETECTOR_AXIS category associate axes with detectors.
Category group(s): **inclusive_group**
diffrn_group
Category key(s): **_diffrn_detector_axis.detector_id**
_diffrn_detector_axis.axis_id

***_diffrn_detector_axis.axis_id** (*code*)
This data item is a pointer to **_axis.id** in the AXIS category.
[diffrn_detector_axis]

***_diffrn_detector_axis.detector_id** (*code*)
_diffrn_detector_axis.id (*cif_img.dic 1.0*)
This data item is a pointer to **_diffrn_detector.id** in the DIFFRN_DETECTOR category. This item was previously named **_diffrn_detector_axis.id** which is now a deprecated name. The old name is provided as an alias but should not be used for new work.
[diffrn_detector_axis]

***_diffrn_detector_axis.id** (*code*)
This data item is a pointer to **_diffrn_detector.id** in the DIFFRN_DETECTOR category. *Deprecated: do not use.*
[diffrn_detector_axis]

DIFFRN_DETECTOR_ELEMENT

Data items in the DIFFRN_DETECTOR_ELEMENT category record the details about spatial layout and other characteristics of each element of a detector which may have multiple elements. In most cases, giving more detailed information in ARRAY_STRUCTURE_LIST and ARRAY_STRUCTURE_LIST_AXIS is preferable to simply providing the centre of the detector element.
Category group(s): **inclusive_group**
array_data_group
Category key(s): **_diffrn_detector_element.id**
_diffrn_detector_element.detector_id

Example 1 – Detector d1 is composed of four CCD detector elements, each 200 mm by 200 mm, arranged in a square, in the pattern
*1 2 * 3 4*
Note that the beam centre is slightly displaced from each of the detector elements, just beyond the lower right corner of 1, the lower left corner of 2, the upper right corner of 3 and the upper left corner of 4.

```
loop_
_diffrn_detector_element.detector_id
_diffrn_detector_element.id
_diffrn_detector_element.center[1]
_diffrn_detector_element.center[2]
d1     d1_ccd_1  201.5 -1.5
d1     d1_ccd_2  -1.8  -1.5
d1     d1_ccd_3  201.6 201.4
d1     d1_ccd_4  -1.7  201.5
```

_diffrn_detector_element.center[1] (*float*)
The value of **_diffrn_detector_element.center[1]** is the X component of the distortion-corrected beam centre in millimetres from the (0, 0) (lower-left) corner of the detector element viewed from the sample side. The X and Y axes are the laboratory coordinate system coordinates defined in the AXIS category measured when all positioning axes for the detector are at their zero settings. If the resulting X or *Y* axis is then orthogonal to the detector, the *Z* axis is used instead of the orthogonal axis.
Where no value is given, the assumed value is '0.0'. **[diffrn_detector_element]**

_diffrn_detector_element.center[2] (*float*)
The value of **_diffrn_detector_element.center[2]** is the Y component of the distortion-corrected beam centre in millimetres from the (0, 0) (lower-left) corner of the detector element viewed from the sample side. The X and Y axes are the laboratory coordinate system coordinates defined in the AXIS category measured when all positioning axes for the detector are at their zero settings. If the resulting X or *Y* axis is then orthogonal to the detector, the *Z* axis is used instead of the orthogonal axis.
Where no value is given, the assumed value is '0.0'. **[diffrn_detector_element]**

***_diffrn_detector_element.detector_id** (*code*)
This item is a pointer to **_diffrn_detector.id** in the DIFFRN_DETECTOR category.
[diffrn_detector_element]

***_diffrn_detector_element.id** (*code*)
The value of **_diffrn_detector_element.id** must uniquely identify each element of a detector.
[diffrn_detector_element]

DIFFRN_FRAME_DATA

Data items in the DIFFRN_FRAME_DATA category record the details about each frame of data. The items in this category are now in the DIFFRN_DATA_FRAME category. The items in the DIFFRN_FRAME_DATA category are now deprecated. The items from this category are provided as aliases in the 1.0 dictionary or, in the case of **_diffrn_frame_data.details**, in the 1.4 dictionary. THESE ITEMS SHOULD *not* BE USED FOR NEW WORK. The items from the old category are provided in this dictionary for completeness but should not be used or cited. To avoid confusion, the example has been removed and the redundant parent-child links to other categories have been removed.
Category group(s): **inclusive_group**
array_data_group
Category key(s): **_diffrn_frame_data.id**
_diffrn_frame_data.detector_element_id

THE DIFFRN_FRAME_DATA category is deprecated and should not be used.

```
# EXAMPLE REMOVED #
```

***_diffrn_frame_data.array_id** (*code*)
This item is a pointer to **_array_structure.id** in the ARRAY_STRUCTURE category. *Deprecated: do not use.*
[diffrn_frame_data]

(*)**_diffrn_frame_data.binary_id** (*int*)
This item is a pointer to **_array_data.binary_id** in the ARRAY_STRUCTURE category. *Deprecated: do not use.*
[diffrn_frame_data]

_diffrn_frame_data.details (*text*)
The value of **_diffrn_data_frame.details** should give a description of special aspects of each frame of data. *Deprecated: do not use.*
[diffrn_frame_data]

***_diffrn_frame_data.detector_element_id** (*code*)
This item is a pointer to **_diffrn_detector_element.id** in the DIFFRN_DETECTOR_ELEMENT category. *Deprecated: do not use.*
[diffrn_frame_data]

***_diffrn_frame_data.id** (*code*)
The value of **_diffrn_frame_data.id** must uniquely identify each complete frame of data. *Deprecated: do not use.*
[diffrn_frame_data]

DIFFRN_MEASUREMENT

Data items in the DIFFRN_MEASUREMENT category record details about the device used to orient and/or position the crystal during data measurement and the manner in which the diffraction data were measured.

Category group(s): **inclusive_group**
diffrn_group

Category key(s): **_diffrn_measurement.device**
_diffrn_measurement.diffrn_id
_diffrn_measurement.id

Example 1 – based on PDB entry 5HVP and laboratory records for the structure corresponding to PDB entry 5HVP

```
_diffrn_measurement.diffrn_id           'd1'
_diffrn_measurement.device              '3-circle camera'
_diffrn_measurement.device_type         'Supper model x'
_diffrn_measurement.device_details      'none'
_diffrn_measurement.method              'omega scan'
_diffrn_measurement.details
; 440 frames, 0.20 degrees, 150 sec, detector distance 12 cm,
detector
  angle 22.5 degrees
;
```

Example 2 – based on data set TOZ of Willis, Beckwith & Tozer [Acta Cryst. (1991), C47, 2276-2277].

```
_diffrn_measurement.diffrn_id        's1'
_diffrn_measurement.device_type      'Philips PW1100/20
diffractometer'
_diffrn_measurement.method           'theta/2theta (\q/2\q)'
```

_diffrn_measurement.details (*text*)

_diffrn_measurement_details (*cif_core.dic 2.0.1*)

A description of special aspects of the intensity measurement.

Example:

```
; 440 frames, 0.20 degrees, 150 sec, detector
  distance 12 cm, detector angle 22.5 degrees
;
```

[diffrn_measurement]

(*)_diffrn_measurement.device (*text*)

_diffrn_measurement_device (*cif_core.dic 2.0.1*)

The general class of goniometer or device used to support and orient the specimen. If the value of **_diffrn_measurement.device** is not given, it is implicitly equal to the value of **_diffrn_measurement.diffrn_id**. Either **_diffrn_measurement.device** or **_diffrn_measurement.id** may be used to link to other categories. If the experimental setup admits multiple devices, then **_diffrn_measurement.id** is used to provide a unique link.

The following item(s) have an equivalent role in their respective categories:

_diffrn_measurement_axis.measurement_device.

Examples: '3-circle camera', '4-circle camera', 'kappa-geometry camera', 'oscillation camera', 'precession camera'.

[diffrn_measurement]

_diffrn_measurement.device_details (*text*)

_diffrn_measurement_device_details (*cif_core.dic 2.0.1*)

A description of special aspects of the device used to measure the diffraction intensities.

Example:

```
; commercial goniometer modified locally to
  allow for 90\% \t arc
;
```

[diffrn_measurement]

_diffrn_measurement.device_type (*text*)

_diffrn_measurement_device_type (*cif_core.dic 2.0.1*)

The make, model or name of the measurement device (goniometer) used.

Examples: 'Supper model q', 'Huber model r', 'Enraf-Nonius model s', 'home-made'. **[diffrn_measurement]**

*_diffrn_measurement.diffrn_id (*code*)

This data item is a pointer to **_diffrn.id** in the DIFFRN category.

(*)_diffrn_measurement.id (*code*)

The value of **_diffrn_measurement.id** must uniquely identify the set of mechanical characteristics of the device used to orient and/or position the sample used during the collection of each diffraction data set. If the value of **_diffrn_measurement.id** is not given, it is implicitly equal to the value of **_diffrn_measurement.diffrn_id**. Either **_diffrn_measurement.device** or **_diffrn_measurement.id** may be used to link to other categories. If the experimental setup admits multiple devices, then **_diffrn_measurement.id** is used to provide a unique link.

The following item(s) have an equivalent role in their respective categories:

_diffrn_measurement_axis.measurement_id. **[diffrn_measurement]**

_diffrn_measurement.method (*text*)

_diffrn_measurement_method (*cif_core.dic 2.0.1*)

Method used to measure intensities.

Example: 'profile data from theta/2theta (\q/2\q) scans'.

[diffrn_measurement]

_diffrn_measurement.number_of_axes (*int*)

The value of **_diffrn_measurement.number_of_axes** gives the number of axes of the positioner for the goniometer or other sample orientation or positioning device identified by **_diffrn_measurement.id**. The description of the axes should be provided by entries in DIFFRN_MEASUREMENT_AXIS.

The permitted range is $[1, \infty)$. **[diffrn_measurement]**

_diffrn_measurement.specimen_support (*text*)

_diffrn_measurement_specimen_support (*cif_core.dic 2.0.1*)

The physical device used to support the crystal during data collection.

Examples: 'glass capillary', 'quartz capillary', 'fiber', 'metal loop'.

[diffrn_measurement]

DIFFRN_MEASUREMENT_AXIS

Data items in the DIFFRN_MEASUREMENT_AXIS category associate axes with goniometers.

Category group(s): **inclusive_group**
diffrn_group

Category key(s): **_diffrn_measurement_axis.measurement_device**
_diffrn_measurement_axis.measurement_id
_diffrn_measurement_axis.axis_id

*_diffrn_measurement_axis.axis_id (*code*)

This data item is a pointer to **_axis.id** in the AXIS category.

[diffrn_measurement_axis]

*_diffrn_measurement_axis.id (*code*)

This data item is a pointer to **_diffrn_measurement.id** in the DIFFRN_MEASUREMENT category. *Deprecated: do not use.*

[diffrn_measurement_axis]

(*)_diffrn_measurement_axis.measurement_device (*text*)

This data item is a pointer to **_diffrn_measurement.device** in the DIFFRN_MEASUREMENT category.

[diffrn_measurement_axis]

(*)_diffrn_measurement_axis.measurement_id (*code*)

_diffrn_measurement_axis.id (*cif_img.dic 1.0*)

This data item is a pointer to **_diffrn_measurement.id** in the DIFFRN_MEASUREMENT category. This item was previously named **_diffrn_measurement_axis.id**, which is now a deprecated name. The old name is provided as an alias but should not be used for new work.

[diffrn_measurement_axis]

DIFFRN_RADIATION

Data items in the DIFFRN_RADIATION category describe the radiation used for measuring diffraction intensities, its collimation and monochromatization before the sample. Post-sample treatment of the beam is described by data items in the DIFFRN_DETECTOR category.

Category group(s): **inclusive_group**
diffrn_group

Category key(s): **_diffrn_radiation.diffrn_id**

Example 1 – based on PDB entry 5HVP and laboratory records for the structure corresponding to PDB entry 5HVP

```
_diffrn_radiation.diffrn_id            'set1'

_diffrn_radiation.collimation          '0.3 mm double
pinhole'
_diffrn_radiation.monochromator        'graphite'
_diffrn_radiation.type                 'Cu K\a'
_diffrn_radiation.wavelength_id         1
```

Example 2 – based on data set TOZ of Willis, Beckwith & Tozer [Acta Cryst. (1991), C47, 2276-2277].

```
_diffrn_radiation.wavelength_id    1
_diffrn_radiation.type             'Cu K\a'
_diffrn_radiation.monochromator    'graphite'
```

_diffrn_radiation.collimation (*text*)
_diffrn_radiation_collimation (*cif_core.dic 2.0.1*)
The collimation or focusing applied to the radiation.
Examples: '0.3 mm double-pinhole', '0.5 mm', 'focusing mirrors'.
[diffrn_radiation]

***_diffrn_radiation.diffrn_id** (*code*)
This data item is a pointer to **_diffrn.id** in the DIFFRN category.

_diffrn_radiation.div_x_source (*float*)
Beam crossfire in degrees parallel to the laboratory X axis (see AXIS category). This is a characteristic of the X-ray beam as it illuminates the sample (or specimen) after all monochromation and collimation. This is the standard uncertainty (e.s.d.) of the directions of photons in the XZ plane around the mean source beam direction. Note that for some synchrotrons this value is specified in milliradians, in which case a conversion is needed. To convert a value in milliradians to a value in degrees, multiply by 0.180 and divide by π.
[diffrn_radiation]

_diffrn_radiation.div_x_y_source (*float*)
Beam crossfire correlation in degrees squared between the crossfire laboratory X axis component and the crossfire laboratory Y axis component (see AXIS category). This is a characteristic of the X-ray beam as it illuminates the sample (or specimen) after all monochromation and collimation. This is the mean of the products of the deviations of the direction of each photon in XZ plane times the deviations of the direction of the same photon in the YZ plane around the mean source beam direction. This will be zero for uncorrelated crossfire. Note that for some synchrotrons this value is specified in milliradians squared, in which case a conversion is needed. To convert a value in milliradians squared to a value in degrees squared, multiply by 0.180^2 and divide by π^2.
Where no value is given, the assumed value is '0.0'. **[diffrn_radiation]**

_diffrn_radiation.div_y_source (*float*)
Beam crossfire in degrees parallel to the laboratory Y axis (see AXIS category). This is a characteristic of the X-ray beam as it illuminates the sample (or specimen) after all monochromation and collimation. This is the standard uncertainty (e.s.d.) of the directions of photons in the YZ plane around the mean source beam direction. Note that for some synchrotrons this value is specified in milliradians, in which case a conversion is needed. To convert a value in milliradians to a value in degrees, multiply by 0.180 and divide by π.
Where no value is given, the assumed value is '0.0'. **[diffrn_radiation]**

_diffrn_radiation.filter_edge (*float*)
_diffrn_radiation_filter_edge (*cif_core.dic 2.0.1*)
Absorption edge in %Ångströms of the radiation filter used.
The permitted range is $[0.0, \infty)$. **[diffrn_radiation]**

_diffrn_radiation.inhomogeneity (*float*)
_diffrn_radiation_inhomogeneity (*cif_core.dic 2.0.1*)
Half-width in millimetres of the incident beam in the direction perpendicular to the diffraction plane.
The permitted range is $[0.0, \infty)$. **[diffrn_radiation]**

_diffrn_radiation.monochromator (*text*)
_diffrn_radiation_monochromator (*cif_core.dic 2.0.1*)
The method used to obtain monochromatic radiation. If a monochromator crystal is used, the material and the indices of the Bragg reflection are specified.
Examples: 'Zr filter', 'Ge 220', 'none', 'equatorial mounted graphite'.
[diffrn_radiation]

_diffrn_radiation.polarisn_norm (*float*)
_diffrn_radiation_polarisn_norm (*cif_core.dic 2.0.1*)
The angle in degrees, as viewed from the specimen, between the perpendicular component of the polarization and the diffraction plane. See **_diffrn_radiation_polarisn_ratio**.
The permitted range is $[-90.0, 90.0]$. **[diffrn_radiation]**

_diffrn_radiation.polarisn_ratio (*float*)
_diffrn_radiation_polarisn_ratio (*cif_core.dic 2.0.1*)
Polarization ratio of the diffraction beam incident on the crystal. This is the ratio of the perpendicularly polarized to the parallel polarized component of the radiation. The perpendicular component forms an angle of **_diffrn_radiation.polarisn_norm** to the normal to the diffraction plane of the sample (*i.e.* the plane containing the incident and reflected beams).
The permitted range is $[0.0, \infty)$. **[diffrn_radiation]**

_diffrn_radiation.polarizn_source_norm (*float*)
The angle in degrees, as viewed from the specimen, between the normal to the polarization plane and the laboratory Y axis as defined in the AXIS category. Note that this is the angle of polarization of the source photons, either directly from a synchrotron beamline or from a monochromater. This differs from the value of **_diffrn_radiation.polarisn_norm** in that **_diffrn_radiation.polarisn_norm** refers to polarization relative to the diffraction plane rather than to the laboratory axis system. In the case of an unpolarized beam, or a beam with true circular polarization, in which no single plane of polarization can be determined, the plane should be taken as the XZ plane and the angle as 0. See **_diffrn_radiation.polarizn_source_ratio**.
The permitted range is $[-90.0, 90.0]$. Where no value is given, the assumed value is '0.0'.
[diffrn_radiation]

_diffrn_radiation.polarizn_source_ratio (*float*)
$(I_p - I_n)/(I_p + I_n)$, where I_p is the intensity (amplitude squared) of the electric vector in the plane of polarization and I_n is the intensity (amplitude squared) of the electric vector in the plane of the normal to the plane of polarization. In the case of an unpolarized beam, or a beam with true circular polarization, in which no single plane of polarization can be determined, the plane is to be taken as the XZ plane and the normal is parallel to the Y axis. Thus, if there was complete polarization in the plane of polarization, the value of **_diffrn_radiation.polarizn_source_ratio** would be 1, and for an unpolarized beam **_diffrn_radiation.polarizn_source_ra** would have a value of 0. If the X axis has been chosen to lie in the plane of polarization, this definition will agree with the definition of 'MONOCHROMATOR' in the *Denzo* glossary, and values of near 1 should be expected for a bending-magnet source. However, if the X axis were perpendicular to the polarization plane (not a common choice), then the *Denzo* value would be the negative of **_diffrn_radiation.polarizn_source_ratio**. [See http://www.hkl-xray.com for information on *Denzo*, and Otwinowski & Minor (1997).] This differs both in the choice of ratio and choice of orientation from **_diffrn_radiation.polarisn_ratio**, which, unlike **_diffrn_radiation.polarizn_source_ratio**, is unbounded.

Reference: Otwinowski, Z. & Minor, W. (1997). *Processing of X-ray diffraction data collected in oscillation mode. Methods Enzymol.* **276**, 307–326.

The permitted range is [−1.0, 1.0]. **[diffrn_radiation]**

_diffrn_radiation.probe (*line*)

_diffrn_radiation_probe (*cif_core.dic 2.0.1*)

Name of the type of radiation used. It is strongly recommended that this be given so that the probe radiation is clearly specified.

The data value must be one of the following:

x-ray
neutron
electron
gamma **[diffrn_radiation]**

_diffrn_radiation.type (*line*)

_diffrn_radiation_type (*cif_core.dic 2.0.1*)

The nature of the radiation. This is typically a description of the X-ray wavelength in Siegbahn notation.

Examples: 'CuK\a', 'Cu K\a~1~', 'Cu K-L~2,3~', 'white-beam'.

[diffrn_radiation]

*__diffrn_radiation.wavelength_id** (*code*)

This data item is a pointer to **_diffrn_radiation_wavelength.id** in the DIFFRN_RADIATION_WAVELENGTH category.

[diffrn_radiation]

_diffrn_radiation.xray_symbol (*line*)

_diffrn_radiation_xray_symbol (*cif_core.dic 2.0.1*)

The IUPAC symbol for the X-ray wavelength for the probe radiation.

The data value must be one of the following:

K-L~3~
: Kα_1 in older Siegbahn notation

K-L~2~
: Kα_2 in older Siegbahn notation

K-M~3~
: Kβ in older Siegbahn notation

K-L~2,3~
: use where K-L_3 and K-L_2 are not resolved

[diffrn_radiation]

DIFFRN_REFLN

This category redefinition has been added to extend the key of the standard DIFFRN_REFLN category.

Category group(s): **inclusive_group**
diffrn_group

Category key(s): **_diffrn_refln.frame_id**

*__diffrn_refln.frame_id** (*code*)

This item is a pointer to **_diffrn_data_frame.id** in the DIFFRN_DATA_FRAME category.

[diffrn_refln]

DIFFRN_SCAN

Data items in the DIFFRN_SCAN category describe the parameters of one or more scans, relating axis positions to frames.

Category group(s): **inclusive_group**
diffrn_group

Category key(s): **_diffrn_scan.id**

Example 1 – derived from a suggestion by R. M. Sweet.

The vector of each axis is not given here, because it is provided in the AXIS category. By making ***_diffrn_scan_axis.scan_id*** *and* ***_diffrn_scan_axis.axis_id*** *keys of the DIFFRN_SCAN_AXIS category, an arbitrary number of scanning and fixed axes can be specified for a scan. In this example, three rotation axes and one translation axis at nonzero values are specified, with one axis stepping. There is no reason why more axes could not have been specified to step. Range information has been specified, but note that it can be calculated from the number of frames and the increment, so the data item* ***_diffrn_scan_axis.angle_range*** *could be dropped.*

Both the sweep data and the data for a single frame are specified.

Note that the information on how the axes are stepped is given twice, once in terms of the overall averages in the value of ***_diffrn_scan.integration_time*** *and the values for DIFFRN_SCAN_AXIS, and precisely for the given frame in the value for* ***_diffrn_scan_frame.integration_time*** *and the values for DIFFRN_SCAN_FRAME_AXIS. If dose-related adjustments are made to scan times and nonlinear stepping is done, these values may differ. Therefore, in interpreting the data for a particular frame it is important to use the frame-specific data.*

```
_diffrn_scan.id                    1
_diffrn_scan.date_start          '2001-11-18T03:26:42'
_diffrn_scan.date_end            '2001-11-18T03:36:45'
_diffrn_scan.integration_time     3.0
_diffrn_scan.frame_id_start       mad_L2_000
_diffrn_scan.frame_id_end         mad_L2_200
_diffrn_scan.frames               201

 loop_
_diffrn_scan_axis.scan_id
_diffrn_scan_axis.axis_id
_diffrn_scan_axis.angle_start
_diffrn_scan_axis.angle_range
_diffrn_scan_axis.angle_increment
_diffrn_scan_axis.displacement_start
_diffrn_scan_axis.displacement_range
_diffrn_scan_axis.displacement_increment

 1 omega 200.0 20.0 0.1 . . .
 1 kappa -40.0  0.0 0.0 . . .
 1 phi   127.5  0.0 0.0 . . .
 1 tranz  . . .   2.3 0.0 0.0

_diffrn_scan_frame.scan_id                  1
_diffrn_scan_frame.date                  '2001-11-18T03:27:33'
_diffrn_scan_frame.integration_time       3.0
_diffrn_scan_frame.frame_id               mad_L2_018
_diffrn_scan_frame.frame_number           18

loop_
_diffrn_scan_frame_axis.frame_id
_diffrn_scan_frame_axis.axis_id
_diffrn_scan_frame_axis.angle
_diffrn_scan_frame_axis.angle_increment
_diffrn_scan_frame_axis.displacement
_diffrn_scan_frame_axis.displacement_increment

 mad_L2_018 omega 201.8  0.1 . .
 mad_L2_018 kappa -40.0  0.0 . .
 mad_L2_018 phi   127.5  0.0 . .
 mad_L2_018 tranz  .     .  2.3 0.0
```

Example 2 – a more extensive example (R. M. Sweet, P. J. Ellis & H. J. Bernstein).

A detector is placed 240 mm along the Z axis from the goniometer. This leads to a choice: either the axes of the detector are defined at the origin, and then a Z setting of -240 is entered, or the axes are defined with the necessary Z offset. In this case, the setting is used and the offset is left as zero. This axis is called DETECTOR_Z.

The axis for positioning the detector in the Y direction depends on the detector Z axis. This axis is called DETECTOR_Y.

The axis for positioning the detector in the X direction depends on the detector Y axis (and therefore on the detector Z axis). This axis is called DETECTOR_X.

This detector may be rotated around the Y axis. This rotation axis depends on the three translation axes. It is called DETECTOR_PITCH.

A coordinate system is defined on the face of the detector in terms of 2300 0.150 mm pixels in each direction. The ELEMENT_X axis is used to index the first array index of the data array and the ELEMENT_Y axis is used to index the second array index. Because the pixels are 0.150mm x 0.150mm, the centre of the first pixel is at (0.075, 0.075) in this coordinate system.

```
###CBF: VERSION 1.1

 data_image_1

 # category DIFFRN
 _diffrn.id P6MB
 _diffrn.crystal_id P6MB_CRYSTAL7

 # category DIFFRN_SOURCE
 loop_
 _diffrn_source.diffrn_id
 _diffrn_source.source
 _diffrn_source.type
  P6MB synchrotron 'SSRL beamline 9-1'

 # category DIFFRN_RADIATION
 loop_
 _diffrn_radiation.diffrn_id
 _diffrn_radiation.wavelength_id
 _diffrn_radiation.monochromator
 _diffrn_radiation.polarizn_source_ratio
 _diffrn_radiation.polarizn_source_norm
 _diffrn_radiation.div_x_source
 _diffrn_radiation.div_y_source
 _diffrn_radiation.div_x_y_source
  P6MB WAVELENGTH1 'Si 111' 0.8 0.0 0.08
 0.01 0.00

 # category DIFFRN_RADIATION_WAVELENGTH
 loop_
 _diffrn_radiation_wavelength.id
 _diffrn_radiation_wavelength.wavelength
 _diffrn_radiation_wavelength.wt
  WAVELENGTH1 0.98 1.0

 # category DIFFRN_DETECTOR
 loop_
 _diffrn_detector.diffrn_id
 _diffrn_detector.id
 _diffrn_detector.type
 _diffrn_detector.number_of_axes
  P6MB MAR345-SN26 'MAR 345' 4

 # category DIFFRN_DETECTOR_AXIS
 loop_
 _diffrn_detector_axis.detector_id
 _diffrn_detector_axis.axis_id
  MAR345-SN26 DETECTOR_X
  MAR345-SN26 DETECTOR_Y
  MAR345-SN26 DETECTOR_Z
  MAR345-SN26 DETECTOR_PITCH

 # category DIFFRN_DETECTOR_ELEMENT
 loop_
 _diffrn_detector_element.id
 _diffrn_detector_element.detector_id
  ELEMENT1 MAR345-SN26

 # category DIFFRN_DATA_FRAME
 loop_
 _diffrn_data_frame.id
 _diffrn_data_frame.detector_element_id
 _diffrn_data_frame.array_id
 _diffrn_data_frame.binary_id
  FRAME1 ELEMENT1 ARRAY1 1
```

height

```
# category DIFFRN_MEASUREMENT
loop_
_diffrn_measurement.diffrn_id
_diffrn_measurement.id
_diffrn_measurement.number_of_axes
_diffrn_measurement.method
 P6MB GONIOMETER 3 rotation

# category DIFFRN_MEASUREMENT_AXIS
loop_
_diffrn_measurement_axis.measurement_id
_diffrn_measurement_axis.axis_id
 GONIOMETER GONIOMETER_PHI
 GONIOMETER GONIOMETER_KAPPA
 GONIOMETER GONIOMETER_OMEGA

# category DIFFRN_SCAN
loop_
_diffrn_scan.id
_diffrn_scan.frame_id_start
_diffrn_scan.frame_id_end
_diffrn_scan.frames
 SCAN1 FRAME1 FRAME1 1

# category DIFFRN_SCAN_AXIS
loop_
_diffrn_scan_axis.scan_id
_diffrn_scan_axis.axis_id
_diffrn_scan_axis.angle_start
_diffrn_scan_axis.angle_range
_diffrn_scan_axis.angle_increment
_diffrn_scan_axis.displacement_start
_diffrn_scan_axis.displacement_range
_diffrn_scan_axis.displacement_increment
 SCAN1 GONIOMETER_OMEGA 12.0 1.0 1.0 0.0 0.0 0.0
 SCAN1 GONIOMETER_KAPPA 23.3 0.0 0.0 0.0 0.0 0.0
 SCAN1 GONIOMETER_PHI -165.8 0.0 0.0 0.0 0.0 0.0
 SCAN1 DETECTOR_Z 0.0 0.0 0.0 -240.0 0.0 0.0
 SCAN1 DETECTOR_Y 0.0 0.0 0.0 0.6 0.0 0.0
 SCAN1 DETECTOR_X 0.0 0.0 0.0 -0.5 0.0 0.0
 SCAN1 DETECTOR_PITCH 0.0 0.0 0.0 0.0 0.0 0.0

# category DIFFRN_SCAN_FRAME
loop_
_diffrn_scan_frame.frame_id
_diffrn_scan_frame.frame_number
_diffrn_scan_frame.integration_time
_diffrn_scan_frame.scan_id
_diffrn_scan_frame.date
 FRAME1 1 20.0 SCAN1 1997-12-04T10:23:48

# category DIFFRN_SCAN_FRAME_AXIS
loop_
_diffrn_scan_frame_axis.frame_id
_diffrn_scan_frame_axis.axis_id
_diffrn_scan_frame_axis.angle
_diffrn_scan_frame_axis.displacement
 FRAME1 GONIOMETER_OMEGA 12.0 0.0
 FRAME1 GONIOMETER_KAPPA 23.3 0.0
 FRAME1 GONIOMETER_PHI -165.8 0.0
 FRAME1 DETECTOR_Z 0.0 -240.0
 FRAME1 DETECTOR_Y 0.0 0.6
 FRAME1 DETECTOR_X 0.0 -0.5
 FRAME1 DETECTOR_PITCH 0.0 0.0
```

```
# category AXIS
loop_
_axis.id
_axis.type
_axis.equipment
_axis.depends_on
_axis.vector[1] _axis.vector[2] _axis.vector[3]
_axis.offset[1] _axis.offset[2] _axis.offset[3]
 GONIOMETER_OMEGA rotation goniometer . 1 0 0 . . .
 GONIOMETER_KAPPA rotation goniometer GONIOMETER_OMEGA
0.64279
 0 0.76604 . . .
 GONIOMETER_PHI   rotation goniometer GONIOMETER_KAPPA 1 0 0
. . .
 SOURCE           general source . 0 0 1 . . .
 GRAVITY          general gravity . 0 -1 0 . . .
 DETECTOR_Z       translation detector . 0 0 1 0 0 0
 DETECTOR_Y       translation detector DETECTOR_Z 0 1 0 0 0
0
 DETECTOR_X       translation detector DETECTOR_Y 1 0 0 0 0
0
 DETECTOR_PITCH   rotation    detector DETECTOR_X 0 1 0 0 0
0
 ELEMENT_X        translation detector DETECTOR_PITCH
1 0 0 172.43 -172.43 0
 ELEMENT_Y        translation detector ELEMENT_X
0 1 0 0 0 0

# category ARRAY_STRUCTURE_LIST
loop_
_array_structure_list.array_id
_array_structure_list.index
_array_structure_list.dimension
_array_structure_list.precedence
_array_structure_list.direction
_array_structure_list.axis_set_id
 ARRAY1 1 2300 1 increasing ELEMENT_X
 ARRAY1 2 2300 2 increasing ELEMENT_Y

# category ARRAY_STRUCTURE_LIST_AXIS
loop_
_array_structure_list_axis.axis_set_id
_array_structure_list_axis.axis_id
_array_structure_list_axis.displacement
_array_structure_list_axis.displacement_increment
 ELEMENT_X ELEMENT_X 0.075 0.150
 ELEMENT_Y ELEMENT_Y 0.075 0.150

# category ARRAY_ELEMENT_SIZE
loop_
_array_element_size.array_id
_array_element_size.index
_array_element_size.size
 ARRAY1 1 150e-6
 ARRAY1 2 150e-6

# category ARRAY_INTENSITIES
loop_
_array_intensities.array_id
_array_intensities.binary_id
_array_intensities.linearity
_array_intensities.gain
_array_intensities.gain_esd
_array_intensities.overload
_array_intensities.undefined_value
 ARRAY1 1 linear 1.15 0.2 240000 0

 # category ARRAY_STRUCTURE
 loop_
 _array_structure.id
 _array_structure.encoding_type
 _array_structure.compression_type
 _array_structure.byte_order
 ARRAY1 "signed 32-bit integer" packed little_endian

# category ARRAY_DATA
loop_
_array_data.array_id
_array_data.binary_id
_array_data.data
 ARRAY1 1
;
--CIF-BINARY-FORMAT-SECTION--
Content-Type: application/octet-stream;
    conversions="x-CBF_PACKED"
Content-Transfer-Encoding: BASE64
X-Binary-Size: 3801324
X-Binary-ID: 1
X-Binary-Element-Type: "signed 32-bit integer"
Content-MD5: 07lZFvF+aOcW85IN7usl8A==

AABRAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA ...

...YDMD6J18Qg83OMr/tgssjMIJMXATDsZobL90AEXc4KigE

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

Example 3 – Example 2 revised for a spiral scan (R. M. Sweet, P. J. Ellis & H. J. Bernstein).

A detector is placed 240 mm along the Z axis from the goniometer, as in Example 2 above, but in this example the image plate is scanned in a spiral pattern from the outside edge in.

The axis for positioning the detector in the Y direction depends on the detector Z axis. This axis is called DETECTOR_Y.

The axis for positioning the detector in the X direction depends on the detector Y axis (and therefore on the detector Z axis). This axis is called DETECTOR_X.

This detector may be rotated around the Y axis. This rotation axis depends on the three translation axes. It is called DETECTOR_PITCH.

A coordinate system is defined on the face of the detector in terms of a coupled rotation axis and radial scan axis to form a spiral scan. The rotation axis is called ELEMENT_ROT and the radial axis is called ELEMENT_RAD. A 150 micrometre radial pitch and a 75 micrometre 'constant velocity' angular pitch are assumed.

Indexing is carried out first on the rotation axis and the radial axis is made to be dependent on it.

The two axes are coupled to form an axis set ELEMENT_SPIRAL.

```
###CBF: VERSION 1.1

 data_image_1

 # category DIFFRN
 _diffrn.id P6MB
 _diffrn.crystal_id P6MB_CRYSTAL7

 # category DIFFRN_SOURCE
 loop_
 _diffrn_source.diffrn_id
 _diffrn_source.source
 _diffrn_source.type
  P6MB synchrotron 'SSRL beamline 9-1'

 # category DIFFRN_RADIATION
      loop_
 _diffrn_radiation.diffrn_id
 _diffrn_radiation.wavelength_id
 _diffrn_radiation.monochromator
 _diffrn_radiation.polarizn_source_ratio
 _diffrn_radiation.polarizn_source_norm
 _diffrn_radiation.div_x_source
 _diffrn_radiation.div_y_source
 _diffrn_radiation.div_x_y_source
  P6MB WAVELENGTH1 'Si 111' 0.8 0.0 0.08
 0.01 0.00

 # category DIFFRN_RADIATION_WAVELENGTH
 loop_
 _diffrn_radiation_wavelength.id
 _diffrn_radiation_wavelength.wavelength
 _diffrn_radiation_wavelength.wt
  WAVELENGTH1 0.98 1.0
```

```
# category DIFFRN_DETECTOR
loop_
_diffrn_detector.diffrn_id
_diffrn_detector.id
_diffrn_detector.type
_diffrn_detector.number_of_axes
 P6MB MAR345-SN26 'MAR 345' 4

# category DIFFRN_DETECTOR_AXIS
loop_
_diffrn_detector_axis.detector_id
_diffrn_detector_axis.axis_id
 MAR345-SN26 DETECTOR_X
 MAR345-SN26 DETECTOR_Y
 MAR345-SN26 DETECTOR_Z
 MAR345-SN26 DETECTOR_PITCH

# category DIFFRN_DETECTOR_ELEMENT
loop_
_diffrn_detector_element.id
_diffrn_detector_element.detector_id
 ELEMENT1 MAR345-SN26

# category DIFFRN_DATA_FRAME
loop_
_diffrn_data_frame.id
_diffrn_data_frame.detector_element_id
_diffrn_data_frame.array_id
_diffrn_data_frame.binary_id
 FRAME1 ELEMENT1 ARRAY1 1

# category DIFFRN_MEASUREMENT
loop_
_diffrn_measurement.diffrn_id
_diffrn_measurement.id
_diffrn_measurement.number_of_axes
_diffrn_measurement.method
 P6MB GONIOMETER 3 rotation

# category DIFFRN_MEASUREMENT_AXIS
loop_
_diffrn_measurement_axis.measurement_id
_diffrn_measurement_axis.axis_id
 GONIOMETER GONIOMETER_PHI
 GONIOMETER GONIOMETER_KAPPA
 GONIOMETER GONIOMETER_OMEGA

# category DIFFRN_SCAN
loop_
_diffrn_scan.id
_diffrn_scan.frame_id_start
_diffrn_scan.frame_id_end
_diffrn_scan.frames
 SCAN1 FRAME1 FRAME1 1

# category DIFFRN_SCAN_AXIS
loop_
_diffrn_scan_axis.scan_id
_diffrn_scan_axis.axis_id
_diffrn_scan_axis.angle_start
_diffrn_scan_axis.angle_range
_diffrn_scan_axis.angle_increment
_diffrn_scan_axis.displacement_start
_diffrn_scan_axis.displacement_range
_diffrn_scan_axis.displacement_increment
 SCAN1 GONIOMETER_OMEGA 12.0 1.0 1.0 0.0 0.0 0.0
 SCAN1 GONIOMETER_KAPPA 23.3 0.0 0.0 0.0 0.0 0.0
 SCAN1 GONIOMETER_PHI -165.8 0.0 0.0 0.0 0.0 0.0
 SCAN1 DETECTOR_Z 0.0 0.0 0.0 -240.0 0.0 0.0
 SCAN1 DETECTOR_Y 0.0 0.0 0.0 0.6 0.0 0.0
 SCAN1 DETECTOR_X 0.0 0.0 0.0 -0.5 0.0 0.0
 SCAN1 DETECTOR_PITCH 0.0 0.0 0.0 0.0 0.0 0.0

# category DIFFRN_SCAN_FRAME
loop_
_diffrn_scan_frame.frame_id
_diffrn_scan_frame.frame_number
_diffrn_scan_frame.integration_time
_diffrn_scan_frame.scan_id
_diffrn_scan_frame.date
 FRAME1 1 20.0 SCAN1 1997-12-04T10:23:48
```

```
# category DIFFRN_SCAN_FRAME_AXIS
loop_
_diffrn_scan_frame_axis.frame_id
_diffrn_scan_frame_axis.axis_id
_diffrn_scan_frame_axis.angle
_diffrn_scan_frame_axis.displacement
 FRAME1 GONIOMETER_OMEGA 12.0 0.0
 FRAME1 GONIOMETER_KAPPA 23.3 0.0
 FRAME1 GONIOMETER_PHI -165.8 0.0
 FRAME1 DETECTOR_Z 0.0 -240.0
 FRAME1 DETECTOR_Y 0.0 0.6
 FRAME1 DETECTOR_X 0.0 -0.5
 FRAME1 DETECTOR_PITCH 0.0 0.0

# category AXIS
loop_
_axis.id
_axis.type
_axis.equipment
_axis.depends_on
_axis.vector[1] _axis.vector[2] _axis.vector[3]
_axis.offset[1] _axis.offset[2] _axis.offset[3]
 GONIOMETER_OMEGA rotation goniometer . 1 0 0 . . .
 GONIOMETER_KAPPA rotation goniometer GONIOMETER_OMEGA
0.64279
 0 0.76604 . . .
 GONIOMETER_PHI   rotation goniometer GONIOMETER_KAPPA 1 0 0
 . . .
 SOURCE           general source . 0 0 1 . . .
 GRAVITY          general gravity . 0 -1 0 . . .
 DETECTOR_Z       translation detector . 0 0 1 0 0 0
 DETECTOR_Y       translation detector DETECTOR_Z 0 1 0 0 0
0
 DETECTOR_X       translation detector DETECTOR_Y 1 0 0 0 0
0
 DETECTOR_PITCH   rotation    detector DETECTOR_X 0 1 0 0 0
0
 ELEMENT_ROT      translation detector DETECTOR_PITCH 0 0 1
0 0 0
 ELEMENT_RAD      translation detector ELEMENT_ROT 0 1 0 0 0
0

# category ARRAY_STRUCTURE_LIST
loop_
_array_structure_list.array_id
_array_structure_list.index
_array_structure_list.dimension
_array_structure_list.precedence
_array_structure_list.direction
_array_structure_list.axis_set_id
 ARRAY1 1 8309900 1 increasing ELEMENT_SPIRAL

# category ARRAY_STRUCTURE_LIST_AXIS
loop_
_array_structure_list_axis.axis_set_id
_array_structure_list_axis.axis_id
_array_structure_list_axis.angle
_array_structure_list_axis.displacement
_array_structure_list_axis.angular_pitch
_array_structure_list_axis.radial_pitch
 ELEMENT_SPIRAL ELEMENT_ROT 0    .  0.075   .
 ELEMENT_SPIRAL ELEMENT_RAD . 172.5  .    -0.150

# category ARRAY_ELEMENT_SIZE
# the actual pixels are 0.075 by 0.150 mm
# We give the coarser dimension here.
loop_
_array_element_size.array_id
_array_element_size.index
_array_element_size.size
 ARRAY1 1 150e-6

# category ARRAY_INTENSITIES
loop_
_array_intensities.array_id
_array_intensities.binary_id
_array_intensities.linearity
_array_intensities.gain
_array_intensities.gain_esd
_array_intensities.overload
_array_intensities.undefined_value
 ARRAY1 1 linear 1.15 0.2 240000 0
```

```
 # category ARRAY_STRUCTURE
 loop_
 _array_structure.id
 _array_structure.encoding_type
 _array_structure.compression_type
 _array_structure.byte_order
 ARRAY1 "signed 32-bit integer" packed little_endian

# category ARRAY_DATA
loop_
_array_data.array_id
_array_data.binary_id
_array_data.data
 ARRAY1 1
;
--CIF-BINARY-FORMAT-SECTION--
Content-Type: application/octet-stream;
    conversions="x-CBF_PACKED"
Content-Transfer-Encoding: BASE64
X-Binary-Size: 3801324
X-Binary-ID: 1
X-Binary-Element-Type: "signed 32-bit integer"
Content-MD5: 07lZFvF+aOcW85IN7usl8A==

AABRAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA ...

...YDMD6J18Qg83OMr/tgssjMIJMXATDsZobL90AEXc4KigE

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

_diffrn_scan.date_end (*yyyy-mm-dd*)

The date and time of the end of the scan. Note that this may be an estimate generated during the scan, before the precise time of the end of the scan is known.

[diffrn_scan]

_diffrn_scan.date_start (*yyyy-mm-dd*)

The date and time of the start of the scan.

[diffrn_scan]

***_diffrn_scan.frame_id_end** (*code*)

The value of this data item is the identifier of the last frame in the scan. This item is a pointer to _diffrn_data_frame.id in the DIFFRN_DATA_FRAME category.

[diffrn_scan]

***_diffrn_scan.frame_id_start** (*code*)

The value of this data item is the identifier of the first frame in the scan. This item is a pointer to _diffrn_data_frame.id in the DIFFRN_DATA_FRAME category.

[diffrn_scan]

_diffrn_scan.frames (*int*)

The value of this data item is the number of frames in the scan.

The permitted range is $[1, \infty)$. [diffrn_scan]

***_diffrn_scan.id** (*code*)

The value of _diffrn_scan.id uniquely identifies each scan. The identifier is used to tie together all the information about the scan.

The following item(s) have an equivalent role in their respective categories:

_diffrn_scan_axis.scan_id,

_diffrn_scan_frame.scan_id. [diffrn_scan]

_diffrn_scan.integration_time (*float*)

Approximate average time in seconds to integrate each step of the scan. The precise time for integration of each particular step must be provided in _diffrn_scan_frame.integration_time, even if all steps have the same integration time.

The permitted range is $[0.0, \infty)$. [diffrn_scan]

DIFFRN_SCAN_AXIS

Data items in the DIFFRN_SCAN_AXIS category describe the settings of axes for particular scans. Unspecified axes are assumed to be at their zero points.

Category group(s): inclusive_group
diffrn_group

Category key(s): _diffrn_scan_axis.scan_id
_diffrn_scan_axis.axis_id

_diffrn_scan_axis.angle_increment (*float*)

The increment for each step for the specified axis in degrees. In general, this will agree with _diffrn_scan_frame_axis.angle_inc The sum of the values of _diffrn_scan_frame_axis.angle and _diffrn_scan_frame_axis.angle_increment is the angular setting of the axis at the end of the integration time for a given frame. If the individual frame values vary, then the value of _diffrn_scan_axis.angle_increment will be representative of the ensemble of values of _diffrn_scan_frame_axis.angle_increment (*e.g.* the mean).

Where no value is given, the assumed value is '0.0'. [diffrn_scan_axis]

_diffrn_scan_axis.angle_range (*float*)

The range from the starting position for the specified axis in degrees.

Where no value is given, the assumed value is '0.0'. [diffrn_scan_axis]

_diffrn_scan_axis.angle_rstrt_incr (*float*)

The increment after each step for the specified axis in degrees. In general, this will agree with _diffrn_scan_frame_axis.angle_rstrt_incr. The sum of the values of _diffrn_scan_frame_axis.angle, _diffrn_scan_frame_axis.angle_increment and _diffrn_scan_ is the angular setting of the axis at the start of the integration time for the next frame relative to a given frame and should equal _diffrn_scan_frame_axis.angle for this next frame. If the individual frame values vary, then the value of _diffrn_scan_axis.angle_rstrt_incr will be representative of the ensemble of values of _diffrn_scan_frame_axis.angle_rstrt_incr (*e.g.* the mean).

Where no value is given, the assumed value is '0.0'. [diffrn_scan_axis]

_diffrn_scan_axis.angle_start (*float*)

The starting position for the specified axis in degrees.

Where no value is given, the assumed value is '0.0'. [diffrn_scan_axis]

***_diffrn_scan_axis.axis_id** (*code*)

The value of this data item is the identifier of one of the axes for the scan for which settings are being specified. Multiple axes may be specified for the same value of _diffrn_scan.id. This item is a pointer to _axis.id in the AXIS category.

[diffrn_scan_axis]

_diffrn_scan_axis.displacement_increment (*float*)

The increment for each step for the specified axis in millimetres. In general, this will agree with _diffrn_scan_frame_axis.displacem The sum of the values of _diffrn_scan_frame_axis.displacement and _diffrn_scan_frame_axis.displacement_increment is the angular setting of the axis at the end of the integration time for a given frame. If the individual frame values vary, then the value of _diffrn_scan_axis.displacement_increment will be representative of the ensemble of values of _diffrn_scan_frame_axis.displacement_increment (*e.g.* the mean).

Where no value is given, the assumed value is '0.0'. [diffrn_scan_axis]

_diffrn_scan_axis.displacement_range (*float*)

The range from the starting position for the specified axis in millimetres.

Where no value is given, the assumed value is '0.0'. [diffrn_scan_axis]

_diffrn_scan_axis.displacement_rstrt_incr (*float*)

The increment for each step for the specified axis in millimetres. In general, this will agree with _diffrn_scan_frame_axis.displacem The sum of the values of _diffrn_scan_frame_axis.displacement _diffrn_scan_frame_axis.displacement_increment and _diffrn_scan_frame_axis.displacement_rstrt_incr is the angular setting of the axis at the start of the integration time for the next frame relative to a given frame and should equal _diffrn_scan_frame_axis.displacement for this next frame. If the individual frame values vary, then the value of _diffrn_scan_axis.displacement_rstrt_incr will be representative of the ensemble of values of _diffrn_scan_frame_axis.displacement_rstrt_incr (*e.g.* the mean).

Where no value is given, the assumed value is '0.0'. [diffrn_scan_axis]

_diffrn_scan_axis.displacement_start (*float*)

The starting position for the specified axis in millimetres.

Where no value is given, the assumed value is '0.0'. **[diffrn_scan_axis]**

***_diffrn_scan_axis.scan_id** (*code*)

The value of this data item is the identifier of the scan for which axis settings are being specified. Multiple axes may be specified for the same value of **_diffrn_scan.id**. This item is a pointer to **_diffrn_scan.id** in the DIFFRN_SCAN category.

[diffrn_scan_axis]

DIFFRN_SCAN_FRAME

Data items in the DIFFRN_SCAN_FRAME category describe the relationships of particular frames to scans.

Category group(s): **inclusive_group**
diffrn_group

Category key(s): **_diffrn_scan_frame.scan_id**
_diffrn_scan_frame.frame_id

_diffrn_scan_frame.date (*yyyy-mm-dd*)

The date and time of the start of the frame being scanned.

[diffrn_scan_frame]

***_diffrn_scan_frame.frame_id** (*code*)

The value of this data item is the identifier of the frame being examined. This item is a pointer to **_diffrn_data_frame.id** in the DIFFRN_DATA_FRAME category.

[diffrn_scan_frame]

_diffrn_scan_frame.frame_number (*int*)

The value of this data item is the number of the frame within the scan, starting with 1. It is not necessarily the same as the value of **_diffrn_scan_frame.frame_id**, but it may be.

The permitted range is $[0, \infty)$. **[diffrn_scan_frame]**

***_diffrn_scan_frame.integration_time** (*float*)

The time in seconds to integrate this step of the scan. This should be the precise time of integration of each particular frame. The value of this data item should be given explicitly for each frame and not inferred from the value of **_diffrn_scan.integration_time**.

The permitted range is $[0.0, \infty)$. **[diffrn_scan_frame]**

***_diffrn_scan_frame.scan_id** (*code*)

The value of **_diffrn_scan_frame.scan_id** identifies the scan containing this frame. This item is a pointer to **_diffrn_scan.id** in the DIFFRN_SCAN category.

[diffrn_scan_frame]

DIFFRN_SCAN_FRAME_AXIS

Data items in the DIFFRN_SCAN_FRAME_AXIS category describe the settings of axes for particular frames. Unspecified axes are assumed to be at their zero points. If, for any given frame, nonzero values apply for any of the data items in this category, those values should be given explicitly in this category and not simply inferred from values in DIFFRN_SCAN_AXIS.

Category group(s): **inclusive_group**
diffrn_group

Category key(s): **_diffrn_scan_frame_axis.frame_id**
_diffrn_scan_frame_axis.axis_id

_diffrn_scan_frame_axis.angle (*float*)

The setting of the specified axis in degrees for this frame. This is the setting at the start of the integration time.

Where no value is given, the assumed value is '0.0'. **[diffrn_scan_frame_axis]**

_diffrn_scan_frame_axis.angle_increment (*float*)

The increment for this frame for the angular setting of the specified axis in degrees. The sum of the values of **_diffrn_scan_frame_axis.angle** and **_diffrn_scan_frame_axis.angle_increment** is the angular setting of the axis at the end of the integration time for this frame.

Where no value is given, the assumed value is '0.0'. **[diffrn_scan_frame_axis]**

_diffrn_scan_frame_axis.angle_rstrt_incr (*float*)

The increment after this frame for the angular setting of the specified axis in degrees. The sum of the values of **_diffrn_scan_frame_axis.angle**, **_diffrn_scan_frame_axis.angle_increment** and **_diffrn_scan_** is the angular setting of the axis at the start of the integration time for the next frame and should equal **_diffrn_scan_frame_axis.angle** for this next frame.

Where no value is given, the assumed value is '0.0'. **[diffrn_scan_frame_axis]**

***_diffrn_scan_frame_axis.axis_id** (*code*)

The value of this data item is the identifier of one of the axes for the frame for which settings are being specified. Multiple axes may be specified for the same value of **_diffrn_scan_frame.frame_id**. This item is a pointer to **_axis.id** in the AXIS category.

[diffrn_scan_frame_axis]

_diffrn_scan_frame_axis.displacement (*float*)

The setting of the specified axis in millimetres for this frame. This is the setting at the start of the integration time.

Where no value is given, the assumed value is '0.0'. **[diffrn_scan_frame_axis]**

_diffrn_scan_frame_axis.displacement_increment (*float*)

The increment for this frame for the displacement setting of the specified axis in millimetres. The sum of the values of **_diffrn_scan_frame_axis.displacement** and **_diffrn_scan_frame_axis.displacement_increment** is the angular setting of the axis at the end of the integration time for this frame.

Where no value is given, the assumed value is '0.0'. **[diffrn_scan_frame_axis]**

_diffrn_scan_frame_axis.displacement_rstrt_incr (*float*)

The increment for this frame for the displacement setting of the specified axis in millimetres. The sum of the values of **_diffrn_scan_frame_axis.displacement**, **_diffrn_scan_frame_axis.displacement_increment** and **_diffrn_scan_frame_axis.displacement_rstrt_incr** is the angular setting of the axis at the start of the integration time for the next frame and should equal **_diffrn_scan_frame_axis.displacement** for this next frame.

Where no value is given, the assumed value is '0.0'. **[diffrn_scan_frame_axis]**

***_diffrn_scan_frame_axis.frame_id** (*code*)

The value of this data item is the identifier of the frame for which axis settings are being specified. Multiple axes may be specified for the same value of **_diffrn_scan_frame.frame_id**. This item is a pointer to **_diffrn_data_frame.id** in the DIFFRN_DATA_FRAME category.

[diffrn_scan_frame_axis]