Draft NeXus-CBF Concordance Summary

Herbert J. Bernstein, Tobias S. Richter, Jonathan Sloan

8 August 2013

Contents

1. Introduction

THIS IS DRAFT FOR DISCUSSION.

NeXus and imgCIF/CBF are frameworks for containers of scientific data that may contain images or other binary data. NeXus is tree-oriented and imgCIF/CBF is table-oriented. In this document we summarize an approach to a concordance between them. Both frameworks allow for multiple alternative representations of the same data. HDF5 provides a particularly appropriate format for the management of large numbers of experimental data sets and for the representation of particularly large data sets.

This is a summary of the current state of the proposal. The proposal has been fairly stable now for several weeks. The only significant change from the drafts in May 2013 is to use the prefix "CBF_axis" for all axis names. The changes from the 6 June 2013 draft consists of minor corrections to details of the CBF to NeXus mapping for consistency and addition to a suggestion for handling polychromatic radiation. This effort combines input from Herbert J. Bernstein, Tobias S. Richter and Jonathan Sloan, as well as comments and suggestions by Mark Koennecke in 2010 (see `http://lists.nexusformat.org/pipermail/nexus-developers/attachments/20100201/a9424156/attachment-0001.el`) and by the members of NIAC.

This draft is by Herbert J. Bernstein, who takes complete responsibility for any mistakes and misunderstandings in this draft. Please send comments, corrections and suggestions to `yayahjb@gmail.com`.

The reader is assumed to already have familiarity with imgCIF/CBF, NeXus and HDF5. A copy of the imgCIF dictionary can be found at

`https://www.sites.google.com/site/nexuscbf/home/cbf-dictionary`

2. How to read this document

While the mappings described here are complex and detailed, the approaches to the mappings are simple. If you are familiar with both CBF and NeXus, you should read section 4 "Mapping from CBF to NeXus" on page 4 through "Identifying images" on page 9 and section 5 "Mapping from NeXus to CBF/imgCIF" on page 25 through "Mapping Fields" on page 26. If you are interested on the handling of the Dectris Eiger detector, see section 7 "Proposed Pixel Array Detector Application Definitions" on page 63

3. General Mapping Issues

A CBF is organized as a set of relational tables, each table is called a "category". The name of a category is essentially the name of a class. Categories are organized into "data blocks" the name of a category must be distinct from the name of all other categories in the same data block. As a relational table, each table has a key. One component of the key is, in general, an "ID".

A NeXus file is organized as a tree. Each node of the tree has a name, which must be distinct from all the other names of nodes that are children of the same parent, each node of the tree is an HDF5 NeXus class instance or an HDF5 dataset or an HDF5 attribute. Each NeXus class instance has a NeXus class.

In most cases the value of the ID in each row of a CBF table will have to be mapped to the name of a NeXus class instance in the NeXus file, and in most cases the name of a NeXus class instance in a NeXus file will have to be mapped to the value of an ID in some CBF table.

To avoid namespace conflicts, in going from CBF to NeXus, we carry the CBF category name prefixed by "CBF_" along with the ID value in "dotted" notation, using a double underscore, "__", in place of the period to conform to NeXus naming conventions, but so as to not convict with the use of the single underscore, "_", to separate components of names on a single hierarchical level. In going from NeXus to CBF, we strip leading dotted notation name components that match the target category name.

Similar considerations will arise throughout this mapping. The major exception is for the fields in NXdetector, in which the well-established practice in NeXus is to give those fields fixed names, such as "data". If there is no possibility of a name conflict, we will retain the standard name and uses suffixes, rather than prefixes in cases that require disambiguation. In the longer term, it would be best to ensure reliable identification of the function of fields with attributes, rather than by name. In each case, an attribute will be added for that purpose.

Most CBF categories accept identifiers for variants. NeXus does not have an equivalent concept at this time. This issue should be discussed.

CBF carries detailed information about the storage of images, such as the compression used, that is not exposed in NeXus. That information is carried in HDF5.

CBF is designed to organize scans, identified by a scan ID, SCANID, in the context of multiple scans per diffraction experiment, where each diffraction experiment is identified by a diffraction ID, DIFFRNID, in the context of multiple experiments per study of a macromolcule, where each macromolecular study is identified by an entry ID, ENTRYID. This creates a 3-level hierarchy of information supporting the eventual report and structural deposition. At present, NeXus, does not support an equivalent of this hierarchy. This issue should be discussed.

4. Mapping from CBF to NeXus

For the following mapping, where a placement current defined NeXus class instances under NXentry has not yet been specified the mapping is shown as "→ ??", but all CBF items have a mapping under an CBF_cbf NeXus class instance that we propose to place under NXentry, so further mapping could be done entirely in NeXus or HDF5, if desired. Each _CATEGORY.COLUMN value of type CBFTYPE in row NN in data block DATABLOCK is mapped to

```
/entry__ENTRYID:NXentry
```

```
/CBF_cbf:CBF_cbf
    /DATABLOCK:CBF_cbfdb
        /CATEGORY:CBF_cbfcat
            /COLUMN:CBF_cbfcol
                /NN
                    /value
```

with the attribute @cbftype=CBFTYPE, so no information will be lost, and all the information in the rest of the NeXus tree will be available in these relational tables. For most traditional NeXus uses of the file, these tables may be ignored or deleted, but they are needed for efficient external management of multiple files in the context of a larger data management system. When very large numbers of datasets have to be handled at a facility, standard practice is to put information on which searches will be done into a relational database. CBF tables are such information. Translating them faithfully into NeXus allows that information to preserved with the NeXus/HDF5 files, rather than having to deal with two different file formats for this information. In order to index information from a NeXus file into a database, as is done, for example in the iCAT project, first the information needs to be in the NeXus file. Until the CBF to NeXus mapping has become complete and automatic, the extra CBF_cbf class ensures that all the information is available for such database use.

4.1. The NeXus Structure Into Which to Map

The following is the target tree structure. All elements of this mapping should be carefully considered and discussed. The attribute and field names with CBF category name prefixes are potential placeholder for shorter more readable names to be discussed. Names with double asterisks are flagged as particularly worth discussing.

```
/CBF_diffrn_scan__SCANID:NXentry
  /CBF_scan_id="SCANID"
  /CBF_diffrn_id="DIFFRNID"
  /CBF_entry_id="ENTRYID"
  /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /start_time=STARTDATETIME
      /end_time=ENDDATETIME
      /CBF_diffrn_scan_frame__date=DATES
      /CBF_diffrn_scan_frame__frame_id=IDS
      /average_count_time=AVGCOUNTTIME
        /@units="sec"
      /average_frame_restart_time=RSTRTTIME
        /@units="sec"
      /average_frame_time=TIMEPER
        /@units="sec"
      /count_time=COUNTTIMES
        /@units="sec"
      /frame_time=TIMEPERS
        /@units="sec"
      /frame_restart_time=RSTRTTIME
        /@units="sec"
      /frame_start_number=FRAMESTARTNO
      /frame_end_number=FRAMEENDNO
      /distance --> /NXentry/NXinstrument/NXsample/CBF_diffrn_measurement__sample_detector_distance
      /data_ARRAYID_BINARYID=DATA
        /@CBF_array_id="ARRAYID"
        /@CBF_binary_id="BINARYID"
        /@CBF_header_contents="HEADER"
        /@CBF_header_convention="HEADERCONVENTION"
        /@x_pixel_size=XPSIZE
        /@y_pixel_size=YPSIZE
        /@CBF_array_intensities__details="DETAILS"
        /@CBF_array_intensities__gain=GAIN
        /@CBF_array_intensities__gain_esd=GAINESD
        /@CBF_array_intensities__linearity="LINEARITY"
        /@CBF_array_intensities__offset=OFFSET
        /@CBF_array_intensities__scaling=SCALING
        /@CBF_array_intensities__overload=OVERLOAD
        /@CBF_array_intensities__undefined_value=UNDEFVAL
        /@CBF_array_intensities__pixel_fast_bin_size=FBINSIZE
        /@CBF_array_intensities__pixel_slow_bin_size=SBINSIZE
        /@CBF_array_intensities__pixel_binning_method="METHOD"
      /deadtime=DTIME
      /description=DESCRIPTION
      /details=DETAILS
      /number_of_axes=NUMDETAXES
      /type=TYPE

      /CBF_array_structure_list__AXISSET1=[]
        /@CBF_array_id="ARRAYID"
        /@CBF_array_structure_list__dimension=DIM1
```

```
    /@CBF_array_structure_list__direction="DIR1"
    /@CBF_array_structure_list__index=1
    /@CBF_axis=PRECEDENCE1
  /CBF_array_structure_list__AXISSET2=[]
    /@CBF_array_id="ARRAYID"
    /@CBF_array_structure_list__dimension=DIM2
    /@CBF_array_structure_list__direction="DIR2"
    /@CBF_array_structure_list__index=2
    /@CBF_axis=PRECEDENCE2
  /CBF_array_structure_list_section__SECTIONID=[]
    /@CBF_array_id="ARRAYID"
    /@CBF_array_structure_list_section__index=INDEX
    /@CBF_array_structure_list_section__end=END
    /@CBF_array_structure_list_section__start=START
    /@CBF_array_structure_list_section__stride=STRIDE
  /CBF_array_structure_list_axis__AXISID=[]          **
    /@CBF_array_structure_list_axis__axis_id="AXISID"          **
    /@CBF_array_structure_list_axis__axis_set_id="AXISSETID"   **
    /@CBF_array_structure_list_axis__angle=ANGLE               **
    /@CBF_array_structure_list_axis__angle_increment=ANGLEINC  **
    /@CBF_array_structure_list_axis__displacement=DISP         **
    /@CBF_array_structure_list_axis__displacement=FRACTDISP    **
    /@CBF_array_structure_list_axis__fract_displacement=DISPINC **
    /@CBF_array_structure_list_axis__fract_displacement_increment=FRACTINC **
    /@CBF_array_structure_list_axis__angular_pitch=ANGPITCH    **
    /@CBF_array_structure_list_axis__radial_pitch=RADPITCH     **
    /@CBF_array_structure_list_axis__reference_angle=REFANG    **
    /@CBF_array_structure_list_axis__reference_displacement=REFDISP **
  /CBF_diffrn_scan_axis__AXISID=[]
     /@CBF_axis_id="AXISID"                                    **
     /@CBF_diffrn_scan_axis__angle_start=ANGSTART
     /@CBF_diffrn_scan_axis__angle_range=ANGRANGE
     /@CBF_diffrn_scan_axis__angle_increment=ANGINC
     /@CBF_diffrn_scan_axis__angle_rstrt_incr=ANGRSTRT
     /@CBF_diffrn_scan_axis__displacement_start=DISPSTART
     /@CBF_diffrn_scan_axis__displacement_range=DISPRANGE
     /@CBF_diffrn_scan_axis__displacement_increment=DISPINC
     /@CBF_diffrn_scan_axis__displacement_rstrt_incr=DISPRSTRT
     /@CBF_diffrn_scan_axis__reference_angle=ANG
     /@CBF_diffrn_scan_axis__reference_displacement=DISP
  /CBF_diffrn_detector_element__id="ELEMENTID1:ELEMENTID2:..."      **
  /CBF_diffrn_detector_element__reference_center_fast=[RCF1,RCF2,...] **
  /CBF_diffrn_detector_element__reference_center_slow=[RCS1,RCS2,...] **
  /CBF_diffrn_detector_element__id="UNITS1:UNITS2:..." **
  /CBF_diffrn_data_frame__section_id=SECTIONIDARRAY **
  /CBF_diffrn_data_frame__binary_id=BINARYIDARRAY  **
  /CBF_diffrn_data_frame__center_fast_slow=CENTERARRAY **
    /@units="UNITS"
  /CBF_diffrn_data_frame__details=DETAILSARRAY

/CBF_diffrn_measurement__GONIOMETER:NXsample
  /CBF_diffrn_measurement__details="DETAILS"
  /CBF_diffrn_measurement__device="DEVICE"
  /CBF_diffrn_measurement__device_details="DEVDETAILS"
  /CBF_diffrn_measurement__device_type="DEVTYPE"
  /CBF_diffrn_measurement__method="METHOD"
  /number_of_axes=NUMBER
  /CBF_diffrn_measurement__sample_detector_distance=DIST
    /@units="mm"
  /CBF_diffrn_measurement__sample_detector_voffset=VOFST
    /@units="mm"
  /CBF_diffrn_measurement__specimen_support="SPECSPRT"
  /CBF_diffrn_radiation__collimation="COLLIMATION"
  /divergence_x=DIVX
    /@units="deg"
  /divergence_y=DIVY
    /@units="deg"
  /CBF_diffrn_radiation__div_x_y_source=DIVXY
    /@units="deg^2"
  /CBF_diffrn_radiation__filter_edge=ABSEDGE
    /@units="angstroms"
  /CBF_diffrn_radiation__inhomogeneity=HWIDTH
    /@units="mm"

/monochromator:NXmonochromator
  /wavelength=WAVELENGTH
  /CBF_diffrn_radiation__monochromator="MONOCHROMATOR"
```

```
   /CBF_diffrn_radiation__polarisn_norm=POLNANG
     /@units="deg"
   /CBF_diffrn_radiation__polarisn_ratio=POLRAT
   /CBF_diffrn_radiation__polarizn_source_norm=POLSNANG
     /@units="deg"
   /CBF_diffrn_radiation__polarizn_source_ratio=POLSRAT
   /CBF_diffrn_radiation__probe="RADIATION"
   /CBF_diffrn_radiation__type="SIEGBAHNTYPE"
   /CBF_diffrn_radiation__xray_symbol="IUPACXRAYSYMB"
 /CBF_diffrn_scan_SCANID:NXscan
 /CBF_diffrn_scan_frame_monitor__DETECTORNAME:NXmonitor
   /data=MONITORVALUES
   /count_time=INTEGRATIONTIMES
     /@units="sec"
```

This is a fragmentary example of the axis definitions as proposed for NeXus taken from the CBF fragment afterwards.

```
/instrument:NXinstrument
   /CBF_diffrn_detector__DETECTOR:NXdetector
      CBF_axis__DETECTOR_PITCH=[0.]
         @units="deg"
         @CBF_location="image_1.axis.vector.10"
         @depends_on="axis__DETECTOR_Y"
         @transformation_type="rotation"
         @vector=[-1, 0, 0]
      CBF_axis__DETECTOR_Y=[0.]
         @units="mm"
         @CBF_location="image_1.axis.vector.9"
         @depends_on="axis__DETECTOR_Z"
         @transformation_type="translation"
         @vector=[0, -1, 0]
      CBF_axis__DETECTOR_Z=[250.]
         @units="mm"
         @CBF_location="image_1.axis.vector.8"
         depends_on="."
         @transformation_type="translation"
         @vector= [0, 0, 1]
      CBF_axis__ELEMENT_X=[0.]
         @units="mm"
         @offset_units="mm"
         @CBF_location="image_1.axis.vector.11"
         @depends_on="axis__DETECTOR_PITCH"
         @transformation_type="translation"
         @vector= [-1, 0, 0]
         @offset= [-211.818, -217.322, 0]
      CBF_axis__ELEMENT_Y=[0.]
         @units="mm"
         @CBF_location=image_1.axis.vector.12
         @depends_on="axis__ELEMENT_X"
         @transformation_type="translation"
         @vector= [0, 1, 0]
   /CBF_coordinate_system:NXcoordinate_system
      /CBF_axis__SLS_X=[]
         @CBF__location="image_1.axis.vector.0"
         @depends_on="."
         @vector= [-1, 0, 0]
      /CBF_axis__SLS_Y=[]
         @CBF__location="image_1.axis.vector.1"
         @depends_on="."
         @vector= [0, -1, 0]
      /CBF_axis__SLS_Z=[]
         @CBF__location="image_1.axis.vector.2"
         @depends_on="."
         @vector= [0, 0, 1]
      /CBF_axis__GRAVITY=[]
         @cbf_location="image_1.axis.vector.7"
         @depends_on="."
         @vector=[0, -1, 0]
      /CBF_axis__BEAM=[]
         @cbf_location="image_1.axis.vector.6"
         @epends_on="."
         @vector= [0, 0, 1]
   /CBF_diffrn_measurement__GONIOMETER:NXsample   (Note:  Changed fron NXgoniometer, 8Aug13)
      /CBF_axis__GONIOMETER_KAPPA=[0]
         @units="deg"
         @CBF__location="image_1.axis.vector.4"
         @depends_on="axis__GONIOMETER_OMEGA"
```

```
                @transformation_type="rotation"
                @vector= [-0.64279, 0.76604, 0]
             /CBF_axis__GONIOMETER_OMEGA=[0]
                @units="deg"
                @CBF__location=image_1.axis.vector.3
                @depends_on="."
                @transformation_type="rotation"
                @vector= [1, 0, 0]
             /CBF_axis__GONIOMETER_PHI=[0]
                @units="deg"
                @CBF__location=image_1.axis.vector.5
                @depends_on="axis__GONIOMETER_KAPPA"
                @transformation_type="rotation"
                @vector= [1, 0, 0]

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]   #___|___________|_____________|_____________|__|__|___|__\__
                  #   |           |             |             |  |  |   |  |  \\
                  #   |           |             |             |  |  |   |  |  |
######################|###########|#############|###############|##|##|###|##|##|
# The SLS Beamline axis convention is similar to the imgCIF convention, but    |
# Y and Z run the other way       |             |               |  |  |   |  |  |
######################|###########|#############|###############|##|##|###|##|##|
#                     |           |             |               |  |  |   |  |  |
  SLS_X             general     general         .               1  0  0 0  0  0
  SLS_Y             general     general         .               0 -1  0 0  0  0
  SLS_Z             general     general         .               0  0 -1 0  0  0
#                     |           |             |               |  |  |   |  |  |
######################|###########|#############|###############|##|##|###|##|##|
# We define a kappa geometry with a left-handed omega and phi and a right- |  |
# handed kappa.  The kappa axis arm is at the top when omega is zero    |  |  |
######################|###########|#############|###############|##|##|###|##|##|
#                     |           |             |               |  |  |   |  |  |
  GONIOMETER_OMEGA  rotation    goniometer      .               -1  0  0  .  .  .
  GONIOMETER_KAPPA  rotation    goniometer      GONIOMETER_OMEGA
                                                          0.64279 0.76604
                                                                  0  .  .  .
  GONIOMETER_PHI    rotation    goniometer      GONIOMETER_KAPPA
                                                                 -1  0  0  .  .  .
######################|###########|#############|###############|##|##|###|##|##|
  BEAM              general     source          .               0  0 -1
.  .  .
  GRAVITY           general     gravity         .               0 -1  0
.  .  .
######################|###########|#############|###############|##|##|###|##|##|
#                     |           |             |               |  |  |   |  |  |
######################|###########|#############|###############|##|##|###|##|##|
# The detector is assumed to be mounted on an arm parallel to the beam     |  |
# with a DETECTOR_Y vertical translation and a pitch axis                  |  |
######################|###########|#############|###############|##|##|###|##|##|
#                     |           |             |               |  |  |   |  |  |
  DETECTOR_Z        translation detector        .               0  0 -1  0  0  0
  DETECTOR_Y        translation detector        DETECTOR_Z      0 -1  0  0  0  0
  DETECTOR_PITCH    rotation    detector        DETECTOR_Y      1  0  0  0  0  0
######################|###########|#############|###############|##|##|###|##|##|
# This detector is assumed to have the 0,0 corner at +X and -Y   |  |   |  |  |
# we assume a 2463 x 2527 detector on a 0.172 mm pixel pitch     |  |   |  |  |
######################|###########|#############|###############|##|##|###|##|##|
#                     |           |             |               |  |  |   |  |  |
  ELEMENT_X         translation detector        DETECTOR_PITCH 1  0  0
                                                                211.818 -217.322 0
  ELEMENT_Y         translation detector        ELEMENT_X       0  1  0   0  0  0
```

4.2. The NeXus top level

The top level presented on the NeXus side of this mapping is

```
/CBF_diffrn_scan__SCANID:NXentry
```

These scans can then be NeXus class instanceed into diffraction experiments and then into macromolecular studies by careful organization of files in tree of directories, but as the number of files and directories become large and individual scans get moved,

there is a serious risk of loss of critical information if the identifiers for higher levels of the hierarchy are not recoverable from individual scans. For this reason, the fields

```
/CBF_scan_id="SCANID"
/CBF_diffrn_id="DIFFRNID"
/CBF_entry_id="ENTRYID"
```

are given at the top level, if available, even if the same information is carried deeper in the hierarchy.

4.3. Identifying images

In many experimental disciplines, including X-ray crystallography, multiple related images are generated a single experiment. In some cases these can be gathered into a single array, but there are also many cases in which multiple data arrays are needed. Therefore, it is not sufficient to reserve the name "data" for the data. In imgCIF and array of data is identified by two identifiers, an array ID, specifying the structure of the array, and a binary ID identifying a specific array. We name such data arrays by composing the commonly used name "data" these two identifiers separated by an underscores "_".

When only one array ID and only one binary ID are present, then it will be sufficient to just use the field name "data".

It is important to note that an image from an single detector may consist of only a slice (or "section") taken from a larger array of data, and that a multi-element detector may be stored either as a single slice from one array common to all the elements, or as separate slices from different arrays, one array (or even more than one array) per detector element.

4.4. The ARRAY_DATA category

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

It is recognized that the data in this category needs to be used in
two distinct ways.  During a data collection the lack of ancillary
data and timing constraints in processing data may dictate the
need to make a 'miniCBF' nothing more than an essential minimum
of information to record the results of the data collection.  In that
case it is proper to use the ARRAY_DATA category as a
container for just a single image and a compacted, beam-line
dependent list of data collection parameter values.  In such
a case, only the tags '_array_data.header_convention',
'_array_data.header_contents' and '_array_data.data' need be
populated.

For full processing and archiving, most of the tags in this
dictionary will need to be populated.
```

- _array_data.array_id ARRAYID →
  /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /data_ARRAYID_BINARYID
  /@CBF_array_id="ARRAYID"
- _array_data.binary_id BINARYID →
  /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /data_ARRAYID_BINARYID
  /@CBF_binary_id="BINARYID"
- _array_data.data DATAARRAY→
  /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /data_ARRAYID_BINARYID=DATAARRAY
- _array_data.header_contents HEADER→
  /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /data__ARRAYID_BINARYID=DATAARRAY
  /@CBF_header_contents="HEADER"
- _array_data.header_convention HEADERCONVENTION →
  /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /data_ARRAYID_BINARYID=DATAARRAY
  /@CBF_header_convention="HEADERCONVENTION"

4.5. The ARRAY_ELEMENT_SIZE category

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

- _array_element_size.array_id ARRAYID→
  /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /?_pixel_size_ARRAYID
  /@CBF_array_id="ARRAYID"
  where "?" is "x", "y", "z" for _array_element_size.index == 1,2, or 3 respectively
- _array_element_size.index See _array_element_size.array_id

- _array_element_size.size SIZE→
  /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /?_pixel_size_ARRAYID=SIZE
        /@units="m"
  where "?" is "x", "y", "z" for _array_element_size.index == 1,2, or 3 respectively
- _array_element_size.variant → ?? NeXus does not handle variants at this time

4.6. The ARRAY_INTENSITIES category

```
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.
```

- _array_intensities.array_id → ARRAYID→
  /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /data_ARRAYID_BINARYID
        /@CBF_array_id="ARRAYID"

- _array_intensities.binary_id →
  /instrument:NXinstrument /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /data_ARRAYID_BINARYID
        /@CBF_binary_id="BINARYID"

- _array_intensities.details DETAILS → /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /data_ARRAYID_BINARYID=DATAARRAY
        /@CBF_array_intensities__details="DETAILS"

- _array_intensities.gain GAIN → /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /data_ARRAYID_BINARYID=DATAARRAY
        /@CBF_array_intensities__gain=GAIN

- _array_intensities.gain_esd GAINESD → /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /data_ARRAYID_BINARYID=DATAARRAY
        /@CBF_array_intensities__gain_esd=GAINESD

- _array_intensities.linearity LINEARITY →
  /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /data_ARRAYID_BINARYID=DATAARRAY
        /@CBF_array_intensities__linearity="LINEARITY"

- _array_intensities.offset OFFSET → /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /data_ARRAYID_BINARYID=DATAARRAY
        /@CBF_array_intensities__offset=OFFSET

- _array_intensities.scaling SCALING →
  /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /data_ARRAYID_BINARYID=DATAARRAY
        /@CBF_array_intensities__scaling=SCALING

- _array_intensities.overload OVERLOAD →
  /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /data_ARRAYID_BINARYID=DATAARRAY
        /@CBF_array_intensities__overload=OVERLOAD

- _array_intensities.undefined_value UNDEFVAL →
  /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /data_ARRAYID_BINARYID=DATAARRAY
        /@CBF_array_intensities__undefined_value=UNDEFVAL

- _array_intensities.pixel_fast_bin_size FBINSIZE →
  /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /data_ARRAYID_BINARYID=DATAARRAY
        /@CBF_array_intensities__pixel_fast_bin_size=FBINSIZE

- _array_intensities.pixel_slow_bin_size SBINSIZE →
  /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /data_ARRAYID_BINARYID=DATAARRAY
        /@CBF_array_intensities__pixel_slow_bin_size=SBINSIZE

- _array_intensities.pixel_binning_method METHOD →
  /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /data_ARRAYID_BINARYID=DATAARRAY
        /@CBF_array_intensities__pixel_binning_method=”METHOD”

- _array_intensities.variant → ?? NeXus does not handle variants at this time

The argument has been made that these attributes are not needed because NeXus files are supposed to have ”true values” stored. In many cases that is true and then none of these attributes are needed. However, with some detectors and some experiments there are good technical and scientific reasons to bring in values that will need processing later to derive ”true values”, and in those case some or all of these attributes will be needed. They are provided for such cases.

4.7. The ARRAY_STRUCTURE category

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

Note that this is essentially a type that may apply to multiple binary images, and corresponds to some of the detailed HDF5 information about an array. The following mapping is a placeholder for the names given for future reference, if needed.

The information in this category is the byte order, the compression information, and the encoding, which is carried in and reteievable from the HDF5 types, properties lists, etc.

At present NeXus does not expose this information. This should be discussed.

4.7.1. The ARRAY_STRUCTURE_LIST category Data items in the ARRAY_STRUCTURE_LIST category record the size and organization of each array dimension.

- _array_structure_list.axis_set_id AXISSET →
  /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /CBF_array_structure_list__AXISSET=[]
- _array_structure_list.array_id ARRAYID →
  /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /CBF_array_structure_list__AXISSET=[]
        /@CBF_array_id=”ARRAYID”
- _array_structure_list.dimension DIM→
  /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /CBF_array_structure_list__AXISSET=[]
        /@CBF_array_structure_list__dimension=DIM
- _array_structure_list.direction DIM→
  /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /CBF_array_structure_list__AXISSET=[]
        /@CBF_array_structure_list__direction=DIR
- _array_structure_list.index INDEX→ /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /CBF_array_structure_list__AXISSET=[]
        /@CBF_array_structure_list__index=INDEX
- _array_structure_list.precedence PRECEDENCE→ /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /CBF_array_structure_list__AXISSET=[]
        /@CBF_array_structure_list__precedence=PRECEDENCE
- _array_structure_list.variant → ?? NeXus does not handle variants at this time

This mapping of precedence of axes and directions may seem redundant with the array storage parameters maintained by HDF5, but changed information is used by applications to remap images to undo rotations and flips and needs to be accessible without rewriting the image, which may be impractical. Failure to expose at least this information in the NeXus will greatly restrict the range of applications that will be able to use the NeXus version of these files.

4.7.2. The ARRAY_STRUCTURE_LIST_SECTION category Data items in the ARRAY_STRUCTURE_LIST_SECTION category identify the dimension-by-dimension start, end and stride of each section of an array that is to be referenced.

For any array of array_id, ARRAYID, array section ids of the form ARRAYID(start1:end1:stride1,start2:end2:stride2, ...) are defined by default.

For the given index, the elements in the section are of indices:
_array_structure_list_section.start, _array_structure_list_section.start + _array_structure_list_section.stride, _array_structure_list_section.start +2*_array_structure_list_section.stride, ...

stopping either when the indices leave the limits of the indices of that dimension or [min(_array_structure_list_section.start, _array_structure_list_section.end), max(_array_structure_list_section.start, _array_structure_list_section.end)].

The ordering of these elements is determined by the overall ordering of _array_structure_list_section.array_id and not by the ordering implied by the stride.

- _array_structure_list_section.array_id ARRAYID → /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_array_structure_list_section__SECTIONID=[]
  /@CBF_array_id=”ARRAYID”
- _array_structure_list_section.id SECTIONID → /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_array_structure_list_section__SECTIONID=[]
- _array_structure_list_section.index INDEX→ /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_array_structure_list_section__SECTIONID=[]
  /@CBF_array_structure_list_section__index=INDEX
- _array_structure_list_section.end END → /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_array_structure_list_section__SECTIONID=[]
  /@CBF_array_structure_list_section__end=END
- _array_structure_list_section.start START → /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_array_structure_list_section__SECTIONID=[]
  /@CBF_array_structure_list_section__start=START
- _array_structure_list_section.stride STRIDE → /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_array_structure_list_section__SECTIONID=[]
  /@CBF_array_structure_list_section__stride=STRIDE
- _array_structure_list_section.variant → ?? NeXus does not handle variants at this time

4.7.3. The ARRAY_STRUCTURE_LIST_AXIS category 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.

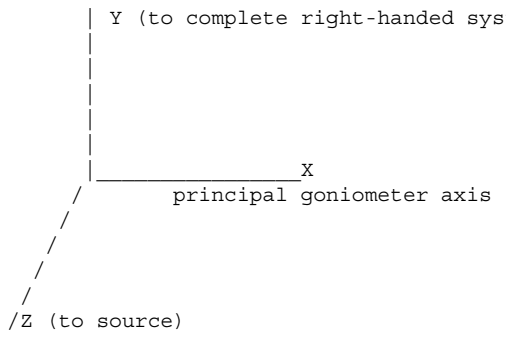
- _array_structure_list_axis.axis_id AXISID →
  /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_axis__AXISID=[]
  /@CBF_array_structure_list_axis__axis_id=”AXISID”
- _array_structure_list_axis.axis_set_id AXISSETID → /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_axis__AXISID=[]
  /@CBF_array_structure_list_axis__axis_set_id=”AXISSETID”
- _array_structure_list_axis.angle ANGLE →
  /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_axis__AXISID=[]
  /@CBF_array_structure_list_axis__angle=ANGLE
- _array_structure_list_axis.angle_increment ANGLEINC →
  /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_axis__AXISID=[]
  /@CBF_array_structure_list_axis__angle_increment=ANGLEINC
- _array_structure_list_axis.displacement DISP →
  /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_axis__AXISID=[]
  /@CBF_array_structure_list_axis__displacement=DISP
- _array_structure_list_axis.fract_displacement FRACTDISP →
  /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_axis__AXISID=[]
  /@CBF_array_structure_list_axis__displacement=FRACTDISP
- _array_structure_list_axis.displacement_increment DISPINC →
  /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_axis__AXISID=[]
  /@CBF_array_structure_list_axis__displacement_increment=DISPINC
- _array_structure_list_axis.fract_displacement_increment FRACTINC →
  /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_axis__AXISID=[]
  /@CBF_array_structure_list_axis__fract_displacement_increment=FRACTINC
- _array_structure_list_axis.angular_pitch ANGPITCH →
  /instrument:NXinstrument

/CBF_diffrn_detector__DETECTORNAME:NXdetector
/CBF_axis__AXISID=[]
/@CBF_array_structure_list_axis__angular_pitch=ANGPITCH

- _array_structure_list_axis.radial_pitch RADPITCH →
/instrument:NXinstrument
/CBF_diffrn_detector__DETECTORNAME:NXdetector
/CBF_axis__AXISID=[]
/@CBF_array_structure_list_axis__radial_pitch=RADPITCH
- _array_structure_list_axis.reference_angle REFANG →
/instrument:NXinstrument
/CBF_diffrn_detector__DETECTORNAME:NXdetector
/CBF_axis__AXISID=[]
/@CBF_array_structure_list_axis__reference_angle=REFANG
- _array_structure_list_axis.reference_displacement REFDISP →
/instrument:NXinstrument
/CBF_diffrn_detector__DETECTORNAME:NXdetector
/CBF_axis__AXISID=[]
/@CBF_array_structure_list_axis__reference_displacement=REFDISP
- _array_structure_list_axis.variant → ?? NeXus does not handle variants at this time

4.8. AXIS category

```
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 or the axes defining the
coordinate system of an image.

The location of each axis is specified by two vectors: the axis
itself, given by a  unit vector in the direction of the axis, and
an offset to the base of the unit vector.

The vectors defining an axis are referenced to an appropriate
coordinate system.  The axis vector, itself, is a dimensionless
unit vector.  Where meaningful, the offset vector is given in
millimetres.  In coordinate systems not measured in metres,
the offset is not specified and is taken as zero.

The available coordinate systems are:

    The imgCIF standard laboratory coordinate system
    The direct lattice (fractional atomic coordinates)
    The orthogonal Cartesian coordinate system (real space)
    The reciprocal lattice
    An abstract orthogonal Cartesian coordinate frame

For consistency in this discussion, we call the three coordinate
system axes X, Y and Z.  This is appropriate for the imgCIF
standard laboratory coordinate system, and last two Cartesian
coordinate systems, but for the direct lattice, X corresponds
to a, Y to b and Z to c, while for the reciprocal lattice,
X corresponds to a*, Y to b* and Z to c*.

For purposes of visualization, all the coordinate systems are
taken as right-handed, i.e., using the convention that the extended
thumb of a right hand could point along the first (X) axis, the
straightened pointer finger could point along the second (Y) axis
and the middle finger folded inward could point along the third (Z)
axis.

THE IMGCIF STANDARD LABORATORY COORDINATE SYSTEM

The imgCIF standard laboratory coordinate system is a right-handed
orthogonal coordinate similar to the MOSFLM coordinate system,
but imgCIF puts Z along the X-ray beam, rather than putting X along the
X-ray beam as in MOSFLM.

The vectors for the imgCIF standard laboratory coordinate system
form a right-handed Cartesian coordinate system with its origin
in the sample or specimen.  The origin of the axis system should,
if possible, be defined in terms of mechanically stable axes to be
be both in the sample and in the beam.  If the sample goniometer or other
sample positioner has two axes the intersection of which defines a
unique point at which the sample should be mounted to be bathed
by the beam, that will be the origin of the axis system.  If no such
point is defined, then the midpoint of the line of intersection
between the sample and the center of the beam will define the origin.
For this definition the sample positioning system will be set at
its initial reference position for the experiment.
```



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 or sample positioning system if the sample positioning system has an axis that intersects the origin and which form an angle of more than 22.5 degrees with the beam axis.

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.

If the conditions for the X-axis can be met, the coordinate system will be based on the goniometer or other sample positioning system and the beam and 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.

In some experimental techniques, there is no goniometer or the principal axis of the goniometer is at a small acute angle with respect to the source axis. In such cases, other reference axes are needed to define a useful coordinate system. The order of priority in defining directions in such cases is to use the detector, then gravity, then north.

If the X-axis cannot be defined as above, then the direction (not the origin) of the X-axis should be parallel to the axis of the primary detector element corresponding to the most rapidly varying dimension of that detector element's data array, with its positive sense corresponding to increasing values of the index for that dimension. If the detector is such that such a direction cannot be defined (as with a point detector) or that direction forms an angle of less than 22.5 degrees with respect to the source axis, then the X-axis should be chosen so that if the Y-axis is chosen in the direction of gravity, and the Z-axis is chosen to be along the source axis, a right-handed orthogonal coordinate system is chosen. In the case of a vertical source axis, as a last resort, the X-axis should be chosen to point North.

All rotations are given in degrees and all translations are given in mm.

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.

The unit cell, reciprocal cell and crystallographic orthogonal
Cartesian coordinate system are defined by the CELL and the matrices
in the ATOM_SITES category.

THE DIRECT LATTICE (FRACTIONAL COORDINATES)

The direct lattice coordinate system is a system of fractional
coordinates aligned to the crystal, rather than to the laboratory.
This is a natural coordinate system for maps and atomic coordinates.
It is the simplest coordinate system in which to apply symmetry.
The axes are determined by the cell edges, and are not necessarily
othogonal. This coordinate system is not uniquely defined and
depends on the cell parameters in the CELL category and the
settings chosen to index the crystal.

Molecules in a crystal studied by X-ray diffracraction are organized
into a repeating regular array of unit cells. Each unit cell is defined
by three vectors, a, b and c. To quote from Drenth,

"The choice of the unit cell is not unique and therefore, guidelines
have been established for selecting the standard basis vectors and
the origin. They are based on symmetry and metric considerations:

 "(1) The axial system should be right handed.
  (2) The basis vectors should coincide as much as possible with
  directions of highest symmetry."
  (3) The cell taken should be the smallest one that satisfies
  condition (2)
  (4) Of all the lattice vectors, none is shorter than a.
  (5) Of those not directed along a, none is shorter than b.
  (6) Of those not lying in the ab plane, none is shorter than c.
  (7) The three angles between the basis vectors a, b and c are
  either all acute (<90\%) or all obtuse (90\%)."

These rules do not produce a unique result that is stable under
the assumption of experimental errors, and the the resulting cell
may not be primitive.

In this coordinate system, the vector (.5, .5, .5) is in the middle
of the given unit cell.

Grid coordinates are an important variation on fractional coordinates
used when working with maps. In imgCIF, the conversion from
fractional to grid coordinates is implicit in the array indexing
specified by _array_structure_list.dimension. Note that this
implicit grid-coordinate scheme is 1-based, not zero-based, i.e.
the origin of the cell for axes along the cell edges with no
specified _array_structure_list_axis.displacement will have
grid coordinates of (1,1,1), i.e. array indices of (1,1,1).

THE ORTHOGONAL CARTESIAN COORDINATE SYSTEM (REAL SPACE)

The orthogonal Cartesian coordinate system is a transformation of
the direct lattice to the actual physical coordinates of atoms in
space. It is similar to the laboratory coordinate system, but
is anchored to and moves with the crystal, rather than being
schored to the laboratory. The transformation from fractional
to orthogonal cartesian coordinates is given by the
         _atom_sites.Cartn_transf_matrix[i][j] and
         _atom_sites.Cartn_transf_vector[i]
tags. A common choice for the matrix of the transformation is
given in the 1992 PDB format document

         | a   b cos(\g)   c cos(\b) |

```
          | 0      b sin(\g)   c (cos(\a) - cos(\b)cos(\g))/sin(\g) |
          | 0      0           V/(a b sin(\g))                      |

   This is a convenient coordinate system in which to do fitting
   of models to maps and in which to understand the chemistry of
   a molecule.

   THE RECIPROCAL LATTICE

   The reciprocal lattice coordinate system is used for diffraction
   intensitities.  It is based on the reciprocal cell, the dual of the cell,
   in which reciprocal cell edges are derived from direct cell faces:

      a* = bc sin(\a)/V  b* = ac sin(\b)/V  c* = ab sin(\g)/V
      cos(\a*) = (cos(\b) cos(\g) - cos(\a))/(sin(\b) sin(\g))
      cos(\b*) = (cos(\a) cos(\g) - cos(\b))/(sin(\a) sin(\g))
      cos(\g*) = (cos(\a) cos(\b) - cos(\g))/(sin(\a) sin(\b))
      V = abc SQRT(1 - cos(\a)^2^
                     - cos(\b)^2^
                     - cos(\g)^2^
                     + 2 cos(\a) cos(\b) cos(\g) )

   In this form the dimensions of the reciprocal lattice are in reciprocal
   \%Angstroms (\%A^-1^).  A dimensionless form can be obtained by
   multiplying by the wavelength.  Reflections are commonly indexed against
   this coordinate system as (h, k, l) triples.


   References:

   Drenth, J., "Introduction to basic crystallography." chapter
   2.1 in Rossmann, M. G. and Arnold, E. "Crystallography of
   biological macromolecules", Volume F of the IUCr's "International
   tables for crystallography", Kluwer, Dordrecht 2001, pp 44 -- 63

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

   Stout, G. H. and Jensen, L. H., "X-ray structure determination",
   2nd ed., Wiley, New York, 1989, 453 pp.

   __, "PROTEIN DATA BANK ATOMIC COORDINATE AND BIBLIOGRAPHIC ENTRY
   FORMAT DESCRIPTION," Brookhaven National Laboratory, February 1992.
```

- _axis.depends_on DEPNAME→
  /CBF_axis__AXISID=[]
  @depends_on="DEPNAME"—
  provided the DEPNAME is in the same NeXus class instance as AXISID. Otherwise a full path needs to be provided.
- _axis.equipment EQUIP →
  see _axis.id AXISID, below
- _axis.id AXISID →
  /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_axis__AXISID=[]
  for EQUIP=="detector"
  /instrument:NXinstrument
  /CBF_diffrn_measurement__GONIOMETERNAME:NXsample
  /CBF_axis__AXISID=[]
  for EQUIP=="goniometer"
  /instrument:NXinstrument
  /coordinate_system:NXcoordinate_system
  /CBF_axis__AXISID=[]
  for EQUIP=="general"
  note that @units="mm" or @units="deg" should also be specified if the settings array is populated
- _axis.offset[1] O1→
  /CBF_axis__AXISID=[]
  @offset=offsetxform([O1,O2,O3])
  @offset_units="mm"
- _axis.offset[2] O2→
  /CBF_axis__AXISID=[]
  @offset=offsetxform([O1,O2,O3])
  @offset_units="mm"
- _axis.offset[3] O3→
  /CBF_axis__AXISID=[]
  @offset=offsetxform([O1,O2,O3])
  @offset_units="mm"

- _axis.type TYPE→
  /CBF_axis__AXISID=[]
  @transformation_type="TYPE"
- _axis.system →
  Only a laboratory coordinate system is handled in NeXus. See the discussion of transformation to the McStas coordinate system below.
- _axis.vector[1] V1 →
  /CBF_axis__AXISID=[]
  @vector=coordxform([V1,V2,V3])
- _axis.vector[2] V2 →
  /CBF_axis__AXISID=[]
  @vector=coordxform([V1,V2,V3])
- _axis.vector[3] V3 →
  /CBF_axis__AXISID=[]
  @vector=coordxform([V1,V2,V3])
- _axis.variant → ?? NeXus does not handle variants at this time

4.8.1. Differences in Coordinate Frames The standard coordinate frame in NeXus is the McStas coordinate frame, in which the Z-axis points in the direction of the incident beam, the X-axis is orthogonal to the Z-axis in the horizontal plane and pointing left as seen from the source and the Y-axis points upwards. The origin is in the sample.

The standard coordinate frame in imgCIF/CBF aligns the X-axis to the principal goniometer axis, chooses the Z-axis to point from the sample into the beam. If the beam is not orthogonal to the X-axis, the Z-axis is the component of the vector points into the beam orthogonal to the X-axis. The Y-axis is chosen to complete a right-handed axis system.

Let us call the NeXus coordinate axes, X_{nx}, Y_{nx} and Z_{nx} and the imgCIF/CBF coordinate axes, X_{cbf}, Y_{cbf} and Z_{cbf} and the direction of gravity, $Gravity$. In order to translate a vector $v_{nx} = (x, y, z)$ from the NeXus coordinate system to the imgCIF coordinate system, we also need two additional axes, as unit vectors, $Gravity_{cbf}$, the downwards direction, and $Beam_{cbf}$, the direction of the beam (*e.g.* $(0, 0, -1)$).

In practice, the beam is not necessarily perfectly horizontal, so Y_{nx} is not necessarily perfectly vertical. Therefore, in order to generate X_{nx}, Y_{nx} and Z_{nx} some care is needed. The cross product between two vectors $\vec{a}$ and $\vec{b}$ is a new vector $\vec{c}$ orthogonal to both $\vec{a}$ and $\vec{b}$, chosen so that $\vec{a}$, $\vec{b}$, $\vec{c}$ is a right handed system. If $\vec{a}$ and $\vec{b}$ are orthogonal unit vectors, this right-handed system is an orthonormal coordinate system.

In the CBF coordinate frame, Z_{nx} is aligned to $Beam_{cbf}$:

$$Z_{nx} = Beam_{cbf}$$

X_{nx} is defined as being horizontal at right angles to the beam, pointing to the left when seen from the source. Assuming the beam is not vertical, we can compute X_{nx} as the normalized cross product of the beam and the gravity:

$$X_{nx} = (Beam_{cbf} \times Gravity_{cbf})/||Beam_{cbf} \times Gravity_{cbf}||$$

To see that this satisfies the constraint of being horizontal and pointing to the left, consider the case of $Beam = (0, 0, -1)$ and $Gravity = (0, 0, 1)$ then we would have $X_{nx} = (1, 0, 0)$ from the cross product above. The normalization is only necessary if the beam is not horizontal.

Finally Y_{nx} is computed as the cross product of the beam and X_{nx}, completing a orthonormal right-handed system with Y_{nx} pointing upwards:

$$Y_{nx} = Beam_{cbf} \times X_{nx}$$

Then we know that in the imgCIF/CBF coordinate frame

$$v_{nx} = x \cdot X_{nx} + y \cdot Y_{nx} + z \cdot Z_{nx}$$

Thus, given the imgCIF/CBF vectors for the true direction of the beam and the true direction of gravity, we have a linear transformation from the NeXus coordinate frame to the imgCIF/CBF coordinate frame. The origins of the two frames agree. The inverse linear transformation will transform a vector in the imgCIF/CBF coordinate frame into the NeXus coordinate frame.

In the common case in which the beam is orthogonal to the principal goniometer axis so that $Beam_{cbf} = (0, 0, -1)$ and the imgCIF/CBF Y-axis points upwards, the transformation inverts the X and Z axes. In the other common case in which the beam is orthogonal to the principal goniometer axis and the imgCIF/CBF Y-axis points downwards, the transformation inverts the Y and Z axes.

4.8.2. Mapping Axes There are two transformations needed: $coordxform(v)$ which takes a vector, v, the the CBF imgCIF Standard Laboratory Coordinate System and returns the equivalent McStas coordinate vector, and $offsetxform(o)$ which takes an offset, o, in the the CBF imgCIF Standard Laboratory Coordinate System and returns the equivalent NeXus offset. As of this writing, it has not been decided as to whether the NeXus offset should also be relative (in which case $offsetxform = coordxform$) or whether the NeXus offset should be absolute.

In imgCIF/CBF all the information about all axes other than their settings are gathered in one AXIS category. The closest equivalent container in NeXus is the NXinstrument class, We put the information about detector axes into an detector:NXdetector NeXus class instance, information about the goniometer into an goniometer:NXsample NeXus class instance, etc. Additionally, in view of the general nature of some axes, such as the coordinate frame axes and gravity, we add a coordinate_system:NXcoordinate_system NeXus class instance with axis__gravity, axis__beam and other axes not tied to specific equipment.

We have applied the coordinate frame transformation changing the CBF laboratory coordinates into McStas coordinates. Notice that X and Z have changed direction, but Y has not. In other experimental setup, other transformations may accur. The offsets for dependent axes are given relative to the total offset of axes on which that axis is dependent. Note that the axis settings do not enter into this calculation, because the offsets of dependent axes are given with all axes at their zero settings

The cbf_location attribute gives a mapping back into the CBF AXIS category in dotted notation. The first component is the data block. The second component is "axis". The third component is either "vector" or "offset" for information drawn from the AXIS.VECTOR[...] or AXIS.OFFSET[...] respectively. The last component is the CBF row number to facilitate recovering the original CBF layout.

4.9. DIFFRN_DATA_FRAME category

```
Data items in the DIFFRN_DATA_FRAME category record
the details about each frame of data.
```

- _diffrn_data_frame.array_id ARRAYID →
  /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /CBF_diffrn_data_frame__section_id=SECTIONIDARRAY
  inserts either ARRAYID (if not SECTIONID is specified or the SECTIONID into the element of SECTIONIDARRY for this frame and for this detector element (see below)
- _diffrn_data_frame.array_section_id SECTIONID →
  /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /CBF_diffrn_data_frame__section_id=SECTIONIDARRAY
  inserts either ARRAYID (if not SECTIONID is specified or the SECTIONID into the element of SECTIONIDARRY for this frame and for this detector element (see below)
- _diffrn_data_frame.binary_id BINID → /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /CBF_diffrn_data_frame__binary_id=BINARYIDARRAY
  inserts BINID into the element of BINARYIDARRAY for this frame and for this detector element (see below)
- _diffrn_data_frame.center_fast CENF→ /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /CBF_diffrn_data_frame__center_fast_slow=CENTERARRAY
  inserts CENF into the element of CENTERARRAY for this frame, for this detector element and for the fast center (see below)
- _diffrn_data_frame.center_slow CENS →
  /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /CBF_diffrn_data_frame__center_fast_slow=CENTERARRAY
  inserts CENS into element of CENTERARRAY for this frame, for this detector element and for the slow center (see below)
- _diffrn_data_frame.center_units UNITS →
  /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /CBF_diffrn_data_frame__center_fast_slow=CENTERARRAY
        @units=”UNITS”
  only one unit is provided. If there is variation, the values in CENTERARRAY should be rescaled to uniform units.
- _diffrn_data_frame.detector_element_id ELEMENTID →
  used to index into the arrays of this category by the ordinal of the matching ELEMENTID in diffrn_detector_element__id for the fast index
- _diffrn_data_frame.id FRAMEID →
  used to index into the arrays of this category by the ordinal of the matching ELEMENTID in diffrn_detector_element__id for the slow index by matching FRAMEID against _diffrn_scan_frame.frame_id and using _diffrn_scan_frame.frame_number from the same row.
- _diffrn_data_frame.details DETAILS → /instrument:NXinstrument
    /CBF_diffrn_detector__DETECTORNAME:NXdetector
      /CBF_diffrn_data_frame__details=DETAILSARRAY
  inserts DETAILS into the element of DETAILSARRAY for this frame and for this detector element (see below)
- _diffrn_data_frame.variant → ??

The arrays created in the mapping have a slow index of the number of frames and a fast index of the number of detector elements. There is a middle index for CENTERARRAY in the order fast and then slow.

4.10. DIFFRN_DETECTOR category

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

- _diffrn_detector.diffrn_id DIFFRNID →
  /CBF_diffrn_scan__SCANID:NXentry
    /CBF_diffrn_id=”DIFFRNID”
- _diffrn_detector.id DETECTORNAME →
  /instrument:NXinstrument/CBF_diffrn_detector__DETECTORNAME:NXdetector
- _diffrn_detector.details →
  /instrument:NXinstrument/CBF_diffrn_detector__DETECTORNAME:NXdetector/details (*a newly proposed field*)
- _diffrn_detector.detector →
  /instrument:NXinstrument/CBF_diffrn_detector__DETECTORNAME:NXdetector/type
- _diffrn_detector.dtime →
  /instrument:NXinstrument/CBF_diffrn_detector__DETECTORNAME:NXdetector/deadtime
- _diffrn_detector.number_of_axes →
  /instrument:NXinstrument/CBF_diffrn_detector__DETECTORNAME:NXdetector/number_of_axes (*a newly proposed field*)
- _diffrn_detector.type →
  /instrument:NXinstrument/CBF_diffrn_detector__DETECTORNAME:NXdectector/description
- _diffrn_detector.variant → ?? (*NeXus does not handle variants at this time*)

4.10.1. DIFFRN_DETECTOR_AXIS category

```
Data items in the DIFFRN_DETECTOR_AXIS category associate
axes with detectors.
```

- _diffrn_detector_axis.axis_id AXISID → /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_axis__AXISID=[]
- _diffrn_detector_axis.detector_id DETECTORNAME → /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
- _diffrn_detector_axis.variant → ?? (*NeXus does not handle variants at this time*)

This information normally will duplicate information obtained from the ARRAY_STRUCTURE_LIST_AXIS.

4.10.2. DIFFRN_DETECTOR_ELEMENT category

```
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.
```

- _diffrn_detector_element.id ELEMENTID → /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_diffrn_detector_element__id=”ELEMENTID1:ELEMENTID2:...”
  inserts ELEMENTID into the colon-separated list of element IDs
- _diffrn_detector_element.detector_id DETECTORNAME → /instrument:NXinstrument /CBF_diffrn_detector__DETECTORNAME:NXdetector

- _diffrn_detector_element.reference_center_fast RCF →
  /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_diffrn_detector_element__reference_center_fast=[RCF1,RCF2,...]
  inserts RCF into the array of reference centers
- _diffrn_detector_element.reference_center_slow RCS →
  /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_diffrn_detector_element__reference_center_slow=[RCS1,RCS2,...]
  inserts RCS into the array of reference centers
- _diffrn_detector_element.reference_center_units UNITS → /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_diffrn_detector_element__id=”UNITS1:UNITS2:...”
  inserts ELEMENTID into the colon-separated list of units
- _diffrn_detector_element.variant → ?? (*NeXus does not handle variants at this time*)

4.11. DIFFRN_MEASUREMENT category

```
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.
```

- _diffrn_measurement.diffrn_id DIFFRNID →
  /CBF_diffrn_scan__SCANID:NXentry
  /CBF_diffrn_id=”DIFFRNID”
- _diffrn_measurement.details DETAILS→
  /instrument:NXinstrument
  /CBF_diffrn_measurement__GONIOMETER:NXsample
  /CBF_diffrn_measurement__details=”DETAILS”
- _diffrn_measurement.device DEVICE →
  /instrument:NXinstrument
  /CBF_diffrn_measurement__GONIOMETER:NXsample
  /CBF_diffrn_measurement__device=”DEVICE”
- _diffrn_measurement.device_details DEVDETAILS → /instrument:NXinstrument
  /CBF_diffrn_measurement__GONIOMETER:NXsample
  /CBF_diffrn_measurement__device_details=”DEVDETAILS”
- _diffrn_measurement.device_type DEVTYPE → /instrument:NXinstrument
  /CBF_diffrn_measurement__GONIOMETER:NXsample
  /CBF_diffrn_measurement__device_type=”DEVTYPE”
- _diffrn_measurement.id GONIOMETER → /instrument:NXinstrument
  /CBF_diffrn_measurement__GONIOMETER:NXsample
- _diffrn_measurement.method METHOD → /instrument:NXinstrument
  /CBF_diffrn_measurement__GONIOMETER:NXsample
  /CBF_diffrn_measurement__method=”METHOD”
- _diffrn_measurement.number_of_axes NUMBER → /instrument:NXinstrument
  /CBF_diffrn_measurement__GONIOMETER:NXsample
  /number_of_axes=NUMBER
- _diffrn_measurement.sample_detector_distance DIST→ /instrument:NXinstrument
  /CBF_diffrn_measurement__GONIOMETER:NXsample
  /distance=DIST
  /@units=”mm”
  and create a link from NXdetector/NXinstrument/NXDetector/distance to here

- _diffrn_measurement.sample_detector_voffset VOFST → /instrument:NXinstrument
  /CBF_diffrn_measurement__GONIOMETER:NXsample
  /CBF_diffrn_measurement__sample_detector_voffset=VOFST
  /@units="mm"
- _diffrn_measurement.specimen_support SPECSPRT → /instrument:NXinstrument
  /CBF_diffrn_measurement__GONIOMETER:NXsample
  /CBF_diffrn_measurement__specimen_support="SPECSPRT"
- _diffrn_measurement.variant → ??(*NeXus does not handle variants at this time*)

4.11.1. DIFFRN_MEASUREMENT_AXIS category

```
Data items in the DIFFRN_MEASUREMENT_AXIS category associate
axes with goniometers.
```

- _diffrn_measurement_axis.axis_id AXISID → /instrument:NXinstrument
  /CBF_diffrn_measurement__GONIOMETER:NXsample
  /CBF_axis__AXISID=[]
- _diffrn_measurement_axis.measurement_device DEVICE →
  /instrument:NXinstrument /CBF_diffrn_measurement__GONIOMETER:NXsample
  /CBF_diffrn_measurement__details="DEVICE"
- _diffrn_measurement_axis.measurement_id GONIOMETER →
  /instrument:NXinstrument
  /CBF_diffrn_measurement__GONIOMETER:NXsample
- _diffrn_measurement_axis.variant → ?? (*NeXus does not handle variants at this time*)

This information normally will duplicate information obtained from the ARRAY_STRUCTURE_LIST_AXIS.

4.12. The DIFFRN_RADIATION category

```
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.
```

- _diffrn_radiation.collimation COLLIMATION →
  /instrument:NXinstrument
  /collimator:NXcollimator
  /CBF_diffrn_radiation__collimation="COLLIMATION"
- _diffrn_radiation.diffrn_id DIFFRNID →
  /CBF_diffrn_scan__SCANID:NXentry
  /CBF_diffrn_id="DIFFRNID"
- _diffrn_radiation.div_x_source DIVX →
  /instrument:NXinstrument
  /collimator:NXcollimator
  /divergence_x=DIVX
  /@units="deg"
- _diffrn_radiation.div_y_source DIVY →
  /instrument:NXinstrument
  /collimator:NXcollimator
  /divergence_y=DIVY
  /@units="deg"
- _diffrn_radiation.div_x_y_source DIVXY →
  /instrument:NXinstrument
  /collimator:NXcollimator
  /CBF_diffrn_radiation__div_x_y_source=DIVXY
  /@units="deg^2"
- _diffrn_radiation.filter_edge ABSEDGE →
  /instrument:NXinstrument
  /collimator:NXcollimator
  /CBF_diffrn_radiation__filter_edge=ABSEDGE
  /@units="angstroms"
- _diffrn_radiation.inhomogeneity HWIDTH →
  /instrument:NXinstrument
  /collimator:NXcollimator
  /CBF_diffrn_radiation__inhomogeneity=HWIDTH
  /@units="mm"
- _diffrn_radiation.monochromator MONOCHROMATOR →
  /instrument:NXinstrument
  /collimator:NXmonochromator
  /CBF_diffrn_radiation__monochromator="MONOCHROMATOR"
- _diffrn_radiation.polarisn_norm POLNANG→
  /instrument:NXinstrument
  /collimator:NXmonochromator
  /CBF_diffrn_radiation__polarisn_norm=POLNANG
  /@units="deg"
- _diffrn_radiation.polarisn_ratio POLRAT →
  /instrument:NXinstrument
  /collimator:NXmonochromator
  /CBF_diffrn_radiation__polarisn_ratio=POLRAT

- _diffrn_radiation.polarizn_source_norm POLSNANG →
  /instrument:NXinstrument
    /collimator:NXmonochromator
      /CBF_diffrn_radiation__polarisn_source_norm=POLSNANG
        /@units="deg"
- _diffrn_radiation.polarizn_source_ratio POLSRAT →
  /instrument:NXinstrument
    /collimator:NXmonochromator
      /CBF_diffrn_radiation__polarizn_source_ratio=POLSRAT
- _diffrn_radiation.probe RADIATION →
  /instrument:NXinstrument
    /collimator:NXmonochromator
      /CBF_diffrn_radiation__probe="RADIATION"
- _diffrn_radiation.type SIEGBAHNTYPE →
  /instrument:NXinstrument
    /collimator:NXmonochromator
      /CBF_diffrn_radiation__type="SIEGBAHNTYPE"
- _diffrn_radiation.xray_symbol IUPACXRAYSYMB →
  /instrument:NXinstrument
    /collimator:NXmonochromator
      /CBF_diffrn_radiation__xray_symbol="IUPACXRAYSYMB"
- _diffrn_radiation.wavelength_id ID → /instrument:NXinstrument
    /collimator:NXmonochromator
      /wavelength=WAVELENGTH
        @units="angstroms"
  where WAVELENGTH is the value of _diffrn_radiation_wavelength.wavelength
  for which _diffrn_radiation_wavelength.id==ID, and for which _diffrn_radiation_wavelength.wt is maximized. This is the dominant wavelength. See the mapping of the DIFFRN_RADIATION_WAVELENGTH category, below, for more complex cases.
- _diffrn_radiation.variant → ?? (*NeXus does not handle variants at this time*)

4.13. The DIFFRN_RADIATION_WAVELENGTH category

```
Data items in the DIFFRN_RADIATION_WAVELENGTH category
describe the wavelength of the radiation used to measure the
diffraction intensities. Items may be looped to identify
and assign weights to distinct components of a
polychromatic beam.
```

- _diffrn_radiation_wavelength.id ID→
  /instrument:NXinstrument
    /CBF_diffrn_radiation_wavelength__id=["ID"]
- _diffrn_radiation_wavelength.wavelength_id WAVELENGTH_ID→
  /instrument:NXinstrument
    /CBF_diffrn_radiation_wavelength__wavelength_id=["WAVELENGTH_ID"]

- _diffrn_radiation_wavelength.wavelength WAVELENGTH →
  /instrument:NXinstrument
    /CBF_diffrn_radiation_wavelength__wavelength=[WAVELENGTH]

- _diffrn_radiation_wavelength.wt WEIGHT→ /instrument:NXinstrument
    /CBF_diffrn_radiation_wavelength__wt=[WEIGHT]

- _diffrn_radiation_wavelength.variant → ?? (*NeXus does not handle variants at this time*)

4.14. The DIFFRN_REFLN category

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

Data items in the DIFFRN_REFLN category record details about
the intensities in the diffraction data set
identified by _diffrn_refln.diffrn_id.

The DIFFRN_REFLN data items refer to individual intensity
measurements and must be included in looped lists.

The DIFFRN_REFLNS data items specify the parameters that apply
to all intensity  measurements in the particular diffraction
data set identified by _diffrn_reflns.diffrn_id and
_diffrn_refln.frame_id
```

- _diffrn_refln.frame_id → ??
- _diffrn_refln.variant → ??

This category will be addressed at a future date.

4.15. The DIFFRN_SCAN category

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

- _diffrn_scan.id →
  /CBF_diffrn_scan__SCANID:NXentry
  /CBF_scan_id="SCANID"

- _diffrn_scan.date_end ENDDATETIME→
  /CBF_diffrn_scan__SCANID:NXentry
  /CBF_scan_id="SCANID"
  /end_time=ENDDATETIME
- _diffrn_scan.date_start STARTDATETIME→
  /CBF_diffrn_scan__SCANID:NXentry
  /CBF_scan_id="SCANID"
  /start_time=STARTDATETIME
- _diffrn_scan.integration_time AVGCOUNTTIME →
  /CBF_diffrn_scan__SCANID:NXentry
  /CBF_scan_id="SCANID"
  /average_count_time=AVGCOUNTTIME
  /@units="sec"
- _diffrn_scan.frame_id_start FRAMESTARTID →
  /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /frame_start_number=FRAMESTARTNO
  where FRAMESTARTNO is the value of _diffrn_scan_frame.frame_number for which the value of _diffrn_scan_frame.frame_id equals FRAMESTARTID
- _diffrn_scan.frame_id_end FRAMEENDID
  /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /frame_end_number=FRAMEENDNO
  where FRAMEENDNO is the value of _diffrn_scan_frame.frame_number for which the value of _diffrn_scan_frame.frame_id equals FRAMEENDID
- _diffrn_scan.frames FRAMES →
  carried in NeXus as the slow dimension of arrays that are organized by frame.
- _diffrn_scan.time_period TIMEPER →
  /CBF_diffrn_scan__SCANID:NXentry
  /CBF_scan_id="SCANID"
  /frame_time=TIMEPER
  /@units="sec"
- _diffrn_scan.time_rstrt_incr RSTRTTIME →
  /CBF_diffrn_scan__SCANID:NXentry
  /CBF_scan_id="SCANID"
  /frame_restart_time=RSTRTTIME
  /@units="sec"
- _diffrn_scan.variant → ?? (*NeXus does not handle variants at this time*)

4.15.1. The DIFFRN_SCAN_AXIS category

```
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.
```

- _diffrn_scan_axis.axis_id AXISID→
  /CBF_diffrn_scan_axis__AXISID=[]
  /@CBF_axis_id=AXISID
  placed under the NXdetector or NXsample to which the axis belongs.
- _diffrn_scan_axis.angle_start ANGSTART →
  /CBF_diffrn_scan_axis__AXISID=[]
  /@diffrn_scan_axis__angle_start=ANGSTART
- _diffrn_scan_axis.angle_range ANGRANGE →
  /CBF_diffrn_scan_axis__AXISID=[]
  /@diffrn_scan_axis__angle_range=ANGRANGE
- _diffrn_scan_axis.angle_increment ANGINC →
  /CBF_diffrn_scan_axis__AXISID=[]
  /@diffrn_scan_axis__angle_increment=ANGINC
- _diffrn_scan_axis.angle_rstrt_incr ANGRSTRT →
  /CBF_diffrn_scan_axis__AXISID=[]
  /@diffrn_scan_axis__angle_rstrt_incr=ANGRSTRT
- _diffrn_scan_axis.displacement_start DISPSTART →
  /CBF_diffrn_scan_axis__AXISID=[]
  /@diffrn_scan_axis__displacement_start=DISPSTART
- _diffrn_scan_axis.displacement_range DISPRANGE →
  /CBF_diffrn_scan_axis__AXISID=[]
  /@diffrn_scan_axis__displacement_range=DISPRANGE
- _diffrn_scan_axis.displacement_increment DISPINC →
  /CBF_diffrn_scan_axis__AXISID=[]
  /@diffrn_scan_axis__displacement_increment=DISPINC
- _diffrn_scan_axis.displacement_rstrt_incr DISPRSTRT →
  /CBF_diffrn_scan_axis__AXISID=[]
  /@diffrn_scan_axis__displacement_rstrt_incr=DISPRSTRT
- _diffrn_scan_axis.reference_angle ANG →

/CBF_diffrn_scan_axis__AXISID=[]
/@diffrn_scan_axis__reference_angle=ANG

- _diffrn_scan_axis.reference_displacement DISP →
  /CBF_diffrn_scan_axis__AXISID=[]
  /@diffrn_scan_axis__reference_displacement=DISP
- _diffrn_scan_axis.scan_id SCANID→
  /CBF_diffrn_scan__SCANID:NXentry
  /CBF_scan_id="SCANID"
- _diffrn_scan_axis.variant → ?? (*NeXus does not handle variants at this time*)

4.15.2. The DIFFRN_SCAN_FRAME category

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

- _diffrn_scan_frame.date DATETIME→
  /CBF_diffrn_scan__SCANID:NXentry
  /CBF_scan_id="SCANID"
  /CBF_diffrn_scan_frame__date=DATES
  inserts DATETIME as the element at index _diffrn_scan_frame.frame_number in the array DATES
- _diffrn_scan_frame.frame_id ID →
  /CBF_diffrn_scan__SCANID:NXentry
  /CBF_scan_id="SCANID"
  /CBF_diffrn_scan_frame__frame_id=IDS
  inserts ID as the element at index _diffrn_scan_frame.frame_number in the array IDS
- _diffrn_scan_frame.frame_number →
  is the slow index for each of the arrays in this category??
- _diffrn_scan_frame.integration_time COUNTTIME →
  /CBF_diffrn_scan__SCANID:NXentry
  /CBF_scan_id="SCANID"
  /count_time=COUNTTIMES
  inserts COUNTTIME as the element at index _diffrn_scan_frame.frame_number in the array COUNTTIMES
- _diffrn_scan_frame.scan_id →
  /CBF_diffrn_scan__SCANID:NXentry
  /CBF_scan_id="SCANID"
- _diffrn_scan_frame.time_period FRAMETIME →
  /CBF_diffrn_scan__SCANID:NXentry
  /CBF_scan_id="SCANID"
  /frame_time=FRAMETIMES
  inserts FRAMETIME as the element at index _diffrn_scan_frame.frame_number in the array FRAMETIMES
- _diffrn_scan_frame.time_rstrt_incr RSTRTTIME →
  /CBF_diffrn_scan__SCANID:NXentry
  /CBF_scan_id="SCANID"
  /frame_restart_time=RSTRTTIMES
  inserts RSTRTTIME as the element at index _diffrn_scan_frame.frame_number in the array RSTRTTIMES
- _diffrn_scan_frame.variant → ?? (*NeXus does not handle variants at this time*)

4.15.3. The DIFFRN_SCAN_FRAME_AXIS category

```
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.
```

- _diffrn_scan_frame_axis.axis_id AXISID→ /instrument:NXinstrument
  /CBF_diffrn_detector__DETECTORNAME:NXdetector
  /CBF_diffrn_scan_axis__AXISID=[]
  for _axis.equipment=="detector" for _axis.id==AXISID
  /instrument:NXinstrument
  /CBF_diffrn_measurement__GONIOMETERNAME:NXsample
  /CBF_diffrn_scan_axis__AXISID=[]
  for _axis.equipment=="goniometer" for _axis.id==AXISID
  /instrument:NXinstrument
  /coordinate_system:NXcoordinate_system
  /CBF_diffrn_scan_axis__AXISID=[]
  for _axis.equipment=="general" for _axis.id==AXISID
  note that @units="mm" or @units="deg" should also be specified.
  The dimensions of the array depend on np (the number of frames = the value of _diffrn_scan.frames), and the presence of any of the ...._increment and ...._rstrt values. Using Fortran memory layout for this discussion, if the increments are present, the array is a 2-dimensional np by 3 array, with np as the fast axis and the [np,1] column being the settings, the [np,2] column being the increments, and the [np,3] column being the restart increments. If the ...reference_... values are present, the the array is extended with be np by 4 array with the last column handling the ...reference_... values If the frame-by-frame increments and reference values are not present the array is a 1-dimensional array of np settings.
- _diffrn_scan_frame_axis.angle ANGLE →
  inserts this as the *i*'th element, counting from 1, in /CBF_diffrn_scan_axis__AXISID if this is a rotation axis, where *i* is the value of _diffrn_scan_frame.frame_number for which the value of _diffrn_scan_frame.frame_id agrees with the value of _diffrn_scan_frame_axis.frame_id

- _diffrn_scan_frame_axis.angle_increment → see _diffrn_scan_frame_axis.axis_id
- _diffrn_scan_frame_axis.angle_rstrt_incr → see _diffrn_scan_frame_axis.axis_id
- _diffrn_scan_frame_axis.displacement DISP → inserts this as the *i*'th element, counting from 1, in /CBF_diffrn_scan_axis__AXISID if this is a translation axis, where *i* is the value of _diffrn_scan_frame.frame_number for which the value of _diffrn_scan_frame.frame_id agrees with the value of _diffrn_scan_frame_axis.frame_id
- _diffrn_scan_frame_axis.displacement_increment → see _diffrn_scan_frame_axis.axis_id
- _diffrn_scan_frame_axis.displacement_rstrt_incr → see _diffrn_scan_frame_axis.axis_id
- _diffrn_scan_frame_axis.reference_angle → see _diffrn_scan_frame_axis.axis_id
- _diffrn_scan_frame_axis.reference_displacement → see _diffrn_scan_frame_axis.axis_id
- _diffrn_scan_frame_axis.frame_id → used to find the frame number
- _diffrn_scan_frame_axis.variant category → NeXus does not handle variants at this time

4.15.4. The DIFFRN_SCAN_FRAME_MONITOR category

```
Data items in the DIFFRN_SCAN_FRAME_MONITOR category record
the values and details about each monitor for each frame of data
during a scan.

Each monitor value is uniquely identified by the combination of
the scan_id given by _diffrn_scan_frame.scan_id
the frame_id given by _diffrn_scan_frame_monitor.frame_id,
the monitor's detector_id given by
_diffrn_scan_frame_monitor.monitor_id,
and a 1-based ordinal given by _diffrn_scan_frame_monitor.id.

If there is only one frame for the scan, the value of
_diffrn_scan_frame_monitor.frame_id may be
omitted.

A single frame may have more than one monitor value, and each
monitor value may be the result of integration over the entire
frame integration time given by the value of
_diffrn_scan_frame.integration_time
or many monitor values may be reported over shorter times given
by the value of _diffrn_scan_frame_monitor.integration_time.  If
only one monitor value for a given monitor is collected during
the integration time of the frame, the value of
_diffrn_scan_frame_monitor.id may be
omitted.
```

- _diffrn_scan_frame_monitor.id MONID →
  selects the column (fast index) into which the monitor data will be stored
- _diffrn_scan_frame_monitor.detector_id DETECTORNAME →
  /instrument:NXinstrument
    /CBF_diffrn_scan_frame_monitor__DETECTORNAME_SCANID:NXmonitor
- _diffrn_scan_frame_monitor.scan_id SCANID →
  /instrument:NXinstrument
    /CBF_diffrn_scan_frame_monitor__DETECTORNAME_SCANID:NXmonitor
- _diffrn_data_frame_monitor.frame_id FRAMEID →
  selects the row (slow index) into which the monitor data will be stored by conversion of the FRAMEID to a frame number
- _diffrn_data_frame_monitor.integration_time INTEGRATIONTIME →
  /instrument:NXinstrument
    /CBF_diffrn_scan_frame_monitor__DETECTORNAME_SCANID:NXmonitor
      /count_time=INTEGRATIONTIMES
  stores INTEGRATIONTIME into slow index np==frame, nm==MONID
- _diffrn_data_frame_monitor.monitor_value MONITORVALUE →
  /instrument:NXinstrument
    /CBF_diffrn_scan_frame_monitor__DETECTORNAME_SCANID:NXmonitor
      /data=MONITORVALUES
  stores MONITORVALUE into slow index np=frame, fast index nm=MONID
- _diffrn_data_frame_monitor.variant category → NeXus does not handle variants at this time

4.16. The MAP category

```
Data items in the MAP category record
the details of a maps. Maps record values of parameters,
such as density, that are functions of position within
a cell or are functions of orthogonal coordinates in
three space.

A map may is composed of one or more map segments
specified in the MAP\_SEGMENT category.
```

- _map.details → ??
- _map.diffrn_id → ??
- _map.entry_id → ??
- _map.id → ??
- _map.variant category → ??

To be mapped to NeXus in the future.

4.16.1. The MAP_SEGMENT category

```
Data items in the MAP_SEGMENT category record
the details about each segment (section or brick) of a map.
```

- _map_segment.array_id → ??
- _map_segment.array_section_id → ??
- _map_segment.binary_id → ??
- _map_segment.mask_array_id → ??
- _map_segment.mask_array_section_id → ??
- _map_segment.mask_binary_id → ??
- _map_segment.id → ??
- _map_segment.map_id → ??
- _map_segment.details → ??
- _map_segment.variant category → ??

To be mapped to NeXus in the future.

4.17. The VARIANT category

```
Data items in the VARIANT category record
the details about sets of VARIANTs of data items.

There is sometimes a need to allow for multiple versions of the
same data items in order to allow for refinements and corrections
to earlier assumptions, observations and calculations.  In order
to allow data sets to contain more than one VARIANT of the same
information, an optional ...variant data item as a pointer to
_variant.variant has been added to the key of every category,
as an implicit data item with a null (empty) default value.

All rows in a category with the same VARIANT value are considered
to be related to one another and to all rows in other categories
with the same VARIANT value.  For a given VARIANT, all such rows
are also considered to be related to all rows with a null VARIANT
value, except that a row with a null VARIANT value is for which
all other components of its key are identical to those entries
in another row with a non-null VARIANT value is not related the
the rows with that non-null VARIANT value.  This behavior is
similar to the convention for identifying alternate conformers
in an atom list.

An optional role may be specified for a VARIANT as the value of
_variant.role.  Possible roles are null, "preferred",
"raw data", "unsuccessful trial".

VARIANTs may carry an optional timestamp as the value of
_variant.timestamp.

VARIANTs may be related to other VARIANTs from which they were
derived by the value of _variant.variant_of

Further details about the VARIANT may be specified as the value
of _variant.details.

In order to allow VARIANT information from multiple datasets to
be combined, _variant.diffrn_id and/or _variant.entry_id may
be used.
```

- _variant.details → ??
- _variant.role → ??
- _variant.timestamp → ??
- _variant.variant → ??
- _variant.variant_of → ??

To be mapped to NeXus in the future.

5. Mapping from NeXus to CBF/imgCIF

Mapping from NeXus to CBF/imgCIF is matter of establishing appropriate tables and columns for each of the NeXus classes. In general, a NeXus class will correspond to a category, while the specific name will be part of the key of that category, usually the id of the category. The terms in NeXus may be defined in base classes (see `http://download.nexusformat.org/doc/html/classes/base_classes/index.html`) or in application definitions (see `http://download.nexusformat.org/doc/html/classes/applications/index.html`). We will consider them one at a time.

As with the mapping from CBF to NeXus, where we do both a complete faithful mapping of a CBF to a NeXus tree in CBF_cbf, and a more structured tag-by-tag mapping throughout the NeXus tree, in his direction we will address a complete faithful mapping of all features of a NeXus tree into a single CBF data block as well as a more structured item-by-item mapping.

5.1. Faithful Mapping of a NeXus Tree

A NeXus tree is a tree of HDF5 groups, each with an associated NeXus class. For the full faithful mapping in to a single CBF datablock, which will be given the name NeXus_Tree, each NeXus class will be mapped to a CBF category of the same name, and

each instance of that class will be mapped to a single row in that table. The unique identifier of a row, in the column NX_tree_path will be the rooted path to the particular NeXus class instance in the tree. Each path component will consist of the NeXus class composed with the actual name of the NeXus class instance in dotted notation, with an underscore before each NeXus class name. Each subgroup of a NeXus class instance (also a NeXus class instance) will be entered into a column named with name of the subgroup, with a value equal to the rooted path to that subgroup.

For example, in the NeXus tree

```
\entry:NXentry
  \instrument:NXinstrument
    \detector:NXdetector
```

the mapping would be to

```
_datablock_NeXus_Tree

loop_
_NXentry.NX_tree_path  "/_NXentry.entry"
_NXentry.instrument
"/_NXentry.entry"    "/_NXentry.entry/_NXinstrument.instrument"

loop_
_NXinstrument.NX_tree_path
_NXinstrument.detector
"/_NXentry.entry/_NXinstrument.instrument" "/_NXentry.entry/_NXinstrument.instrument/_NXdetector__detector"

_NXdetector.detector "/_NXentry.entry/_NXinstrument.instrument/_NXdetector__detector"
```

An attribute of a NeXus class other than the HDF5 attribute for the NeXus class name itself will be given a column name composed from the prefix "NX_class_attribute__" and the name of the attribute.

5.2. Mapping Fields

Fields share some of the characteristics of classes and some if the characteristics of attributes. When a rooted path to a field is needed in this discussion, the rooted path be composed of the NeXus class composed with the name of the NeXus class instance composed with the name of the field in dotted notation, with an underscore before each NeXus class name.

If a NeXus class instance is given in the NeXus tree by a link, rather than directly an extra tag having the name of the column with the suffix "__link" will be used to carry the link path to the target in addition to giving the path to the origin of the link.

The mapping of fields in NeXus class instances will be handled in one of two ways:

- If instances of a NeXus class are each permitted to contain multiple instances of a field, each field instance will be handled like a NeXus class instance, with the modification to rooted paths noted above. Each such field will be assigned a column name that is generic to the instances, such as "data" or "axis_poise", and a category name beginning with "NX_" will be generated to hold the field instance value and attribute values. The value in the column will be a list of the rooted paths to the field instances.
- if each instance of a NeXus class may only contain one unique instance of a field, then the field name will be used as the column name. The value of the column for a field will be the verbatim value of the field, using the CBF binary data type when needed, or CIF 2 lists and tuples when needed. Attributes of fields are handled by creating a column beginning with the name of the column of the field and appending the attribute name in dotted notation, converting the dots to double underscores.

For example, in the NeXus tree

```
\entry:NXentry
  \instrument:NXinstrument
    \detector:NXdetector
       \data=BINDATA
         \@signal=1
```

the mapping would be to

```
_datablock_NeXus_Tree


_loop_
_NXentry.NX_tree_path  "/_NXentry.entry"
_NXentry.instrument
"/_NXentry.entry"    "/_NXentry.entry/_NXinstrument.instrument"

loop_
_NXinstrument.NX_tree_path
_NXinstrument.detector
"/_NXentry.entry/_NXinstrument.instrument" "/_NXentry.entry/_NXinstrument.instrument/_NXdetector__detector"

loop_
_NXdetector.NX_tree_path
_NXdetector.data      BINDATA
_NXdetector.data__signal

"/_NXentry.entry/_NXinstrument.instrument/_NXdetector__detector"  BINDATA  1
```

While, in the NeXus tree

```
/entry:NXentry
   /instrument:NXinstrument
      /CBF\_diffrn_detector__detector:NXdetector
         CBF_axis__DETECTOR_PITCH=[0.]
            @units="deg"
            @cbf_location="image_1.axis.vector.10"
            @depends_on="CBF_axis__DETECTOR_Y"
            @transformation_type="rotation"
            @vector=[-1, 0, 0]
         CBF_axis__DETECTOR_Y=[0.]
            @units="mm"
            @cbf_location="image_1.axis.vector.9"
            @depends_on="CBF_axis__DETECTOR_Z"
            @transformation_type="translation"
            @vector=[0, -1, 0]
         CBF_axis__DETECTOR_Z=[250.]
            @units="mm"
            @cbf_location="image_1.axis.vector.8"
            depends_on="."
            @transformation_type="translation"
            @vector= [0, 0, 1]
         CBF_axis__ELEMENT_X=[0.]
            @units="mm"
            @offset_units="mm"
            @cbf_location="image_1.axis.vector.11"
            @depends_on="CBF_axis__DETECTOR_PITCH"
            @transformation_type="translation"
            @vector= [-1, 0, 0]
            @offset= [-211.818, -217.322, 0]
         CBF_axis__ELEMENT_Y=[0.]
            @units="mm"
            @cbf_location=image_1.axis.vector.12
            @depends_on="CBF_axis__ELEMENT_X"
            @transformation_type="translation"
            @vector= [0, 1, 0]
```

the mapping would be to

```
_datablock_NeXus_Tree


_loop_
_NXentry.NX_tree_path  "/_NXentry.entry"
_NXentry.instrument
"/_NXentry.entry"   "/_NXentry.entry/_NXinstrument.instrument"

loop_
_NXinstrument.NX_tree_path
_NXinstrument.detector
"/_NXentry.entry/_NXinstrument.instrument" "/_NXentry.entry/_NXinstrument.instrument/_NXdetector__diffrn_detector__det

loop_
_NXdetector.NX_tree_path
_NXdetector.axis_poise

"/_NXentry.entry/_NXinstrument.instrument/_NXdetector__diffrn_detector__detector__axis_poise"
["CBF_axis__DETECTOR_PITCH","CBF_axis__DETECTOR_Y","CBF_axis__DETECTOR_Z","CBF_axis__ELEMENT_X","CBF_axis__ELEMENT_Y"]

loop_
_NX_axis_poise.NX_tree_path
_NX_axis_poise.value
_NX_axis_poise.units
_NX_axis_poise.offset_units
_NX_axis_poise.cbf_location
_NX_axis_poise.depends_on
_NX_axis_poise.transformation_type
_NX_axis_poise.vector
_NX_axis_poise.offset

"/_NXentry.entry/_NXinstrument.instrument/_NXdetector__diffrn_detector__detector__axis_poise__axis__DETECTOR_PITCH"
    0. "deg"    .    "image_1.axis.vector.10"     "axis__DETECTOR_Y"    "rotation" [-1,0,0]                      .
"/_NXentry.entry/_NXinstrument.instrument/_NXdetector__diffrn_detector__detector__axis_poise__axis__DETECTOR_Y"
    0.  "mm"    .    "image_1.axis.vector.9"      "axis__DETECTOR_Z" "translation" [0,-1,0]                     .
"/_NXentry.entry/_NXinstrument.instrument/_NXdetector__diffrn_detector__detector__axis_poise__axis__DETECTOR_Z"
  250.  "mm"    .    "image_1.axis.vector.8"                      .  "translation" [0,0,1]                       .
"/_NXentry.entry/_NXinstrument.instrument/_NXdetector__diffrn_detector__detector__axis_poise__axis__ELEMENT_X"
```

```
    0.  "mm"    "mm" "image_1.axis.vector.11" "axis__DETECTOR_PITCH" "translation" [-1,0,0]  [-211.818,-217.322,0]
"/_NXentry.entry/_NXinstrument.instrument/_NXdetector__diffrn_detector__detector__axis_poise__axis__ELEMENT_Y"
    0.  "mm"    .    "image_1.axis.vector.12"      "axis__ELEMENT_X" "translation" [0,1,0]                        .
```

While the full faithful mapping of the NeXus tree into CBF is primarily for development and debugging, portions of it will also become essentially the final detailed mapping. Notice, for example, that the NX_axis_poise category in the faithful mapping has all the information needed to populate the standard CBF axis category will minimal transformations. In the final detailed mapping, in addition to the NX_tree_path, each category will have a "NX_id" column to provide a unique identifier when a CBF originates the data and no path is available to use as the key. Normally this will be the last component of the path.

6. NeXus Base Class Mapping

The details of the NeXus base classes are provided at
`http://download.nexusformat.org/doc/html/classes/base_classes/index.html`
which should be consulted for getails.

6.1. NXaperture

Template of a beamline aperture.

```
NXaperture (base class, version 1.0)
  description:NX_CHAR
  material:NX_CHAR
  NXgeometry
  NXgeometry
  NXnote
```

- APERTURE:NXaperture →
  _NXaperture.NX_tree_path NEXUSTREEPATH
  _NXaperture.NX_id APERTURE
  _NXaperture.NX_scan_id SCANID
  _NXaperture.NX_diffrn_id DIFFRNID
  _NXaperture.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXaperture__APERTURE" where APERTURE is the name of this group, typically "aperture". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- description:NX_CHAR=DESCRIPTION →
  _NXaperture.description DESCRIPTION
- material:NX_CHAR=MATERIAL →
  _NXaperture.material MATERIAL
- geometry_aperture:NXgeometry →
  _NXaperture.NXgeometry_id geometry_aperture
- geometry_blades:NXgeometry →
  _NXaperture.NXgeometry_id geometry_blades
- note:NXnote →
  _NXaperture.NXnote_id note

6.2. NXattenuator

Description of a device that reduces the intensity of a beam by attenuation. If uncertain whether to use NXfilter (band-pass filter) or NXattenuator (reduces beam intensity), then choose NXattenuator.

```
NXattenuator (base class, version 1.0)
  absorption_cross_section:NX_FLOAT
  attenuator_transmission:NX_FLOAT
  distance:NX_FLOAT
  scattering_cross_section:NX_FLOAT
  status:NX_CHAR
    @time
  thickness:NX_FLOAT
  type:NX_CHAR
```

- ATTENUATOR:NXattenuator →
  _NXattenuator.NX_tree_path NEXUSTREEPATH
  _NXattenuator.NX_id ATTENUATOR
  _NXattenuator.NX_scan_id SCANID
  _NXattenuator.NX_diffrn_id DIFFRNID
  _NXattenuator.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXattenuator__ATTENUATOR" where ATTENUATOR is the name of this group, typically "attenuator". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- absorption_cross_section:NX_FLOAT=ABSORPTION_CROSS_SECTION →
  _NXattenuator.absorption_cross_section ABSORPTION_CROSS_SECTION
- attenuator_transmission:NX_FLOAT=ATTENUATOR_TRANSMISSION →
  _NXattenuator.attenuator_transmission ATTENUATOR_TRANSMISSION
- distance:NX_FLOAT=DISTANCE →
  _NXattenuator.distance DISTANCE

- scattering_cross_section:NX_FLOAT=SCATTERING_CROSS_SECTION →
  _NXattenuator.scattering_cross_section SCATTERING_CROSS_SECTION
- status:NX_CHAR=STATUS →
  _NXattenuator.status STATUS
- @time=TIME →
  _NXattenuator.status__time TIME
- thickness:NX_FLOAT=THICKNESS →
  _NXattenuator.thickness THICKNESS
- type:NX_CHAR=TYPE →
  _NXattenuator.type TYPE

6.3. NXbeam

Template of the state of the neutron or X-ray beam at any location. It will be referenced by beamline component groups within the NXinstrument group or by the NXsample group. Note that variables such as the incident energy could be scalar values or arrays. This group is especially valuable in storing the results of instrument simulations in which it is useful to specify the beam profile, time distribution etc. at each beamline component. Otherwise, its most likely use is in the NXsample group in which it defines the results of the neutron scattering by the sample, e.g., energy transfer, polarizations.

Note: there has been a recent NIAC discussion on this class, in which Tobias Richter pointed out that the polarization is unspecified, and the Stokes parameters would be 4 array position, not 2. In addition, there was a "j" index floating around and the 2-dimensional arrays appear to have been transposed. The version has added the 2-parameter Denzo polarization parameters in the order norm, then ratio. In addition we have added the Stokes parameters in the order I, Q, U, V. We propose that the unspecified polarization be deprecated.

```
NXbeam (base class, version 1.0)
  distance:NX_FLOAT
  energy_transfer:NX_FLOAT[i]
  final_beam_divergence:NX_FLOAT[i,2]
  final_energy:NX_FLOAT[i]
  final_polarization:NX_FLOAT[i,2]
  final_polarization_Denzo:NX_FLOAT[i,2]
  final_polarization_Stokes:NX_FLOAT[i,4]
  final_wavelength:NX_FLOAT[i]
  final_wavelength_spread:NX_FLOAT[i]
  flux:NX_FLOAT[i]
  incident_beam_divergence:NX_FLOAT[i,2]
  incident_energy:NX_FLOAT[i]
  incident_polarization:NX_FLOAT[i,2]
  incident_polarization_Denzo:NX_FLOAT[i,2]
  incident_polarization_Stokes:NX_FLOAT[i,4]
  incident_wavelength:NX_FLOAT[i]
  incident_wavelength_spread:NX_FLOAT[i]
  NXdata
```

- BEAM:NXbeam →
  _NXbeam.NX_tree_path NEXUSTREEPATH
  _NXbeam.NX_id BEAM
  _NXbeam.NX_scan_id SCANID
  _NXbeam.NX_diffrn_id DIFFRNID
  _NXbeam.NX_entry_id ENTRYID

  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXbeam__BEAM" where BEAM is the name of this group, typically "beam". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- distance:NX_FLOAT=DISTANCE →
  _NXbeam.distance DISTANCE

  is an alias for
  _diffrn_measurement.sample_detector_distance DISTANCE
- energy_transfer:NX_FLOAT[i]=ENERGY_TRANSFER →
  _NXbeam.energy_transfer ENERGY_TRANSFER
- final_beam_divergence:NX_FLOAT[i,2]=FINAL_BEAM_DIVERGENCE →
  _NXbeam.final_beam_divergence FINAL_BEAM_DIVERGENCE
- final_energy:NX_FLOAT[i]=FINAL_ENERGY →
  _NXbeam.final_energy FINAL_ENERGY
- final_polarization:NX_FLOAT[i,2]=FINAL_POLARIZATION →
  _NXbeam.final_polarization FINAL_POLARIZATION
- final_polarization_Denzo:NX_FLOAT[i,2]=FINAL_POLARIZATION_DENZO →
  _NXbeam.final_polarization FINAL_POLARIZATION_DENZO
- final_polarization_Stokes:NX_FLOAT[i,4]=FINAL_POLARIZATION_STOKES →
  _NXbeam.final_polarization FINAL_POLARIZATION_STOKES
- final_wavelength:NX_FLOAT[i]=FINAL_WAVELENGTH →
  _NXbeam.final_wavelength FINAL_WAVELENGTH
- final_wavelength_spread:NX_FLOAT[i]=FINAL_WAVELENGTH_SPREAD →
  _NXbeam.final_wavelength_spread FINAL_WAVELENGTH_SPREAD
- flux:NX_FLOAT[i]=FLUX →
  _NXbeam.flux FLUX
- incident_beam_divergence:NX_FLOAT[i,2]=INCIDENT_BEAM_DIVERGENCE →
  _NXbeam.incident_beam_divergence INCIDENT_BEAM_DIVERGENCE
- incident_energy:NX_FLOAT[i]=INCIDENT_ENERGY →
  _NXbeam.incident_energy INCIDENT_ENERGY

- incident_polarization:NX_FLOAT[i,2]=INCIDENT_POLARIZATION →
  _NXbeam.incident_polarization INCIDENT_POLARIZATION
- incident_polarization_Denzo:NX_FLOAT[i,4]=INCIDENT_POLARIZATION_DENZO →
  _NXbeam.incident_polarization_Stokes INCIDENT_POLARIZATION_DENZO
- incident_polarization_Stokes:NX_FLOAT[i,4]=INCIDENT_POLARIZATION_STOKES →
  _NXbeam.incident_polarization_Stokes INCIDENT_POLARIZATION_STOKES
- incident_wavelength:NX_FLOAT[i]=INCIDENT_WAVELENGTH →
  _NXbeam.incident_wavelength INCIDENT_WAVELENGTH
- incident_wavelength_spread:NX_FLOAT[i]=INCIDENT_WAVELENGTH_SPREAD →
  _NXbeam.incident_wavelength_spread INCIDENT_WAVELENGTH_SPREAD
- data:NXdata →
  _NXbeam.NXdata_id data

The final Denzo polarization from the beam component immediately prior to the beam being incident on the sample should agree with the values of the CBF tags _diffrn_radiation.polarizn_source_norm and _diffrn_radiation.polarizn_source_ratio.

6.4. NXbeam_stop

```
A class for a beamstop. Beamstops and their positions are important for SANS and SAXS experiments.

NXbeam_stop (base class, version 1.0)
  description:NX_CHAR
  distance_to_detector:NX_FLOAT
  size:NX_FLOAT
  status:NX_CHAR
  x:NX_FLOAT
  y:NX_FLOAT
  NXgeometry
```

- BEAM_STOP:NXbeam_stop →
  _NXbeam_stop.NX_tree_path NEXUSTREEPATH
  _NXbeam_stop.NX_id BEAM_STOP
  _NXbeam_stop.NX_scan_id SCANID
  _NXbeam_stop.NX_diffrn_id DIFFRNID
  _NXbeam_stop.NX_entry_id ENTRYID

  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXbeam_stop__BEAM_STOP" where BEAM_STOP is the name of this group, typically "beam_stop". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- description:NX_CHAR=DESCRIPTION →
  _NXbeam_stop.description DESCRIPTION
- distance_to_detector:NX_FLOAT=DISTANCE_TO_DETECTOR →
  _NXbeam_stop.distance_to_detector DISTANCE_TO_DETECTOR
- size:NX_FLOAT=SIZE →
  _NXbeam_stop.size SIZE
- status:NX_CHAR=STATUS →
  _NXbeam_stop.status STATUS
- x:NX_FLOAT=X →
  _NXbeam_stop.x X
- y:NX_FLOAT=Y →
  _NXbeam_stop.y Y
- geometry1:NXgeometry →
  _NXbeam_stop.NXgeometry_id geometry1

6.5. NXbending_magnet

```
description for a bending magnet


NXbending_magnet (base class, version 1.0)
  accepted_photon_beam_divergence:NX_FLOAT
  bending_radius:NX_FLOAT
  critical_energy:NX_FLOAT
  divergence_x_minus:NX_FLOAT
  divergence_x_plus:NX_FLOAT
  divergence_y_minus:NX_FLOAT
  divergence_y_plus:NX_FLOAT
  magnetic_field:NX_FLOAT
  source_distance_x:NX_FLOAT
  source_distance_y:NX_FLOAT
  spectrum:NXdata
  NXgeometry
```

- BENDING_MAGNET:NXbending_magnet →
  _NXbending_magnet.NX_tree_path NEXUSTREEPATH
  _NXbending_magnet.NX_id BENDING_MAGNET
  _NXbending_magnet.NX_scan_id SCANID
  _NXbending_magnet.NX_diffrn_id DIFFRNID

_NXbending_magnet.NX_entry_id ENTRYID
where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXbending_magnet__BENDING_MAGNET" where BENDING_MAGNET is the name of this group, typically "bending_magnet". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.

- accepted_photon_beam_divergence:NX_FLOAT=ACCEPTED_PHOTON_BEAM_DIVERGENCE →
  _NXbending_magnet.accepted_photon_beam_divergence ACCEPTED_PHOTON_BEAM_DIVERGENCE
- bending_radius:NX_FLOAT=BENDING_RADIUS →
  _NXbending_magnet.bending_radius BENDING_RADIUS
- critical_energy:NX_FLOAT=CRITICAL_ENERGY →
  _NXbending_magnet.critical_energy CRITICAL_ENERGY
- divergence_x_minus:NX_FLOAT=DIVERGENCE_X_MINUS →
  _NXbending_magnet.divergence_x_minus DIVERGENCE_X_MINUS
- divergence_x_plus:NX_FLOAT=DIVERGENCE_X_PLUS →
  _NXbending_magnet.divergence_x_plus DIVERGENCE_X_PLUS
- divergence_y_minus:NX_FLOAT=DIVERGENCE_Y_MINUS →
  _NXbending_magnet.divergence_y_minus DIVERGENCE_Y_MINUS
- divergence_y_plus:NX_FLOAT=DIVERGENCE_Y_PLUS →
  _NXbending_magnet.divergence_y_plus DIVERGENCE_Y_PLUS
- magnetic_field:NX_FLOAT=MAGNETIC_FIELD →
  _NXbending_magnet.magnetic_field MAGNETIC_FIELD
- source_distance_x:NX_FLOAT=SOURCE_DISTANCE_X →
  _NXbending_magnet.source_distance_x SOURCE_DISTANCE_X
- source_distance_y:NX_FLOAT=SOURCE_DISTANCE_Y →
  _NXbending_magnet.source_distance_y SOURCE_DISTANCE_Y
- spectrum:NXdata →
  _NXbending_magnet.NXdata_id spectrum
- geometry1:NXgeometry →
  _NXbending_magnet.NXgeometry_id geometry1

6.6. NXcapillary

This is a dictionary of field names to use for describing a capillary as used in X-ray beamlines. Based on information provided by Gerd Wellenreuther.

```
NXcapillary (base class, version 1.0)
  accepting_aperture:NX_FLOAT
  focal_size:NX_FLOAT
  manufacturer:NX_CHAR
  maximum_incident_angle:NX_FLOAT
  type:NX_CHAR
  working_distance:NX_FLOAT
  gain:NXdata
  transmission:NXdata
```

- CAPILLARY:NXcapillary →
  _NXcapillary.NX_tree_path NEXUSTREEPATH
  _NXcapillary.NX_id CAPILLARY
  _NXcapillary.NX_scan_id SCANID
  _NXcapillary.NX_diffrn_id DIFFRNID
  _NXcapillary.NX_entry_id ENTRYID

  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXcapillary__CAPILLARY" where CAPILLARY is the name of this group, typically "capillary". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- accepting_aperture:NX_FLOAT=ACCEPTING_APERTURE →
  _NXcapillary.accepting_aperture ACCEPTING_APERTURE
- focal_size:NX_FLOAT=FOCAL_SIZE →
  _NXcapillary.focal_size FOCAL_SIZE
- manufacturer:NX_CHAR=MANUFACTURER →
  _NXcapillary.manufacturer MANUFACTURER
- maximum_incident_angle:NX_FLOAT=MAXIMUM_INCIDENT_ANGLE →
  _NXcapillary.maximum_incident_angle MAXIMUM_INCIDENT_ANGLE
- type:NX_CHAR=TYPE →
  _NXcapillary.type TYPE
- working_distance:NX_FLOAT=WORKING_DISTANCE →
  _NXcapillary.working_distance WORKING_DISTANCE
- gain:NXdata →
  _NXcapillary.NXdata_id gain
- transmission:NXdata →
  _NXcapillary.NXdata_id transmission

6.7. NXcharacterization

note: This base class may be removed in future releases of NXDL. If you have a use for this base class, please provide a description of your intended use to the NIAC (nexus-committee@nexusformat.org).

```
NXcharacterization (base class, version 1.0)
  @source
```

```
  @location
  @mime_type
  definition:NX_CHAR
    @version
    @URL
```

- CHARACTERIZATION:NXcharacterization →
  _NXcharacterization.NX_tree_path NEXUSTREEPATH
  _NXcharacterization.NX_id CHARACTERIZATION
  _NXcharacterization.NX_scan_id SCANID
  _NXcharacterization.NX_diffrn_id DIFFRNID
  _NXcharacterization.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXcharacterization__CHARACTERIZATION" where CHARACTERIZATION is the name of this group, typically "characterization". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- @source=SOURCE →
  _NXcharacterization.NX_class_attribute__source SOURCE
- @location=LOCATION →
  _NXcharacterization.NX_class_attribute__location LOCATION
- @mime_type=MIME_TYPE →
  _NXcharacterization.NX_class_attribute__mime_type MIME_TYPE
- definition:NX_CHAR=DEFINITION →
  _NXcharacterization.definition DEFINITION
- @version=VERSION →
  _NXcharacterization.definition__version VERSION
- @URL=URL →
  _NXcharacterization.definition__URL URL

6.8. NXcollection

Use NXcollection to gather together any set of terms. The original suggestion is to use this as a container class for the description of a beamline.

For NeXus validation, NXcollection will always generate a warning since it is always an optional group. Anything (groups, fields, or attributes) placed in an NXcollection group will not be validated.

```
NXcollection (contributed definition, version 1.0)
  beamline:NX_CHAR
```

- COLLECTION:NXcollection →
  _NXcollection.NX_tree_path NEXUSTREEPATH
  _NXcollection.NX_id COLLECTION
  _NXcollection.NX_scan_id SCANID
  _NXcollection.NX_diffrn_id DIFFRNID
  _NXcollection.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXcollection__COLLECTION" where COLLECTION is the name of this group, typically "collection". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- beamline:NX_CHAR=BEAMLINE →
  _NXcollection.beamline BEAMLINE

6.9. NXcollimator

Template of a beamline collimator.

```
NXcollimator (base class, version 1.0)
  absorbing_material:NX_CHAR
  blade_spacing:NX_FLOAT
  blade_thickness:NX_FLOAT
  divergence_x:NX_FLOAT
  divergence_y:NX_FLOAT
  frequency:NX_FLOAT
  soller_angle:NX_FLOAT
  transmitting_material:NX_CHAR
  type:NX_CHAR
  NXgeometry
  frequency_log:NXlog
```

- COLLIMATOR:NXcollimator →
  _NXcollimator.NX_tree_path NEXUSTREEPATH
  _NXcollimator.NX_id COLLIMATOR
  _NXcollimator.NX_scan_id SCANID
  _NXcollimator.NX_diffrn_id DIFFRNID
  _NXcollimator.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXcollimator__COLLIMATOR" where COLLIMATOR is the name of this group, typically "collimator". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.

- absorbing_material:NX_CHAR=ABSORBING_MATERIAL →
  _NXcollimator.absorbing_material ABSORBING_MATERIAL
- blade_spacing:NX_FLOAT=BLADE_SPACING →
  _NXcollimator.blade_spacing BLADE_SPACING
- blade_thickness:NX_FLOAT=BLADE_THICKNESS →
  _NXcollimator.blade_thickness BLADE_THICKNESS
- divergence_x:NX_FLOAT=DIVERGENCE_X →
  _NXcollimator.divergence_x DIVERGENCE_X
- divergence_y:NX_FLOAT=DIVERGENCE_Y →
  _NXcollimator.divergence_y DIVERGENCE_Y
- frequency:NX_FLOAT=FREQUENCY →
  _NXcollimator.frequency FREQUENCY
- soller_angle:NX_FLOAT=SOLLER_ANGLE →
  _NXcollimator.soller_angle SOLLER_ANGLE
- transmitting_material:NX_CHAR=TRANSMITTING_MATERIAL →
  _NXcollimator.transmitting_material TRANSMITTING_MATERIAL
- type:NX_CHAR=TYPE →
  _NXcollimator.type TYPE
- geometry1:NXgeometry →
  _NXcollimator.NXgeometry_id geometry1
- frequency_log:NXlog →
  _NXcollimator.NXlog_id frequency_log

See also _diffrn_radiation.div_x_source and _diffrn_radiation.div_y_source

6.10. NXcrystal

Template of a crystal monochromator or analyzer. Permits double bent monochromator comprised of multiple segments with anisotropic Gaussian mosaic.

If curvatures are set to zero or are absent, array is considered to be flat.

Scattering vector is perpendicular to surface. Crystal is oriented parallel to beam incident on crystal before rotation, and lies in vertical plane.

```
NXcrystal (base class, version 1.0)
  azimuthal_angle:NX_FLOAT[i]
  bragg_angle:NX_FLOAT[i]
  chemical_formula:NX_CHAR
  curvature_horizontal:NX_FLOAT
  curvature_vertical:NX_FLOAT
  cut_angle:NX_FLOAT
  cylindrical_orientation_angle:NX_NUMBER
  d_spacing:NX_FLOAT
  density:NX_NUMBER
  is_cylindrical:NX_BOOLEAN
  mosaic_horizontal:NX_FLOAT
  mosaic_vertical:NX_FLOAT
  order_no:NX_INT
  orientation_matrix:NX_FLOAT[3,3]
  polar_angle:NX_FLOAT[i]
  reflection:NX_INT[3]
  scattering_vector:NX_FLOAT
  segment_columns:NX_FLOAT
  segment_gap:NX_FLOAT
  segment_height:NX_FLOAT
  segment_rows:NX_FLOAT
  segment_thickness:NX_FLOAT
  segment_width:NX_FLOAT
  space_group:NX_CHAR
  temperature:NX_FLOAT
  temperature_coefficient:NX_FLOAT
  thickness:NX_FLOAT
  type:NX_CHAR
  unit_cell:NX_FLOAT[n_comp,6]
  unit_cell_a:NX_FLOAT
  unit_cell_alpha:NX_FLOAT
  unit_cell_b:NX_FLOAT
  unit_cell_beta:NX_FLOAT
  unit_cell_c:NX_FLOAT
  unit_cell_gamma:NX_FLOAT
  unit_cell_volume:NX_FLOAT
  usage:NX_CHAR
  wavelength:NX_FLOAT[i]
  reflectivity:NXdata
  transmission:NXdata
  NXgeometry
  temperature_log:NXlog
  shape:NXshape
```

- CRYSTAL:NXcrystal →
  _NXcrystal.NX_tree_path NEXUSTREEPATH

_NXcrystal.NX_id CRYSTAL
_NXcrystal.NX_scan_id SCANID
_NXcrystal.NX_diffrn_id DIFFRNID
_NXcrystal.NX_entry_id ENTRYID

where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXcrystal__CRYSTAL" where CRYSTAL is the name of this group, typically "crystal". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.

- azimuthal_angle:NX_FLOAT[i]=AZIMUTHAL_ANGLE →
  _NXcrystal.azimuthal_angle AZIMUTHAL_ANGLE
- bragg_angle:NX_FLOAT[i]=BRAGG_ANGLE →
  _NXcrystal.bragg_angle BRAGG_ANGLE
- chemical_formula:NX_CHAR=CHEMICAL_FORMULA →
  _NXcrystal.chemical_formula CHEMICAL_FORMULA
- curvature_horizontal:NX_FLOAT=CURVATURE_HORIZONTAL →
  _NXcrystal.curvature_horizontal CURVATURE_HORIZONTAL
- curvature_vertical:NX_FLOAT=CURVATURE_VERTICAL →
  _NXcrystal.curvature_vertical CURVATURE_VERTICAL
- cut_angle:NX_FLOAT=CUT_ANGLE →
  _NXcrystal.cut_angle CUT_ANGLE
- cylindrical_orientation_angle:NX_NUMBER=CYLINDRICAL_ORIENTATION_ANGLE →
  _NXcrystal.cylindrical_orientation_angle CYLINDRICAL_ORIENTATION_ANGLE
- d_spacing:NX_FLOAT=D_SPACING →
  _NXcrystal.d_spacing D_SPACING
- density:NX_NUMBER=DENSITY →
  _NXcrystal.density DENSITY
- is_cylindrical:NX_BOOLEAN=IS_CYLINDRICAL →
  _NXcrystal.is_cylindrical IS_CYLINDRICAL
- mosaic_horizontal:NX_FLOAT=MOSAIC_HORIZONTAL →
  _NXcrystal.mosaic_horizontal MOSAIC_HORIZONTAL
- mosaic_vertical:NX_FLOAT=MOSAIC_VERTICAL →
  _NXcrystal.mosaic_vertical MOSAIC_VERTICAL
- order_no:NX_INT=ORDER_NO →
  _NXcrystal.order_no ORDER_NO
- orientation_matrix:NX_FLOAT[3,3]=ORIENTATION_MATRIX →
  _NXcrystal.orientation_matrix ORIENTATION_MATRIX
- polar_angle:NX_FLOAT[i]=POLAR_ANGLE →
  _NXcrystal.polar_angle POLAR_ANGLE
- reflection:NX_INT[3]=REFLECTION →
  _NXcrystal.reflection REFLECTION
- scattering_vector:NX_FLOAT=SCATTERING_VECTOR →
  _NXcrystal.scattering_vector SCATTERING_VECTOR
- segment_columns:NX_FLOAT=SEGMENT_COLUMNS →
  _NXcrystal.segment_columns SEGMENT_COLUMNS
- segment_gap:NX_FLOAT=SEGMENT_GAP →
  _NXcrystal.segment_gap SEGMENT_GAP
- segment_height:NX_FLOAT=SEGMENT_HEIGHT →
  _NXcrystal.segment_height SEGMENT_HEIGHT
- segment_rows:NX_FLOAT=SEGMENT_ROWS →
  _NXcrystal.segment_rows SEGMENT_ROWS
- segment_thickness:NX_FLOAT=SEGMENT_THICKNESS →
  _NXcrystal.segment_thickness SEGMENT_THICKNESS
- segment_width:NX_FLOAT=SEGMENT_WIDTH →
  _NXcrystal.segment_width SEGMENT_WIDTH
- space_group:NX_CHAR=SPACE_GROUP →
  _NXcrystal.space_group SPACE_GROUP
- temperature:NX_FLOAT=TEMPERATURE →
  _NXcrystal.temperature TEMPERATURE
- temperature_coefficient:NX_FLOAT=TEMPERATURE_COEFFICIENT →
  _NXcrystal.temperature_coefficient TEMPERATURE_COEFFICIENT
- thickness:NX_FLOAT=THICKNESS →
  _NXcrystal.thickness THICKNESS
- type:NX_CHAR=TYPE →
  _NXcrystal.type TYPE
- unit_cell:NX_FLOAT[n_comp,6]=UNIT_CELL →
  _NXcrystal.unit_cell UNIT_CELL
- unit_cell_a:NX_FLOAT=UNIT_CELL_A →
  _NXcrystal.unit_cell_a UNIT_CELL_A
- unit_cell_alpha:NX_FLOAT=UNIT_CELL_ALPHA →
  _NXcrystal.unit_cell_alpha UNIT_CELL_ALPHA
- unit_cell_b:NX_FLOAT=UNIT_CELL_B →
  _NXcrystal.unit_cell_b UNIT_CELL_B
- unit_cell_beta:NX_FLOAT=UNIT_CELL_BETA →
  _NXcrystal.unit_cell_beta UNIT_CELL_BETA
- unit_cell_c:NX_FLOAT=UNIT_CELL_C →
  _NXcrystal.unit_cell_c UNIT_CELL_C
- unit_cell_gamma:NX_FLOAT=UNIT_CELL_GAMMA →

_NXcrystal.unit_cell_gamma UNIT_CELL_GAMMA
- unit_cell_volume:NX_FLOAT=UNIT_CELL_VOLUME →
  _NXcrystal.unit_cell_volume UNIT_CELL_VOLUME
- usage:NX_CHAR=USAGE →
  _NXcrystal.usage USAGE
- wavelength:NX_FLOAT[i]=WAVELENGTH →
  _NXcrystal.wavelength WAVELENGTH
- reflectivity:NXdata →
  _NXcrystal.NXdata_id reflectivity
- transmission:NXdata →
  _NXcrystal.NXdata_id transmission
- geometry1:NXgeometry →
  _NXcrystal.NXgeometry_id geometry1
- temperature_log:NXlog →
  _NXcrystal.NXlog_id temperature_log
- shape:NXshape →
  _NXcrystal.NXshape_id shape

6.11. NXdata

(required) NXdata is a template of plottable data and their dimension scales. It is mandatory that there is at least one NXdata group in each NXentry group. Note that the variable and data can be defined with different names. The signal and axes attribute of the data item define which items are plottable data and which are dimension scales.

Each NXdata group will consist of only one data set containing plottable data and their standard deviations. This data set may be of arbitrary rank up to a maximum of NX_MAXRANK=32. The plottable data will be identified by the attribute: signal=1 The plottable data will identify the dimension scale specification(s) in the axes attribute.

If available, the standard deviations of the data are to be stored in a data set of the same rank and dimensions, with the name errors.

For each data dimension, there should be a one-dimensional array of the same length. These one-dimensional arrays are the dimension scales of the data, i.e. the values of the independent variables at which the data is measured, such as scattering angle or energy transfer.

There are two methods of linking each data dimension to its respective dimension scale.

The preferred (and recommended) method uses the axes attribute to specify the names of each dimension scale.

The older method uses the axis attribute on each dimension scale to identify with an integer the axis whose value is the number of the dimension.

NXdata is used to implement one of the basic motivations in NeXus, to provide a default plot for the data of this NXentry. The actual data might be stored in another group and (hard) linked to the NXdata group.

```
NXdata (base class, version 1.0)
  data:NX_NUMBER[n]
    @signal
    @axes
    @uncertainties
    @long_name
  errors:NX_NUMBER[n]
  offset:NX_FLOAT
  scaling_factor:NX_FLOAT
  variable:NX_NUMBER[n]
    @long_name
    @distribution
    @first_good
    @last_good
    @axis
  variable_errors:NX_NUMBER[n]
  x:NX_FLOAT[nx]
  y:NX_FLOAT[ny]
  z:NX_FLOAT[nz]
```

- DATA:NXdata →
  _NXdata.NX_tree_path NEXUSTREEPATH
  _NXdata.NX_id DATA
  _NXdata.NX_scan_id SCANID
  _NXdata.NX_diffrn_id DIFFRNID
  _NXdata.NX_entry_id ENTRYID

  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXdata__DATA" where DATA is the name of this group, typically "data". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.

  _NXdata.NX_id DATA

  is an alias for

  _array_intensities.array_id DATA and requires that we have a row in ARRAY_DATA for which we have
  _array_data.array_id DATAID
- data:NX_NUMBER[n]=DATA →
  _NXdata.data DATA

  is an alias of _array_data.data DATA for a row of ARRAY_DATA for which we have _array_data.array_id DATAID
- @signal=SIGNAL →
  _NXdata.data__signal SIGNAL
- @axes=AXES →
  _NXdata.data__axes AXES

- @uncertainties=UNCERTAINTIES →
  _NXdata.data__uncertainties UNCERTAINTIES
- @long_name=LONG_NAME →
  _NXdata.data__long_name LONG_NAME
- errors:NX_NUMBER[n]=ERRORS →
  _NXdata.errors ERRORS
- offset:NX_FLOAT=OFFSET →
  _NXdata.offset OFFSET is an alias for _array_intensities.offset OFFSET
- scaling_factor:NX_FLOAT=SCALING_FACTOR →
  _NXdata.scaling_factor SCALING_FACTOR
  is an alias for _array_intensities.scaling SCALING_FACTOR
- variable:NX_NUMBER[n]=VARIABLE →
  _NXdata.variable VARIABLE
- @long_name=LONG_NAME →
  _NXdata.variable__long_name LONG_NAME
- @distribution=DISTRIBUTION →
  _NXdata.variable__distribution DISTRIBUTION
- @first_good=FIRST_GOOD →
  _NXdata.variable__first_good FIRST_GOOD
- @last_good=LAST_GOOD →
  _NXdata.variable__last_good LAST_GOOD
- @axis=AXIS →
  _NXdata.variable__axis AXIS
- variable_errors:NX_NUMBER[n]=VARIABLE_ERRORS →
  _NXdata.variable_errors VARIABLE_ERRORS
- x:NX_FLOAT[nx]=X →
  _NXdata.x X
- y:NX_FLOAT[ny]=Y →
  _NXdata.y Y
- z:NX_FLOAT[nz]=Z →
  _NXdata.z Z

These items have a relationship with the CBF ARRAY_INTENSITIES category, as noted above as aliases, Additional mappings into the various ARRAY categories should be established.

6.12. NXdetector

Template of a detector, detector bank, or multidetector.

```
NXdetector (base class, version 1.0)
  acquisition_mode:NX_CHAR
  angular_calibration:NX_FLOAT[i,j]
  angular_calibration_applied:NX_BOOLEAN
  azimuthal_angle:NX_FLOAT[np,i,j]
  beam_center_x:NX_FLOAT
  beam_center_y:NX_FLOAT
  bit_depth_readout:NX_INT
  calibration_date:NX_DATE_TIME
  count_time:NX_NUMBER[np]
  countrate_correction__applied:NX_BOOLEAN
  crate:NX_INT[i,j]
    @local_name
  data:NX_NUMBER[np,i,j,tof]
    @signal
    @axes
    @long_name
    @check_sum
    @link
  data_error:NX_NUMBER[np,i,j,tof]
    @units
    @link
  dead_time:NX_FLOAT[np,i,j]
  description:NX_CHAR
  detection_gas_path:NX_FLOAT
  detector_number:NX_INT[i,j]
  detector_readout_time:NX_FLOAT
  diameter:NX_FLOAT
  distance:NX_FLOAT[np,i,j]
  flatfield:NX_FLOAT[i,j]
  flatfield_applied:NX_BOOLEAN
  flatfield_error:NX_FLOAT[i,j]
  frame_start_number:NX_INT
  frame_time:NX_FLOAT[NP]
  gain_setting:NX_CHAR
  gas_pressure:NX_FLOAT[i,j]
  input:NX_INT[i,j]
    @local_name
  layout:NX_CHAR
  local_name:NX_CHAR
```

```
number_of_cycles:NX_INT
pixel_mask:NX_FLOAT[i,j]
pixel_mask_applied:NX_BOOLEAN
polar_angle:NX_FLOAT[np,i,j]
raw_time_of_flight:NX_INT[tof+1]
  @frequency
saturation_value:NX_INT
sensor_material:NX_CHAR
sensor_thickness:NX_FLOAT
sequence_number:NX_CHAR
slot:NX_INT[i,j]
  @local_name
solid_angle:NX_FLOAT[i,j]
threshold_energy:NX_FLOAT
time_of_flight:NX_FLOAT[tof+1]
  @axis
  @primary
  @long_name
  @link
trigger_dead_time:NX_FLOAT
trigger_delay_time:NX_FLOAT
type:NX_CHAR
x_pixel_offset:NX_FLOAT[i,j]
  @axis
  @primary
  @long_name
  @link
x_pixel_size:NX_FLOAT[i,j]
y_pixel_offset:NX_FLOAT[i,j]
  @axis
  @primary
  @long_name
y_pixel_size:NX_FLOAT[i,j]
NXcharacterization
efficiency:NXdata
  efficiency:NX_FLOAT[i,j,k]
  real_time:NX_NUMBER[i,j,k]
  wavelength:NX_FLOAT[i,j,k]
NXgeometry
calibration_method:NXnote
data_file:NXnote
```

- DETECTOR:NXdetector →
  _NXdetector.NX_tree_path NEXUSTREEPATH
  _NXdetector.NX_id DETECTOR
  _NXdetector.NX_scan_id SCANID
  _NXdetector.NX_diffrn_id DIFFRNID
  _NXdetector.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXdetector__DETECTOR" where DETECTOR is the name of this group, typically "detector". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- acquisition_mode:NX_CHAR=ACQUISITION_MODE →
  _NXdetector.acquisition_mode ACQUISITION_MODE
- angular_calibration:NX_FLOAT[i,j]=ANGULAR_CALIBRATION →
  _NXdetector.angular_calibration ANGULAR_CALIBRATION
- angular_calibration_applied:NX_BOOLEAN=ANGULAR_CALIBRATION_APPLIED →
  _NXdetector.angular_calibration_applied ANGULAR_CALIBRATION_APPLIED
- azimuthal_angle:NX_FLOAT[np,i,j]=AZIMUTHAL_ANGLE →
  _NXdetector.azimuthal_angle AZIMUTHAL_ANGLE
- beam_center_x:NX_FLOAT=BEAM_CENTER_X →
  _NXdetector.beam_center_x BEAM_CENTER_X
- beam_center_y:NX_FLOAT=BEAM_CENTER_Y →
  _NXdetector.beam_center_y BEAM_CENTER_Y
- bit_depth_readout:NX_INT=BIT_DEPTH_READOUT →
  _NXdetector.bit_depth_readout BIT_DEPTH_READOUT
- calibration_date:NX_DATE_TIME=CALIBRATION_DATE →
  _NXdetector.calibration_date CALIBRATION_DATE
- count_time:NX_NUMBER[np]=COUNT_TIME →
  _NXdetector.count_time COUNT_TIME
- countrate_correction__applied:NX_BOOLEAN=COUNTRATE_CORRECTION__APPLIED →
  _NXdetector.countrate_correction__applied COUNTRATE_CORRECTION__APPLIED
- crate:NX_INT[i,j]=CRATE →
  _NXdetector.crate CRATE
- @local_name=LOCAL_NAME →
  _NXdetector.crate__local_name LOCAL_NAME
- data:NX_NUMBER[np,i,j,tof]=DATA →
  _NXdetector.data DATA

- @signal=SIGNAL →
  _NXdetector.data__signal SIGNAL
- @axes=AXES →
  _NXdetector.data__axes AXES
- @long_name=LONG_NAME →
  _NXdetector.data__long_name LONG_NAME
- @check_sum=CHECK_SUM →
  _NXdetector.data__check_sum CHECK_SUM
- @link=LINK →
  _NXdetector.data__link LINK
- data_error:NX_NUMBER[np,i,j,tof]=DATA_ERROR →
  _NXdetector.data_error DATA_ERROR
- @units=UNITS →
  _NXdetector.data_error__units UNITS
- @link=LINK →
  _NXdetector.data_error__link LINK
- dead_time:NX_FLOAT[np,i,j]=DEAD_TIME →
  _NXdetector.dead_time DEAD_TIME

  is an alias for

  _diffrn_detector.dtime DEAD_TIME
- description:NX_CHAR=DESCRIPTION →
  _NXdetector.description DESCRIPTION
- detection_gas_path:NX_FLOAT=DETECTION_GAS_PATH →
  _NXdetector.detection_gas_path DETECTION_GAS_PATH
- detector_number:NX_INT[i,j]=DETECTOR_NUMBER →
  _NXdetector.detector_number DETECTOR_NUMBER
- detector_readout_time:NX_FLOAT=DETECTOR_READOUT_TIME →
  _NXdetector.detector_readout_time DETECTOR_READOUT_TIME
- diameter:NX_FLOAT=DIAMETER →
  _NXdetector.diameter DIAMETER
- distance:NX_FLOAT[np,i,j]=DISTANCE →
  _NXdetector.distance DISTANCE
- flatfield:NX_FLOAT[i,j]=FLATFIELD →
  _NXdetector.flatfield FLATFIELD
- flatfield_applied:NX_BOOLEAN=FLATFIELD_APPLIED →
  _NXdetector.flatfield_applied FLATFIELD_APPLIED
- flatfield_error:NX_FLOAT[i,j]=FLATFIELD_ERROR →
  _NXdetector.flatfield_error FLATFIELD_ERROR
- frame_start_number:NX_INT=FRAME_START_NUMBER →
  _NXdetector.frame_start_number FRAME_START_NUMBER
- frame_time:NX_FLOAT[NP]=FRAME_TIME →
  _NXdetector.frame_time FRAME_TIME
- gain_setting:NX_CHAR=GAIN_SETTING →
  _NXdetector.gain_setting GAIN_SETTING
- gas_pressure:NX_FLOAT[i,j]=GAS_PRESSURE →
  _NXdetector.gas_pressure GAS_PRESSURE
- input:NX_INT[i,j]=INPUT →
  _NXdetector.input INPUT
- @local_name=LOCAL_NAME →
  _NXdetector.input__local_name LOCAL_NAME
- layout:NX_CHAR=LAYOUT →
  _NXdetector.layout LAYOUT
- local_name:NX_CHAR=LOCAL_NAME →
  _NXdetector.local_name LOCAL_NAME
- number_of_cycles:NX_INT=NUMBER_OF_CYCLES →
  _NXdetector.number_of_cycles NUMBER_OF_CYCLES
- pixel_mask:NX_FLOAT[i,j]=PIXEL_MASK →
  _NXdetector.pixel_mask PIXEL_MASK
- pixel_mask_applied:NX_BOOLEAN=PIXEL_MASK_APPLIED →
  _NXdetector.pixel_mask_applied PIXEL_MASK_APPLIED
- polar_angle:NX_FLOAT[np,i,j]=POLAR_ANGLE →
  _NXdetector.polar_angle POLAR_ANGLE
- raw_time_of_flight:NX_INT[tof+1]=RAW_TIME_OF_FLIGHT →
  _NXdetector.raw_time_of_flight RAW_TIME_OF_FLIGHT
- @frequency=FREQUENCY →
  _NXdetector.raw_time_of_flight__frequency FREQUENCY
- saturation_value:NX_INT=SATURATION_VALUE →
  _NXdetector.saturation_value SATURATION_VALUE
- sensor_material:NX_CHAR=SENSOR_MATERIAL →
  _NXdetector.sensor_material SENSOR_MATERIAL
- sensor_thickness:NX_FLOAT=SENSOR_THICKNESS →
  _NXdetector.sensor_thickness SENSOR_THICKNESS
- sequence_number:NX_CHAR=SEQUENCE_NUMBER →
  _NXdetector.sequence_number SEQUENCE_NUMBER
- slot:NX_INT[i,j]=SLOT →
  _NXdetector.slot SLOT

- @local_name=LOCAL_NAME →
  _NXdetector.slot__local_name LOCAL_NAME
- solid_angle:NX_FLOAT[i,j]=SOLID_ANGLE →
  _NXdetector.solid_angle SOLID_ANGLE
- threshold_energy:NX_FLOAT=THRESHOLD_ENERGY →
  _NXdetector.threshold_energy THRESHOLD_ENERGY
- time_of_flight:NX_FLOAT[tof+1]=TIME_OF_FLIGHT →
  _NXdetector.time_of_flight TIME_OF_FLIGHT
- @axis=AXIS →
  _NXdetector.time_of_flight__axis AXIS
- @primary=PRIMARY →
  _NXdetector.time_of_flight__primary PRIMARY
- @long_name=LONG_NAME →
  _NXdetector.time_of_flight__long_name LONG_NAME
- @link=LINK →
  _NXdetector.time_of_flight__link LINK
- trigger_dead_time:NX_FLOAT=TRIGGER_DEAD_TIME →
  _NXdetector.trigger_dead_time TRIGGER_DEAD_TIME
- trigger_delay_time:NX_FLOAT=TRIGGER_DELAY_TIME →
  _NXdetector.trigger_delay_time TRIGGER_DELAY_TIME
- type:NX_CHAR=TYPE →
  _NXdetector.type TYPE
- x_pixel_offset:NX_FLOAT[i,j]=X_PIXEL_OFFSET →
  _NXdetector.x_pixel_offset X_PIXEL_OFFSET
- @axis=AXIS →
  _NXdetector.x_pixel_offset__axis AXIS
- @primary=PRIMARY →
  _NXdetector.x_pixel_offset__primary PRIMARY
- @long_name=LONG_NAME →
  _NXdetector.x_pixel_offset__long_name LONG_NAME
- @link=LINK →
  _NXdetector.x_pixel_offset__link LINK
- x_pixel_size:NX_FLOAT[i,j]=X_PIXEL_SIZE →
  _NXdetector.x_pixel_size X_PIXEL_SIZE
- y_pixel_offset:NX_FLOAT[i,j]=Y_PIXEL_OFFSET →
  _NXdetector.y_pixel_offset Y_PIXEL_OFFSET
- @axis=AXIS →
  _NXdetector.y_pixel_offset__axis AXIS
- @primary=PRIMARY →
  _NXdetector.y_pixel_offset__primary PRIMARY
- @long_name=LONG_NAME →
  _NXdetector.y_pixel_offset__long_name LONG_NAME
- y_pixel_size:NX_FLOAT[i,j]=Y_PIXEL_SIZE →
  _NXdetector.y_pixel_size Y_PIXEL_SIZE
- characterization1:NXcharacterization →
  _NXdetector.NXcharacterization_id characterization1
- efficiency:NXdata →
  _NXdetector.NXdata_id efficiency
- efficiency:NX_FLOAT[i,j,k]=EFFICIENCY →
  _NXdetector.efficiency EFFICIENCY
- real_time:NX_NUMBER[i,j,k]=REAL_TIME →
  _NXdetector.real_time REAL_TIME
- wavelength:NX_FLOAT[i,j,k]=WAVELENGTH →
  _NXdetector.wavelength WAVELENGTH

  is an alias for
  _diffrn_radiation_wavelength.wavelength WAVELENGTH
- geometry1:NXgeometry →
  _NXdetector.NXgeometry_id geometry1
- calibration_method:NXnote →
  _NXdetector.NXnote_id calibration_method
- data_file:NXnote →
  _NXdetector.NXnote_id data_file

6.13. NXdetector_group

```
NXdetector_group (base class, version 1.0)
  group_index:NX_INT[i]
  group_names:NX_CHAR
  group_parent:NX_INT[]
  group_type:NX_INT[]
```

- DETECTOR_GROUP:NXdetector_group →
  _NXdetector_group.NX_tree_path NEXUSTREEPATH
  _NXdetector_group.NX_id DETECTOR_GROUP
  _NXdetector_group.NX_scan_id SCANID
  _NXdetector_group.NX_diffrn_id DIFFRNID
  _NXdetector_group.NX_entry_id ENTRYID

  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the

name of the group, ending with "/NXdetector_group__DETECTOR_GROUP" where DETECTOR_GROUP is the name of this group, typically "detector_group". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.

- group_index:NX_INT[i]=GROUP_INDEX → _NXdetector_group.group_index GROUP_INDEX
- group_names:NX_CHAR=GROUP_NAMES → _NXdetector_group.group_names GROUP_NAMES
- group_parent:NX_INT[]=GROUP_PARENT → _NXdetector_group.group_parent GROUP_PARENT
- group_type:NX_INT[]=GROUP_TYPE → _NXdetector_group.group_type GROUP_TYPE

This is closely related to the DIFFRN_DETECTOR_ELEMENT category and a method of integration needs to be found.

6.14. NXdisk_chopper

```
NXdisk_chopper (base class, version 1.0)
  distance:NX_FLOAT
  pair_separation:NX_FLOAT
  phase:NX_FLOAT
  radius:NX_FLOAT
  ratio:NX_INT
  rotation_speed:NX_FLOAT
  slit_angle:NX_FLOAT
  slit_height:NX_FLOAT
  slits:NX_INT
  type:NX_CHAR
  wavelength_range:NX_FLOAT[2]
  NXgeometry
```

- DISK_CHOPPER:NXdisk_chopper →
  _NXdisk_chopper.NX_tree_path NEXUSTREEPATH
  _NXdisk_chopper.NX_id DISK_CHOPPER
  _NXdisk_chopper.NX_scan_id SCANID
  _NXdisk_chopper.NX_diffrn_id DIFFRNID
  _NXdisk_chopper.NX_entry_id ENTRYID

  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXdisk_chopper__DISK_CHOPPER" where DISK_CHOPPER is the name of this group, typically "disk_chopper". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- distance:NX_FLOAT=DISTANCE → _NXdisk_chopper.distance DISTANCE
- pair_separation:NX_FLOAT=PAIR_SEPARATION → _NXdisk_chopper.pair_separation PAIR_SEPARATION
- phase:NX_FLOAT=PHASE → _NXdisk_chopper.phase PHASE
- radius:NX_FLOAT=RADIUS → _NXdisk_chopper.radius RADIUS
- ratio:NX_INT=RATIO → _NXdisk_chopper.ratio RATIO
- rotation_speed:NX_FLOAT=ROTATION_SPEED → _NXdisk_chopper.rotation_speed ROTATION_SPEED
- slit_angle:NX_FLOAT=SLIT_ANGLE → _NXdisk_chopper.slit_angle SLIT_ANGLE
- slit_height:NX_FLOAT=SLIT_HEIGHT → _NXdisk_chopper.slit_height SLIT_HEIGHT
- slits:NX_INT=SLITS → _NXdisk_chopper.slits SLITS
- type:NX_CHAR=TYPE → _NXdisk_chopper.type TYPE
- wavelength_range:NX_FLOAT[2]=WAVELENGTH_RANGE → _NXdisk_chopper.wavelength_range WAVELENGTH_RANGE
- geometry1:NXgeometry → _NXdisk_chopper.NXgeometry_id geometry1

6.15. NXentry

```
NXentry (base class, version 1.0)
  @IDF_Version
  collection_description:NX_CHAR
  collection_identifier:NX_CHAR
  collection_time:NX_FLOAT
  definition:NX_CHAR
    @version
    @URL
  definition_local:NX_CHAR
    @version
    @URL
  duration:NX_INT
  end_time:NX_DATE_TIME
```

```
entry_identifier:NX_CHAR
experiment_description:NX_CHAR
experiment_identifier:NX_CHAR
pre_sample_flightpath:NX_FLOAT
program_name:NX_CHAR
  @version
  @configuration
revision:NX_CHAR
  @comment
run_cycle:NX_CHAR
start_time:NX_DATE_TIME
title:NX_CHAR
NXcharacterization
NXdata
NXinstrument
NXmonitor
experiment_documentation:NXnote
notes:NXnote
thumbnail:NXnote
  @mime_type
NXprocess
NXsample
NXsubentry
NXuser
```

- ENTRY:NXentry →
  _NXentry.NX_tree_path NEXUSTREEPATH
  _NXentry.NX_id ENTRY
  _NXentry.NX_scan_id SCANID
  _NXentry.NX_diffrn_id DIFFRNID
  _NXentry.NX_entry_id ENTRYID

  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXentry__ENTRY" where ENTRY is the name of this group, typically "entry". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- @IDF_Version=IDF_VERSION →
  _NXentry.NX_class_attribute__IDF_Version IDF_VERSION
- collection_description:NX_CHAR=COLLECTION_DESCRIPTION →
  _NXentry.collection_description COLLECTION_DESCRIPTION
- collection_identifier:NX_CHAR=COLLECTION_IDENTIFIER →
  _NXentry.collection_identifier COLLECTION_IDENTIFIER
- collection_time:NX_FLOAT=COLLECTION_TIME →
  _NXentry.collection_time COLLECTION_TIME
- definition:NX_CHAR=DEFINITION →
  _NXentry.definition DEFINITION
- @version=VERSION →
  _NXentry.definition__version VERSION
- @URL=URL →
  _NXentry.definition__URL URL
- definition_local:NX_CHAR=DEFINITION_LOCAL →
  _NXentry.definition_local DEFINITION_LOCAL
- @version=VERSION →
  _NXentry.definition_local__version VERSION
- @URL=URL →
  _NXentry.definition_local__URL URL
- duration:NX_INT=DURATION →
  _NXentry.duration DURATION
- end_time:NX_DATE_TIME=END_TIME →
  _NXentry.end_time END_TIME
- entry_identifier:NX_CHAR=ENTRY_IDENTIFIER →
  _NXentry.entry_identifier ENTRY_IDENTIFIER
- experiment_description:NX_CHAR=EXPERIMENT_DESCRIPTION →
  _NXentry.experiment_description EXPERIMENT_DESCRIPTION
- experiment_identifier:NX_CHAR=EXPERIMENT_IDENTIFIER →
  _NXentry.experiment_identifier EXPERIMENT_IDENTIFIER
- pre_sample_flightpath:NX_FLOAT=PRE_SAMPLE_FLIGHTPATH →
  _NXentry.pre_sample_flightpath PRE_SAMPLE_FLIGHTPATH
- program_name:NX_CHAR=PROGRAM_NAME →
  _NXentry.program_name PROGRAM_NAME
- @version=VERSION →
  _NXentry.program_name__version VERSION
- @configuration=CONFIGURATION →
  _NXentry.program_name__configuration CONFIGURATION
- revision:NX_CHAR=REVISION →
  _NXentry.revision REVISION
- @comment=COMMENT →
  _NXentry.revision__comment COMMENT
- run_cycle:NX_CHAR=RUN_CYCLE →
  _NXentry.run_cycle RUN_CYCLE

- start_time:NX_DATE_TIME=START_TIME →
  _NXentry.start_time START_TIME
- title:NX_CHAR=TITLE →
  _NXentry.title TITLE
- characterization1:NXcharacterization →
  _NXentry.NXcharacterization_id characterization1
- data1:NXdata →
  _NXentry.NXdata_id data1
- instrument1:NXinstrument →
  _NXentry.NXinstrument_id instrument1
- monitor1:NXmonitor →
  _NXentry.NXmonitor_id monitor1
- experiment_documentation:NXnote →
  _NXentry.NXnote_id experiment_documentation
- notes:NXnote →
  _NXentry.NXnote_id notes
- thumbnail:NXnote →
  _NXentry.NXnote_id thumbnail
- @mime_type=MIME_TYPE →
  _NXentry.title__mime_type MIME_TYPE
- process1:NXprocess →
  _NXentry.NXprocess_id process1
- sample1:NXsample →
  _NXentry.NXsample_id sample1
- subentry1:NXsubentry →
  _NXentry.NXsubentry_id subentry1
- user1:NXuser →
  _NXentry.NXuser_id user1

6.16. NXenvironment

```
NXenvironment (base class, version 1.0)
  description:NX_CHAR
  name:NX_CHAR
  program:NX_CHAR
  short_name:NX_CHAR
  type:NX_CHAR
  position:NXgeometry
  NXnote
  NXsensor
```

- ENVIRONMENT:NXenvironment →
  _NXenvironment.NX_tree_path NEXUSTREEPATH
  _NXenvironment.NX_id ENVIRONMENT
  _NXenvironment.NX_scan_id SCANID
  _NXenvironment.NX_diffrn_id DIFFRNID
  _NXenvironment.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXenvironment__ENVIRONMENT" where ENVIRONMENT is the name of this group, typically "environment". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- description:NX_CHAR=DESCRIPTION →
  _NXenvironment.description DESCRIPTION
- name:NX_CHAR=NAME →
  _NXenvironment.name NAME
- program:NX_CHAR=PROGRAM →
  _NXenvironment.program PROGRAM
- short_name:NX_CHAR=SHORT_NAME →
  _NXenvironment.short_name SHORT_NAME
- type:NX_CHAR=TYPE →
  _NXenvironment.type TYPE
- position:NXgeometry →
  _NXenvironment.NXgeometry_id position
- note1:NXnote →
  _NXenvironment.NXnote_id note1
- sensor1:NXsensor →
  _NXenvironment.NXsensor_id sensor1

6.17. NXevent_data

```
NXevent_data (base class, version 1.0)
  events_per_pulse:NX_INT[j]
  pixel_number:NX_INT[i]
  pulse_height:NX_FLOAT[i,k]
  pulse_time:NX_INT[j]
    @offset
  time_of_flight:NX_INT[i]
```

- EVENT_DATA:NXevent_data →
  _NXevent_data.NX_tree_path NEXUSTREEPATH

_NXevent_data.NX_id EVENT_DATA
_NXevent_data.NX_scan_id SCANID
_NXevent_data.NX_diffrn_id DIFFRNID
_NXevent_data.NX_entry_id ENTRYID

where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXevent_data__EVENT_DATA" where EVENT_DATA is the name of this group, typically "event_data". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.

- events_per_pulse:NX_INT[j]=EVENTS_PER_PULSE →
  _NXevent_data.events_per_pulse EVENTS_PER_PULSE
- pixel_number:NX_INT[i]=PIXEL_NUMBER →
  _NXevent_data.pixel_number PIXEL_NUMBER
- pulse_height:NX_FLOAT[i,k]=PULSE_HEIGHT →
  _NXevent_data.pulse_height PULSE_HEIGHT
- pulse_time:NX_INT[j]=PULSE_TIME →
  _NXevent_data.pulse_time PULSE_TIME
- @offset=OFFSET →
  _NXevent_data.pulse_time__offset OFFSET
- time_of_flight:NX_INT[i]=TIME_OF_FLIGHT →
  _NXevent_data.time_of_flight TIME_OF_FLIGHT

6.18. NXfermi_chopper

```
NXfermi_chopper (base class, version 1.0)
  absorbing_material:NX_CHAR
  distance:NX_FLOAT
  energy:NX_FLOAT
  height:NX_FLOAT
  number:NX_INT
  r_slit:NX_FLOAT
  radius:NX_FLOAT
  rotation_speed:NX_FLOAT
  slit:NX_FLOAT
  transmitting_material:NX_CHAR
  type:NX_CHAR
  wavelength:NX_FLOAT
  width:NX_FLOAT
  NXgeometry
```

- FERMI_CHOPPER:NXfermi_chopper →
  _NXfermi_chopper.NX_tree_path NEXUSTREEPATH
  _NXfermi_chopper.NX_id FERMI_CHOPPER
  _NXfermi_chopper.NX_scan_id SCANID
  _NXfermi_chopper.NX_diffrn_id DIFFRNID
  _NXfermi_chopper.NX_entry_id ENTRYID

  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXfermi_chopper__FERMI_CHOPPER" where FERMI_CHOPPER is the name of this group, typically "fermi_chopper". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- absorbing_material:NX_CHAR=ABSORBING_MATERIAL →
  _NXfermi_chopper.absorbing_material ABSORBING_MATERIAL
- distance:NX_FLOAT=DISTANCE →
  _NXfermi_chopper.distance DISTANCE
- energy:NX_FLOAT=ENERGY →
  _NXfermi_chopper.energy ENERGY
- height:NX_FLOAT=HEIGHT →
  _NXfermi_chopper.height HEIGHT
- number:NX_INT=NUMBER →
  _NXfermi_chopper.number NUMBER
- r_slit:NX_FLOAT=R_SLIT →
  _NXfermi_chopper.r_slit R_SLIT
- radius:NX_FLOAT=RADIUS →
  _NXfermi_chopper.radius RADIUS
- rotation_speed:NX_FLOAT=ROTATION_SPEED →
  _NXfermi_chopper.rotation_speed ROTATION_SPEED
- slit:NX_FLOAT=SLIT →
  _NXfermi_chopper.slit SLIT
- transmitting_material:NX_CHAR=TRANSMITTING_MATERIAL →
  _NXfermi_chopper.transmitting_material TRANSMITTING_MATERIAL
- type:NX_CHAR=TYPE →
  _NXfermi_chopper.type TYPE
- wavelength:NX_FLOAT=WAVELENGTH →
  _NXfermi_chopper.wavelength WAVELENGTH
- width:NX_FLOAT=WIDTH →
  _NXfermi_chopper.width WIDTH
- geometry1:NXgeometry →
  _NXfermi_chopper.NXgeometry_id geometry1

```
NXfilter (base class, version 1.0)
  chemical_formula:NX_CHAR
  coating_material:NX_CHAR
  coating_roughness:NX_FLOAT[nsurf]
  density:NX_NUMBER
  description:NX_CHAR
  m_value:NX_FLOAT
  orientation_matrix:NX_FLOAT[n_comp,3,3]
  status:NX_CHAR
  substrate_material:NX_CHAR
  substrate_roughness:NX_FLOAT
  substrate_thickness:NX_FLOAT
  temperature:NX_FLOAT
  thickness:NX_FLOAT
  unit_cell_a:NX_FLOAT
  unit_cell_alpha:NX_FLOAT
  unit_cell_b:NX_FLOAT
  unit_cell_beta:NX_FLOAT
  unit_cell_c:NX_FLOAT
  unit_cell_gamma:NX_FLOAT
  unit_cell_volume:NX_FLOAT[n_comp]
  transmission:NXdata
  NXgeometry
  temperature_log:NXlog
  sensor_type:NXsensor
```

- FILTER:NXfilter →
  _NXfilter.NX_tree_path NEXUSTREEPATH
  _NXfilter.NX_id FILTER
  _NXfilter.NX_scan_id SCANID
  _NXfilter.NX_diffrn_id DIFFRNID
  _NXfilter.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXfilter__FILTER" where FILTER is the name of this group, typically "filter". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- chemical_formula:NX_CHAR=CHEMICAL_FORMULA →
  _NXfilter.chemical_formula CHEMICAL_FORMULA
- coating_material:NX_CHAR=COATING_MATERIAL →
  _NXfilter.coating_material COATING_MATERIAL
- coating_roughness:NX_FLOAT[nsurf]=COATING_ROUGHNESS →
  _NXfilter.coating_roughness COATING_ROUGHNESS
- density:NX_NUMBER=DENSITY →
  _NXfilter.density DENSITY
- description:NX_CHAR=DESCRIPTION →
  _NXfilter.description DESCRIPTION
- m_value:NX_FLOAT=M_VALUE →
  _NXfilter.m_value M_VALUE
- orientation_matrix:NX_FLOAT[n_comp,3,3]=ORIENTATION_MATRIX →
  _NXfilter.orientation_matrix ORIENTATION_MATRIX
- status:NX_CHAR=STATUS →
  _NXfilter.status STATUS
- substrate_material:NX_CHAR=SUBSTRATE_MATERIAL →
  _NXfilter.substrate_material SUBSTRATE_MATERIAL
- substrate_roughness:NX_FLOAT=SUBSTRATE_ROUGHNESS →
  _NXfilter.substrate_roughness SUBSTRATE_ROUGHNESS
- substrate_thickness:NX_FLOAT=SUBSTRATE_THICKNESS →
  _NXfilter.substrate_thickness SUBSTRATE_THICKNESS
- temperature:NX_FLOAT=TEMPERATURE →
  _NXfilter.temperature TEMPERATURE
- thickness:NX_FLOAT=THICKNESS →
  _NXfilter.thickness THICKNESS
- unit_cell_a:NX_FLOAT=UNIT_CELL_A →
  _NXfilter.unit_cell_a UNIT_CELL_A
- unit_cell_alpha:NX_FLOAT=UNIT_CELL_ALPHA →
  _NXfilter.unit_cell_alpha UNIT_CELL_ALPHA
- unit_cell_b:NX_FLOAT=UNIT_CELL_B →
  _NXfilter.unit_cell_b UNIT_CELL_B
- unit_cell_beta:NX_FLOAT=UNIT_CELL_BETA →
  _NXfilter.unit_cell_beta UNIT_CELL_BETA
- unit_cell_c:NX_FLOAT=UNIT_CELL_C →
  _NXfilter.unit_cell_c UNIT_CELL_C
- unit_cell_gamma:NX_FLOAT=UNIT_CELL_GAMMA →
  _NXfilter.unit_cell_gamma UNIT_CELL_GAMMA
- unit_cell_volume:NX_FLOAT[n_comp]=UNIT_CELL_VOLUME →
  _NXfilter.unit_cell_volume UNIT_CELL_VOLUME
- transmission:NXdata →
  _NXfilter.NXdata_id transmission

- geometry1:NXgeometry →
  _NXfilter.NXgeometry_id geometry1
- temperature_log:NXlog →
  _NXfilter.NXlog_id temperature_log
- sensor_type:NXsensor →
  _NXfilter.NXsensor_id sensor_type

6.20. NXflipper

```
NXflipper (base class, version 1.0)
  comp_current:NX_FLOAT
  comp_turns:NX_FLOAT
  flip_current:NX_FLOAT
  flip_turns:NX_FLOAT
  guide_current:NX_FLOAT
  guide_turns:NX_FLOAT
  thickness:NX_FLOAT
  type:NX_CHAR
```

- FLIPPER:NXflipper →
  _NXflipper.NX_tree_path NEXUSTREEPATH
  _NXflipper.NX_id FLIPPER
  _NXflipper.NX_scan_id SCANID
  _NXflipper.NX_diffrn_id DIFFRNID
  _NXflipper.NX_entry_id ENTRYID

  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXflipper__FLIPPER" where FLIPPER is the name of this group, typically "flipper". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- comp_current:NX_FLOAT=COMP_CURRENT →
  _NXflipper.comp_current COMP_CURRENT
- comp_turns:NX_FLOAT=COMP_TURNS →
  _NXflipper.comp_turns COMP_TURNS
- flip_current:NX_FLOAT=FLIP_CURRENT →
  _NXflipper.flip_current FLIP_CURRENT
- flip_turns:NX_FLOAT=FLIP_TURNS →
  _NXflipper.flip_turns FLIP_TURNS
- guide_current:NX_FLOAT=GUIDE_CURRENT →
  _NXflipper.guide_current GUIDE_CURRENT
- guide_turns:NX_FLOAT=GUIDE_TURNS →
  _NXflipper.guide_turns GUIDE_TURNS
- thickness:NX_FLOAT=THICKNESS →
  _NXflipper.thickness THICKNESS
- type:NX_CHAR=TYPE →
  _NXflipper.type TYPE

6.21. NXgeometry

```
NXgeometry (base class, version 1.0)
  component_index:NX_INT
  description:NX_CHAR
  NXorientation
  NXshape
  NXtranslation
```

- GEOMETRY:NXgeometry →
  _NXgeometry.NX_tree_path NEXUSTREEPATH
  _NXgeometry.NX_id GEOMETRY
  _NXgeometry.NX_scan_id SCANID
  _NXgeometry.NX_diffrn_id DIFFRNID
  _NXgeometry.NX_entry_id ENTRYID

  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXgeometry__GEOMETRY" where GEOMETRY is the name of this group, typically "geometry". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- component_index:NX_INT=COMPONENT_INDEX →
  _NXgeometry.component_index COMPONENT_INDEX
- description:NX_CHAR=DESCRIPTION →
  _NXgeometry.description DESCRIPTION
- orientation1:NXorientation →
  _NXgeometry.NXorientation_id orientation1
- shape1:NXshape →
  _NXgeometry.NXshape_id shape1
- translation1:NXtranslation →
  _NXgeometry.NXtranslation_id translation1

6.22. NXguide

```
NXguide (base class, version 1.0)
  bend_angle_x:NX_FLOAT
  bend_angle_y:NX_FLOAT
```

```
  coating_material:NX_FLOAT[nsurf]
  coating_roughness:NX_FLOAT[nsurf]
  description:NX_CHAR
  external_material:NX_CHAR
  incident_angle:NX_FLOAT
  interior_atmosphere:NX_CHAR
  m_value:NX_FLOAT[nsurf]
  number_sections:NX_INT
  substrate_material:NX_FLOAT[nsurf]
  substrate_roughness:NX_FLOAT[nsurf]
  substrate_thickness:NX_FLOAT[nsurf]
  reflectivity:NXdata
    data:NX_NUMBER[nsurf,nwl]
      @signal
      @axes
    surface:NX_NUMBER[nsurf]
    wavelength:NX_NUMBER[nwl]
  NXgeometry
```

- GUIDE:NXguide →
  _NXguide.NX_tree_path NEXUSTREEPATH
  _NXguide.NX_id GUIDE
  _NXguide.NX_scan_id SCANID
  _NXguide.NX_diffrn_id DIFFRNID
  _NXguide.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXguide__GUIDE" where GUIDE is the name of this group, typically "guide". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- bend_angle_x:NX_FLOAT=BEND_ANGLE_X →
  _NXguide.bend_angle_x BEND_ANGLE_X
- bend_angle_y:NX_FLOAT=BEND_ANGLE_Y →
  _NXguide.bend_angle_y BEND_ANGLE_Y
- coating_material:NX_FLOAT[nsurf]=COATING_MATERIAL →
  _NXguide.coating_material COATING_MATERIAL
- coating_roughness:NX_FLOAT[nsurf]=COATING_ROUGHNESS →
  _NXguide.coating_roughness COATING_ROUGHNESS
- description:NX_CHAR=DESCRIPTION →
  _NXguide.description DESCRIPTION
- external_material:NX_CHAR=EXTERNAL_MATERIAL →
  _NXguide.external_material EXTERNAL_MATERIAL
- incident_angle:NX_FLOAT=INCIDENT_ANGLE →
  _NXguide.incident_angle INCIDENT_ANGLE
- interior_atmosphere:NX_CHAR=INTERIOR_ATMOSPHERE →
  _NXguide.interior_atmosphere INTERIOR_ATMOSPHERE
- m_value:NX_FLOAT[nsurf]=M_VALUE →
  _NXguide.m_value M_VALUE
- number_sections:NX_INT=NUMBER_SECTIONS →
  _NXguide.number_sections NUMBER_SECTIONS
- substrate_material:NX_FLOAT[nsurf]=SUBSTRATE_MATERIAL →
  _NXguide.substrate_material SUBSTRATE_MATERIAL
- substrate_roughness:NX_FLOAT[nsurf]=SUBSTRATE_ROUGHNESS →
  _NXguide.substrate_roughness SUBSTRATE_ROUGHNESS
- substrate_thickness:NX_FLOAT[nsurf]=SUBSTRATE_THICKNESS →
  _NXguide.substrate_thickness SUBSTRATE_THICKNESS
- reflectivity:NXdata →
  _NXguide.NXdata_id reflectivity
- data:NX_NUMBER[nsurf,nwl]=DATA →
  _NXguide.data DATA
- @signal=SIGNAL →
  _NXguide.data__signal SIGNAL
- @axes=AXES →
  _NXguide.data__axes AXES
- surface:NX_NUMBER[nsurf]=SURFACE →
  _NXguide.surface SURFACE
- wavelength:NX_NUMBER[nwl]=WAVELENGTH →
  _NXguide.wavelength WAVELENGTH
- geometry1:NXgeometry →
  _NXguide.NXgeometry_id geometry1

6.23. NXinsertion_device

```
NXinsertion_device (base class, version 1.0)
  bandwidth:NX_FLOAT
  energy:NX_FLOAT
  gap:NX_FLOAT
  harmonic:NX_INT
  k:NX_FLOAT
  length:NX_FLOAT
```

```
  magnetic_wavelength:NX_FLOAT
  phase:NX_FLOAT
  poles:NX_INT
  power:NX_FLOAT
  taper:NX_FLOAT
  type:NX_CHAR
  spectrum:NXdata
  NXgeometry
```

- INSERTION_DEVICE:NXinsertion_device →
  _NXinsertion_device.NX_tree_path NEXUSTREEPATH
  _NXinsertion_device.NX_id INSERTION_DEVICE
  _NXinsertion_device.NX_scan_id SCANID
  _NXinsertion_device.NX_diffrn_id DIFFRNID
  _NXinsertion_device.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXinsertion_device__INSERTION_DEVICE" where INSERTION_DEVICE is the name of this group, typically "insertion_device". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- bandwidth:NX_FLOAT=BANDWIDTH →
  _NXinsertion_device.bandwidth BANDWIDTH
- energy:NX_FLOAT=ENERGY →
  _NXinsertion_device.energy ENERGY
- gap:NX_FLOAT=GAP →
  _NXinsertion_device.gap GAP
- harmonic:NX_INT=HARMONIC →
  _NXinsertion_device.harmonic HARMONIC
- k:NX_FLOAT=K →
  _NXinsertion_device.k K
- length:NX_FLOAT=LENGTH →
  _NXinsertion_device.length LENGTH
- magnetic_wavelength:NX_FLOAT=MAGNETIC_WAVELENGTH →
  _NXinsertion_device.magnetic_wavelength MAGNETIC_WAVELENGTH
- phase:NX_FLOAT=PHASE →
  _NXinsertion_device.phase PHASE
- poles:NX_INT=POLES →
  _NXinsertion_device.poles POLES
- power:NX_FLOAT=POWER →
  _NXinsertion_device.power POWER
- taper:NX_FLOAT=TAPER →
  _NXinsertion_device.taper TAPER
- type:NX_CHAR=TYPE →
  _NXinsertion_device.type TYPE
- spectrum:NXdata →
  _NXinsertion_device.NXdata_id spectrum
- geometry1:NXgeometry →
  _NXinsertion_device.NXgeometry_id geometry1

6.24. NXinstrument

```
NXinstrument (base class, version 1.0)
  name:NX_CHAR
    @short_name
  NXaperture
  NXattenuator
  NXbeam
  NXbeam_stop
  NXbending_magnet
  NXcollimator
  NXcrystal
  NXdetector
  NXdisk_chopper
  NXfermi_chopper
  NXfilter
  NXflipper
  NXguide
  NXinsertion_device
  NXmirror
  NXmoderator
  NXpolarizer
  NXsource
  NXvelocity_selector
```

- INSTRUMENT:NXinstrument →
  _NXinstrument.NX_tree_path NEXUSTREEPATH
  _NXinstrument.NX_id INSTRUMENT
  _NXinstrument.NX_scan_id SCANID
  _NXinstrument.NX_diffrn_id DIFFRNID

_NXinstrument.NX_entry_id ENTRYID
where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXinstrument__INSTRUMENT" where INSTRUMENT is the name of this group, typically "instrument". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.

- name:NX_CHAR=NAME →
  _NXinstrument.name NAME
- @short_name=SHORT_NAME →
  _NXinstrument.name__short_name SHORT_NAME
- aperture1:NXaperture →
  _NXinstrument.NXaperture_id aperture1
- attenuator1:NXattenuator →
  _NXinstrument.NXattenuator_id attenuator1
- beam1:NXbeam →
  _NXinstrument.NXbeam_id beam1
- beam_stop1:NXbeam_stop →
  _NXinstrument.NXbeam_stop_id beam_stop1
- bending_magnet1:NXbending_magnet →
  _NXinstrument.NXbending_magnet_id bending_magnet1
- collimator1:NXcollimator →
  _NXinstrument.NXcollimator_id collimator1
- crystal1:NXcrystal →
  _NXinstrument.NXcrystal_id crystal1
- detector1:NXdetector →
  _NXinstrument.NXdetector_id detector1
- disk_chopper1:NXdisk_chopper →
  _NXinstrument.NXdisk_chopper_id disk_chopper1
- fermi_chopper1:NXfermi_chopper →
  _NXinstrument.NXfermi_chopper_id fermi_chopper1
- filter1:NXfilter →
  _NXinstrument.NXfilter_id filter1
- flipper1:NXflipper →
  _NXinstrument.NXflipper_id flipper1
- guide1:NXguide →
  _NXinstrument.NXguide_id guide1
- insertion_device1:NXinsertion_device →
  _NXinstrument.NXinsertion_device_id insertion_device1
- mirror1:NXmirror →
  _NXinstrument.NXmirror_id mirror1
- moderator1:NXmoderator →
  _NXinstrument.NXmoderator_id moderator1
- polarizer1:NXpolarizer →
  _NXinstrument.NXpolarizer_id polarizer1
- source1:NXsource →
  _NXinstrument.NXsource_id source1
- velocity_selector1:NXvelocity_selector →
  _NXinstrument.NXvelocity_selector_id velocity_selector1

6.25. NXlog

```
NXlog (base class, version 1.0)
  average_value:NX_FLOAT
  average_value_error:NX_FLOAT
  description:NX_CHAR
  duration:NX_FLOAT
  maximum_value:NX_FLOAT
  minimum_value:NX_FLOAT
  raw_value:NX_NUMBER
  time:NX_FLOAT
    @start
  value:NX_NUMBER
```

- LOG:NXlog →
  _NXlog.NX_tree_path NEXUSTREEPATH
  _NXlog.NX_id LOG
  _NXlog.NX_scan_id SCANID
  _NXlog.NX_diffrn_id DIFFRNID
  _NXlog.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXlog__LOG" where LOG is the name of this group, typically "log". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- average_value:NX_FLOAT=AVERAGE_VALUE →
  _NXlog.average_value AVERAGE_VALUE
- average_value_error:NX_FLOAT=AVERAGE_VALUE_ERROR →
  _NXlog.average_value_error AVERAGE_VALUE_ERROR
- description:NX_CHAR=DESCRIPTION →
  _NXlog.description DESCRIPTION

- duration:NX_FLOAT=DURATION →
  _NXlog.duration DURATION
- maximum_value:NX_FLOAT=MAXIMUM_VALUE →
  _NXlog.maximum_value MAXIMUM_VALUE
- minimum_value:NX_FLOAT=MINIMUM_VALUE →
  _NXlog.minimum_value MINIMUM_VALUE
- raw_value:NX_NUMBER=RAW_VALUE →
  _NXlog.raw_value RAW_VALUE
- time:NX_FLOAT=TIME →
  _NXlog.time TIME
- @start=START →
  _NXlog.time__start START
- value:NX_NUMBER=VALUE →
  _NXlog.value VALUE

6.26. NXmirror

```
NXmirror (base class, version 1.0)
  bend_angle_x:NX_FLOAT
  bend_angle_y:NX_FLOAT
  coating_material:NX_CHAR
  coating_roughness:NX_FLOAT
  description:NX_CHAR
  even_layer_density:NX_FLOAT
  even_layer_material:NX_CHAR
  external_material:NX_CHAR
  incident_angle:NX_FLOAT
  interior_atmosphere:NX_CHAR
  layer_thickness:NX_FLOAT
  m_value:NX_FLOAT
  odd_layer_density:NX_FLOAT
  odd_layer_material:NX_CHAR
  substrate_density:NX_FLOAT
  substrate_material:NX_CHAR
  substrate_roughness:NX_FLOAT
  substrate_thickness:NX_FLOAT
  type:NX_CHAR
  reflectivity:NXdata
  figure_data:NXdata
  NXgeometry
  shape:NXshape
```

- MIRROR:NXmirror →
  _NXmirror.NX_tree_path NEXUSTREEPATH
  _NXmirror.NX_id MIRROR
  _NXmirror.NX_scan_id SCANID
  _NXmirror.NX_diffrn_id DIFFRNID
  _NXmirror.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXmirror__MIRROR" where MIRROR is the name of this group, typically "mirror". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- bend_angle_x:NX_FLOAT=BEND_ANGLE_X →
  _NXmirror.bend_angle_x BEND_ANGLE_X
- bend_angle_y:NX_FLOAT=BEND_ANGLE_Y →
  _NXmirror.bend_angle_y BEND_ANGLE_Y
- coating_material:NX_CHAR=COATING_MATERIAL →
  _NXmirror.coating_material COATING_MATERIAL
- coating_roughness:NX_FLOAT=COATING_ROUGHNESS →
  _NXmirror.coating_roughness COATING_ROUGHNESS
- description:NX_CHAR=DESCRIPTION →
  _NXmirror.description DESCRIPTION
- even_layer_density:NX_FLOAT=EVEN_LAYER_DENSITY →
  _NXmirror.even_layer_density EVEN_LAYER_DENSITY
- even_layer_material:NX_CHAR=EVEN_LAYER_MATERIAL →
  _NXmirror.even_layer_material EVEN_LAYER_MATERIAL
- external_material:NX_CHAR=EXTERNAL_MATERIAL →
  _NXmirror.external_material EXTERNAL_MATERIAL
- incident_angle:NX_FLOAT=INCIDENT_ANGLE →
  _NXmirror.incident_angle INCIDENT_ANGLE
- interior_atmosphere:NX_CHAR=INTERIOR_ATMOSPHERE →
  _NXmirror.interior_atmosphere INTERIOR_ATMOSPHERE
- layer_thickness:NX_FLOAT=LAYER_THICKNESS →
  _NXmirror.layer_thickness LAYER_THICKNESS
- m_value:NX_FLOAT=M_VALUE →
  _NXmirror.m_value M_VALUE
- odd_layer_density:NX_FLOAT=ODD_LAYER_DENSITY →
  _NXmirror.odd_layer_density ODD_LAYER_DENSITY

- odd_layer_material:NX_CHAR=ODD_LAYER_MATERIAL →
  _NXmirror.odd_layer_material ODD_LAYER_MATERIAL
- substrate_density:NX_FLOAT=SUBSTRATE_DENSITY →
  _NXmirror.substrate_density SUBSTRATE_DENSITY
- substrate_material:NX_CHAR=SUBSTRATE_MATERIAL →
  _NXmirror.substrate_material SUBSTRATE_MATERIAL
- substrate_roughness:NX_FLOAT=SUBSTRATE_ROUGHNESS →
  _NXmirror.substrate_roughness SUBSTRATE_ROUGHNESS
- substrate_thickness:NX_FLOAT=SUBSTRATE_THICKNESS →
  _NXmirror.substrate_thickness SUBSTRATE_THICKNESS
- type:NX_CHAR=TYPE →
  _NXmirror.type TYPE
- reflectivity:NXdata →
  _NXmirror.NXdata_id reflectivity
- figure_data:NXdata →
  _NXmirror.NXdata_id figure_data
- geometry1:NXgeometry →
  _NXmirror.NXgeometry_id geometry1
- shape:NXshape →
  _NXmirror.NXshape_id shape

6.27. NXmoderator

```
NXmoderator (base class, version 1.0)
  coupled:NX_BOOLEAN
  coupling_material:NX_CHAR
  distance:NX_FLOAT
  poison_depth:NX_FLOAT
  poison_material:NX_CHAR
  temperature:NX_FLOAT
  type:NX_CHAR
  pulse_shape:NXdata
  NXgeometry
  temperature_log:NXlog
```

- MODERATOR:NXmoderator →
  _NXmoderator.NX_tree_path NEXUSTREEPATH
  _NXmoderator.NX_id MODERATOR
  _NXmoderator.NX_scan_id SCANID
  _NXmoderator.NX_diffrn_id DIFFRNID
  _NXmoderator.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXmoderator__MODERATOR" where MODERATOR is the name of this group, typically "moderator". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- coupled:NX_BOOLEAN=COUPLED →
  _NXmoderator.coupled COUPLED
- coupling_material:NX_CHAR=COUPLING_MATERIAL →
  _NXmoderator.coupling_material COUPLING_MATERIAL
- distance:NX_FLOAT=DISTANCE →
  _NXmoderator.distance DISTANCE
- poison_depth:NX_FLOAT=POISON_DEPTH →
  _NXmoderator.poison_depth POISON_DEPTH
- poison_material:NX_CHAR=POISON_MATERIAL →
  _NXmoderator.poison_material POISON_MATERIAL
- temperature:NX_FLOAT=TEMPERATURE →
  _NXmoderator.temperature TEMPERATURE
- type:NX_CHAR=TYPE →
  _NXmoderator.type TYPE
- pulse_shape:NXdata →
  _NXmoderator.NXdata_id pulse_shape
- geometry1:NXgeometry →
  _NXmoderator.NXgeometry_id geometry1
- temperature_log:NXlog →
  _NXmoderator.NXlog_id temperature_log

6.28. NXmonitor

```
NXmonitor (base class, version 1.0)
  count_time:NX_FLOAT
  data:NX_NUMBER[n]
    @signal
    @axes
  distance:NX_FLOAT
  efficiency:NX_NUMBER[]
  end_time:NX_DATE_TIME
  integral:NX_NUMBER
  mode:NX_CHAR
  preset:NX_NUMBER
```

```
  range:NX_FLOAT[2]
  sampled_fraction:NX_FLOAT
  start_time:NX_DATE_TIME
  time_of_flight:NX_FLOAT[]
  type:NX_CHAR
  NXgeometry
  integral_log:NXlog
```

- MONITOR:NXmonitor →
  _NXmonitor.NX_tree_path NEXUSTREEPATH
  _NXmonitor.NX_id MONITOR
  _NXmonitor.NX_scan_id SCANID
  _NXmonitor.NX_diffrn_id DIFFRNID
  _NXmonitor.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXmonitor__MONITOR" where MONITOR is the name of this group, typically "monitor". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- count_time:NX_FLOAT=COUNT_TIME →
  _NXmonitor.count_time COUNT_TIME
- data:NX_NUMBER[n]=DATA →
  _NXmonitor.data DATA
- @signal=SIGNAL →
  _NXmonitor.data_signal SIGNAL
- @axes=AXES →
  _NXmonitor.data_axes AXES
- distance:NX_FLOAT=DISTANCE →
  _NXmonitor.distance DISTANCE
- efficiency:NX_NUMBER[]=EFFICIENCY →
  _NXmonitor.efficiency EFFICIENCY
- end_time:NX_DATE_TIME=END_TIME →
  _NXmonitor.end_time END_TIME
- integral:NX_NUMBER=INTEGRAL →
  _NXmonitor.integral INTEGRAL
- mode:NX_CHAR=MODE →
  _NXmonitor.mode MODE
- preset:NX_NUMBER=PRESET →
  _NXmonitor.preset PRESET
- range:NX_FLOAT[2]=RANGE →
  _NXmonitor.range RANGE
- sampled_fraction:NX_FLOAT=SAMPLED_FRACTION →
  _NXmonitor.sampled_fraction SAMPLED_FRACTION
- start_time:NX_DATE_TIME=START_TIME →
  _NXmonitor.start_time START_TIME
- time_of_flight:NX_FLOAT[]=TIME_OF_FLIGHT →
  _NXmonitor.time_of_flight TIME_OF_FLIGHT
- type:NX_CHAR=TYPE →
  _NXmonitor.type TYPE
- geometry1:NXgeometry →
  _NXmonitor.NXgeometry_id geometry1
- integral_log:NXlog →
  _NXmonitor.NXlog_id integral_log

6.29. NXmonochromator

```
NXmonochromator (base class, version 1.0)
  energy:NX_FLOAT
  energy_error:NX_FLOAT
  wavelength:NX_FLOAT
  wavelength_error:NX_FLOAT
  NXcrystal
  distribution:NXdata
  geometry:NXgeometry
  NXvelocity_selector
```

- MONOCHROMATOR:NXmonochromator →
  _NXmonochromator.NX_tree_path NEXUSTREEPATH
  _NXmonochromator.NX_id MONOCHROMATOR
  _NXmonochromator.NX_scan_id SCANID
  _NXmonochromator.NX_diffrn_id DIFFRNID
  _NXmonochromator.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXmonochromator__MONOCHROMATOR" where MONOCHROMATOR is the name of this group, typically "monochromator". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- energy:NX_FLOAT=ENERGY →
  _NXmonochromator.energy ENERGY
- energy_error:NX_FLOAT=ENERGY_ERROR →
  _NXmonochromator.energy_error ENERGY_ERROR

- wavelength:NX_FLOAT=WAVELENGTH →
  _NXmonochromator.wavelength WAVELENGTH
- wavelength_error:NX_FLOAT=WAVELENGTH_ERROR →
  _NXmonochromator.wavelength_error WAVELENGTH_ERROR
- crystal1:NXcrystal →
  _NXmonochromator.NXcrystal_id crystal1
- distribution:NXdata →
  _NXmonochromator.NXdata_id distribution
- geometry:NXgeometry →
  _NXmonochromator.NXgeometry_id geometry
- velocity_selector1:NXvelocity_selector →
  _NXmonochromator.NXvelocity_selector_id velocity_selector1

6.30. NXnote

```
NXnote (base class, version 1.0)
  author:NX_CHAR
  data:NX_BINARY
  date:NX_DATE_TIME
  description:NX_CHAR
  file_name:NX_CHAR
  type:NX_CHAR
```

- NOTE:NXnote →
  _NXnote.NX_tree_path NEXUSTREEPATH
  _NXnote.NX_id NOTE
  _NXnote.NX_scan_id SCANID
  _NXnote.NX_diffrn_id DIFFRNID
  _NXnote.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXnote__NOTE" where NOTE is the name of this group, typically "note". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- author:NX_CHAR=AUTHOR →
  _NXnote.author AUTHOR
- data:NX_BINARY=DATA →
  _NXnote.data DATA
- date:NX_DATE_TIME=DATE →
  _NXnote.date DATE
- description:NX_CHAR=DESCRIPTION →
  _NXnote.description DESCRIPTION
- file_name:NX_CHAR=FILE_NAME →
  _NXnote.file_name FILE_NAME
- type:NX_CHAR=TYPE →
  _NXnote.type TYPE

6.31. NXorientation

```
NXorientation (base class, version 1.0)
  value:NX_FLOAT[numobj,6]
  NXgeometry
```

- ORIENTATION:NXorientation →
  _NXorientation.NX_tree_path NEXUSTREEPATH
  _NXorientation.NX_id ORIENTATION
  _NXorientation.NX_scan_id SCANID
  _NXorientation.NX_diffrn_id DIFFRNID
  _NXorientation.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXorientation__ORIENTATION" where ORIENTATION is the name of this group, typically "orientation". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- value:NX_FLOAT[numobj,6]=VALUE →
  _NXorientation.value VALUE
- geometry1:NXgeometry →
  _NXorientation.NXgeometry_id geometry1

6.32. NXparameters

```
NXparameters (base class, version 1.0)
  term:NX_CHAR
    @units
```

- PARAMETERS:NXparameters →
  _NXparameters.NX_tree_path NEXUSTREEPATH
  _NXparameters.NX_id PARAMETERS
  _NXparameters.NX_scan_id SCANID
  _NXparameters.NX_diffrn_id DIFFRNID
  _NXparameters.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXparameters__PARAMETERS" where PARAMETERS is the name of this group,

typically “parameters”. The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.

- term:NX_CHAR=TERM →
  _NXparameters.term TERM
- @units=UNITS →
  _NXparameters.term__units UNITS

6.33. NXpolarizer

```
NXpolarizer (base class, version 1.0)
  composition:NX_CHAR
  efficiency:NX_FLOAT
  reflection:NX_INT[3]
  type:NX_CHAR
```

- POLARIZER:NXpolarizer →
  _NXpolarizer.NX_tree_path NEXUSTREEPATH
  _NXpolarizer.NX_id POLARIZER
  _NXpolarizer.NX_scan_id SCANID
  _NXpolarizer.NX_diffrn_id DIFFRNID
  _NXpolarizer.NX_entry_id ENTRYID

  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with “/NXpolarizer__POLARIZER” where POLARIZER is the name of this group, typically “polarizer”. The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- composition:NX_CHAR=COMPOSITION →
  _NXpolarizer.composition COMPOSITION
- efficiency:NX_FLOAT=EFFICIENCY →
  _NXpolarizer.efficiency EFFICIENCY
- reflection:NX_INT[3]=REFLECTION →
  _NXpolarizer.reflection REFLECTION
- type:NX_CHAR=TYPE →
  _NXpolarizer.type TYPE

6.34. NXpositioner

```
NXpositioner (base class, version 1.0)
  acceleration_time:NX_NUMBER
  controller_record:NX_CHAR
  description:NX_CHAR
  name:NX_CHAR
  raw_value:NX_NUMBER[n]
  soft_limit_max:NX_NUMBER
  soft_limit_min:NX_NUMBER
  target_value:NX_NUMBER[n]
  tolerance:NX_NUMBER[n]
  value:NX_NUMBER[n]
  velocity:NX_NUMBER
```

- POSITIONER:NXpositioner →
  _NXpositioner.NX_tree_path NEXUSTREEPATH
  _NXpositioner.NX_id POSITIONER
  _NXpositioner.NX_scan_id SCANID
  _NXpositioner.NX_diffrn_id DIFFRNID
  _NXpositioner.NX_entry_id ENTRYID

  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with “/NXpositioner__POSITIONER” where POSITIONER is the name of this group, typically “positioner”. The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- acceleration_time:NX_NUMBER=ACCELERATION_TIME →
  _NXpositioner.acceleration_time ACCELERATION_TIME
- controller_record:NX_CHAR=CONTROLLER_RECORD →
  _NXpositioner.controller_record CONTROLLER_RECORD
- description:NX_CHAR=DESCRIPTION →
  _NXpositioner.description DESCRIPTION
- name:NX_CHAR=NAME →
  _NXpositioner.name NAME
- raw_value:NX_NUMBER[n]=RAW_VALUE →
  _NXpositioner.raw_value RAW_VALUE
- soft_limit_max:NX_NUMBER=SOFT_LIMIT_MAX →
  _NXpositioner.soft_limit_max SOFT_LIMIT_MAX
- soft_limit_min:NX_NUMBER=SOFT_LIMIT_MIN →
  _NXpositioner.soft_limit_min SOFT_LIMIT_MIN
- target_value:NX_NUMBER[n]=TARGET_VALUE →
  _NXpositioner.target_value TARGET_VALUE
- tolerance:NX_NUMBER[n]=TOLERANCE →
  _NXpositioner.tolerance TOLERANCE
- value:NX_NUMBER[n]=VALUE →
  _NXpositioner.value VALUE

- velocity:NX_NUMBER=VELOCITY →
  _NXpositioner.velocity VELOCITY

6.35. NXprocess

```
NXprocess (base class, version 1.0)
  date:NX_DATE_TIME
  program:NX_CHAR
  version:NX_CHAR
  NXnote
```

- PROCESS:NXprocess →
  _NXprocess.NX_tree_path NEXUSTREEPATH
  _NXprocess.NX_id PROCESS
  _NXprocess.NX_scan_id SCANID
  _NXprocess.NX_diffrn_id DIFFRNID
  _NXprocess.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXprocess__PROCESS" where PROCESS is the name of this group, typically "process". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- date:NX_DATE_TIME=DATE →
  _NXprocess.date DATE
- program:NX_CHAR=PROGRAM →
  _NXprocess.program PROGRAM
- version:NX_CHAR=VERSION →
  _NXprocess.version VERSION
- note1:NXnote →
  _NXprocess.NXnote_id note1

6.36. NXroot

```
NXroot (base class, version 1.0)
  @NX_class
  @file_time
  @file_name
  @file_update_time
  @NeXus_version
  @HDF_version
  @HDF5_Version
  @XML_version
  @creator
  NXentry
```

- ROOT:NXroot →
  _NXroot.NX_tree_path NEXUSTREEPATH
  _NXroot.NX_id ROOT
  _NXroot.NX_scan_id SCANID
  _NXroot.NX_diffrn_id DIFFRNID
  _NXroot.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the group, ending with "/NXroot__ROOT" where ROOT is the name of this group, typically "root". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- @NX_class=NX_CLASS →
  _NXroot.NX_class_attribute__NX_class NX_CLASS
- @file_time=FILE_TIME →
  _NXroot.NX_class_attribute__file_time FILE_TIME
- @file_name=FILE_NAME →
  _NXroot.NX_class_attribute__file_name FILE_NAME
- @file_update_time=FILE_UPDATE_TIME →
  _NXroot.NX_class_attribute__file_update_time FILE_UPDATE_TIME
- @NeXus_version=NEXUS_VERSION →
  _NXroot.NX_class_attribute__NeXus_version NEXUS_VERSION
- @HDF_version=HDF_VERSION →
  _NXroot.NX_class_attribute__HDF_version HDF_VERSION
- @HDF5_Version=HDF5_VERSION →
  _NXroot.NX_class_attribute__HDF5_Version HDF5_VERSION
- @XML_version=XML_VERSION →
  _NXroot.NX_class_attribute__XML_version XML_VERSION
- @creator=CREATOR →
  _NXroot.NX_class_attribute__creator CREATOR
- entry1:NXentry →
  _NXroot.NXentry_id entry1

6.37. NXsample

```
NXsample (base class, version 1.0)
  changer_position:NX_INT
  chemical_formula:NX_CHAR
  component:NX_CHAR
```

```
concentration:NX_FLOAT[n_comp]
density:NX_FLOAT[n_comp]
description:NX_CHAR
distance:NX_FLOAT
electric_field:NX_FLOAT[n_eField]
  @direction
external_DAC:NX_FLOAT
magnetic_field:NX_FLOAT[n_mField]
  @direction
mass:NX_FLOAT[n_comp]
name:NX_CHAR
orientation_matrix:NX_FLOAT[n_comp,3,3]
path_length:NX_FLOAT
path_length_window:NX_FLOAT
preparation_date:NX_DATE_TIME
pressure:NX_FLOAT[n_pField]
relative_molecular_mass:NX_FLOAT[n_comp]
rotation_angle:NX_FLOAT
sample_component:NX_CHAR
sample_orientation:NX_FLOAT[3]
scattering_length_density:NX_FLOAT[n_comp]
short_title:NX_CHAR
situation:NX_CHAR
stress_field:NX_FLOAT[n_sField]
  @direction
temperature:NX_FLOAT[n_Temp]
thickness:NX_FLOAT
type:NX_CHAR
unit_cell:NX_FLOAT[n_comp,6]
unit_cell_class:NX_CHAR
unit_cell_group:NX_CHAR
unit_cell_volume:NX_FLOAT[n_comp]
volume_fraction:NX_FLOAT[n_comp]
x_translation:NX_FLOAT
NXbeam
transmission:NXdata
temperature_env:NXenvironment
magnetic_field_env:NXenvironment
geometry:NXgeometry
temperature_log:NXlog
magnetic_field_log:NXlog
external_ADC:NXlog
```

- SAMPLE:NXsample →
  _NXsample.NX_tree_path NEXUSTREEPATH
  _NXsample.NX_id SAMPLE
  _NXsample.NX_scan_id SCANID
  _NXsample.NX_diffrn_id DIFFRNID
  _NXsample.NX_entry_id ENTRYID

  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the NeXus class instance, ending with "/NXsample__SAMPLE" where SAMPLE is the name of this NeXus class instance, typically "sample". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- changer_position:NX_INT=CHANGER_POSITION →
  _NXsample.changer_position CHANGER_POSITION
- chemical_formula:NX_CHAR=CHEMICAL_FORMULA →
  _NXsample.chemical_formula CHEMICAL_FORMULA
- component:NX_CHAR=COMPONENT →
  _NXsample.component COMPONENT
- concentration:NX_FLOAT[n_comp]=CONCENTRATION →
  _NXsample.concentration CONCENTRATION
- density:NX_FLOAT[n_comp]=DENSITY →
  _NXsample.density DENSITY
- description:NX_CHAR=DESCRIPTION →
  _NXsample.description DESCRIPTION
- distance:NX_FLOAT=DISTANCE →
  _NXsample.distance DISTANCE
- electric_field:NX_FLOAT[n_eField]=ELECTRIC_FIELD →
  _NXsample.electric_field ELECTRIC_FIELD
- @direction=DIRECTION →
  _NXsample.electric_field__direction DIRECTION
- external_DAC:NX_FLOAT=EXTERNAL_DAC →
  _NXsample.external_DAC EXTERNAL_DAC
- magnetic_field:NX_FLOAT[n_mField]=MAGNETIC_FIELD →
  _NXsample.magnetic_field MAGNETIC_FIELD
- @direction=DIRECTION →
  _NXsample.magnetic_field__direction DIRECTION

- mass:NX_FLOAT[n_comp]=MASS →
  _NXsample.mass MASS
- name:NX_CHAR=NAME →
  _NXsample.name NAME
- orientation_matrix:NX_FLOAT[n_comp,3,3]=ORIENTATION_MATRIX →
  _NXsample.orientation_matrix ORIENTATION_MATRIX
- path_length:NX_FLOAT=PATH_LENGTH →
  _NXsample.path_length PATH_LENGTH
- path_length_window:NX_FLOAT=PATH_LENGTH_WINDOW →
  _NXsample.path_length_window PATH_LENGTH_WINDOW
- preparation_date:NX_DATE_TIME=PREPARATION_DATE →
  _NXsample.preparation_date PREPARATION_DATE
- pressure:NX_FLOAT[n_pField]=PRESSURE →
  _NXsample.pressure PRESSURE
- relative_molecular_mass:NX_FLOAT[n_comp]=RELATIVE_MOLECULAR_MASS →
  _NXsample.relative_molecular_mass RELATIVE_MOLECULAR_MASS
- rotation_angle:NX_FLOAT=ROTATION_ANGLE →
  _NXsample.rotation_angle ROTATION_ANGLE
- sample_component:NX_CHAR=SAMPLE_COMPONENT →
  _NXsample.sample_component SAMPLE_COMPONENT
- sample_orientation:NX_FLOAT[3]=SAMPLE_ORIENTATION →
  _NXsample.sample_orientation SAMPLE_ORIENTATION
- scattering_length_density:NX_FLOAT[n_comp]=SCATTERING_LENGTH_DENSITY →
  _NXsample.scattering_length_density SCATTERING_LENGTH_DENSITY
- short_title:NX_CHAR=SHORT_TITLE →
  _NXsample.short_title SHORT_TITLE
- situation:NX_CHAR=SITUATION →
  _NXsample.situation SITUATION
- stress_field:NX_FLOAT[n_sField]=STRESS_FIELD →
  _NXsample.stress_field STRESS_FIELD
- @direction=DIRECTION →
  _NXsample.stress_field__direction DIRECTION
- temperature:NX_FLOAT[n_Temp]=TEMPERATURE →
  _NXsample.temperature TEMPERATURE
- thickness:NX_FLOAT=THICKNESS →
  _NXsample.thickness THICKNESS
- type:NX_CHAR=TYPE →
  _NXsample.type TYPE
- unit_cell:NX_FLOAT[n_comp,6]=UNIT_CELL →
  _NXsample.unit_cell UNIT_CELL
- unit_cell_class:NX_CHAR=UNIT_CELL_CLASS →
  _NXsample.unit_cell_class UNIT_CELL_CLASS
- unit_cell_group:NX_CHAR=UNIT_CELL_GROUP →
  _NXsample.unit_cell_group UNIT_CELL_GROUP
- unit_cell_volume:NX_FLOAT[n_comp]=UNIT_CELL_VOLUME →
  _NXsample.unit_cell_volume UNIT_CELL_VOLUME
- volume_fraction:NX_FLOAT[n_comp]=VOLUME_FRACTION →
  _NXsample.volume_fraction VOLUME_FRACTION
- x_translation:NX_FLOAT=X_TRANSLATION →
  _NXsample.x_translation X_TRANSLATION
- beam1:NXbeam →
  _NXsample.NXbeam_id beam1
- transmission:NXdata →
  _NXsample.NXdata_id transmission
- temperature_env:NXenvironment →
  _NXsample.NXenvironment_id temperature_env
- magnetic_field_env:NXenvironment →
  _NXsample.NXenvironment_id magnetic_field_env
- geometry:NXgeometry →
  _NXsample.NXgeometry_id geometry
- temperature_log:NXlog →
  _NXsample.NXlog_id temperature_log
- magnetic_field_log:NXlog →
  _NXsample.NXlog_id magnetic_field_log
- external_ADC:NXlog →
  _NXsample.NXlog_id external_ADC

6.38. NXsensor

```
NXsensor (base class, version 1.0)
  attached_to:NX_CHAR
  external_field_brief:NX_CHAR
  high_trip_value:NX_FLOAT
  low_trip_value:NX_FLOAT
  measurement:NX_CHAR
  model:NX_CHAR
  name:NX_CHAR
  run_control:NX_BOOLEAN
```

```
  short_name:NX_CHAR
  type:NX_CHAR
  value:NX_FLOAT[n]
  value_deriv1:NX_FLOAT[]
  value_deriv2:NX_FLOAT[]
  geometry:NXgeometry
  value_log:NXlog
  value_deriv1_log:NXlog
  value_deriv2_log:NXlog
  external_field_full:NXorientation
```

- SENSOR:NXsensor →
  _NXsensor.NX_tree_path NEXUSTREEPATH
  _NXsensor.NX_id SENSOR
  _NXsensor.NX_scan_id SCANID
  _NXsensor.NX_diffrn_id DIFFRNID
  _NXsensor.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the NeXus class instance, ending with "/NXsensor__SENSOR" where SENSOR is the name of this NeXus class instance, typically "sensor". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- attached_to:NX_CHAR=ATTACHED_TO →
  _NXsensor.attached_to ATTACHED_TO
- external_field_brief:NX_CHAR=EXTERNAL_FIELD_BRIEF →
  _NXsensor.external_field_brief EXTERNAL_FIELD_BRIEF
- high_trip_value:NX_FLOAT=HIGH_TRIP_VALUE →
  _NXsensor.high_trip_value HIGH_TRIP_VALUE
- low_trip_value:NX_FLOAT=LOW_TRIP_VALUE →
  _NXsensor.low_trip_value LOW_TRIP_VALUE
- measurement:NX_CHAR=MEASUREMENT →
  _NXsensor.measurement MEASUREMENT
- model:NX_CHAR=MODEL →
  _NXsensor.model MODEL
- name:NX_CHAR=NAME →
  _NXsensor.name NAME
- run_control:NX_BOOLEAN=RUN_CONTROL →
  _NXsensor.run_control RUN_CONTROL
- short_name:NX_CHAR=SHORT_NAME →
  _NXsensor.short_name SHORT_NAME
- type:NX_CHAR=TYPE →
  _NXsensor.type TYPE
- value:NX_FLOAT[n]=VALUE →
  _NXsensor.value VALUE
- value_deriv1:NX_FLOAT[]=VALUE_DERIV1 →
  _NXsensor.value_deriv1 VALUE_DERIV1
- value_deriv2:NX_FLOAT[]=VALUE_DERIV2 →
  _NXsensor.value_deriv2 VALUE_DERIV2
- geometry:NXgeometry →
  _NXsensor.NXgeometry_id geometry
- value_log:NXlog →
  _NXsensor.NXlog_id value_log
- value_deriv1_log:NXlog →
  _NXsensor.NXlog_id value_deriv1_log
- value_deriv2_log:NXlog →
  _NXsensor.NXlog_id value_deriv2_log
- external_field_full:NXorientation →
  _NXsensor.NXorientation_id external_field_full

6.39. NXshape

```
NXshape (base class, version 1.0)
  direction:NX_CHAR
  shape:NX_CHAR
  size:NX_FLOAT[numobj,nshapepar]
```

- SHAPE:NXshape →
  _NXshape.NX_tree_path NEXUSTREEPATH
  _NXshape.NX_id SHAPE
  _NXshape.NX_scan_id SCANID
  _NXshape.NX_diffrn_id DIFFRNID
  _NXshape.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the NeXus class instance, ending with "/NXshape__SHAPE" where SHAPE is the name of this NeXus class instance, typically "shape". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- direction:NX_CHAR=DIRECTION →
  _NXshape.direction DIRECTION
- shape:NX_CHAR=SHAPE →
  _NXshape.shape SHAPE

- size:NX_FLOAT[numobj,nshapepar]=SIZE →
  _NXshape.size SIZE

6.40. NXsource

```
NXsource (base class, version 1.0)
  bunch_distance:NX_FLOAT
  bunch_length:NX_FLOAT
  current:NX_FLOAT
  distance:NX_FLOAT
  emittance_x:NX_FLOAT
  emittance_y:NX_FLOAT
  energy:NX_FLOAT
  flux:NX_FLOAT
  frequency:NX_FLOAT
  last_fill:NX_NUMBER
    @time
  mode:NX_CHAR
  name:NX_CHAR
    @short_name
  number_of_bunches:NX_INT
  period:NX_FLOAT
  power:NX_FLOAT
  probe:NX_CHAR
  pulse_width:NX_FLOAT
  sigma_x:NX_FLOAT
  sigma_y:NX_FLOAT
  target_material:NX_CHAR
  top_up:NX_BOOLEAN
  type:NX_CHAR
  voltage:NX_FLOAT
  bunch_pattern:NXdata
    title:NX_CHAR
  pulse_shape:NXdata
  distribution:NXdata
  geometry:NXgeometry
  notes:NXnote
```

- SOURCE:NXsource →
  _NXsource.NX_tree_path NEXUSTREEPATH
  _NXsource.NX_id SOURCE
  _NXsource.NX_scan_id SCANID
  _NXsource.NX_diffrn_id DIFFRNID
  _NXsource.NX_entry_id ENTRYID

  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the NeXus class instance, ending with "/NXsource__SOURCE" where SOURCE is the name of this NeXus class instance, typically "source". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- bunch_distance:NX_FLOAT=BUNCH_DISTANCE →
  _NXsource.bunch_distance BUNCH_DISTANCE
- bunch_length:NX_FLOAT=BUNCH_LENGTH →
  _NXsource.bunch_length BUNCH_LENGTH
- current:NX_FLOAT=CURRENT →
  _NXsource.current CURRENT
- distance:NX_FLOAT=DISTANCE →
  _NXsource.distance DISTANCE
- emittance_x:NX_FLOAT=EMITTANCE_X →
  _NXsource.emittance_x EMITTANCE_X
- emittance_y:NX_FLOAT=EMITTANCE_Y →
  _NXsource.emittance_y EMITTANCE_Y
- energy:NX_FLOAT=ENERGY →
  _NXsource.energy ENERGY
- flux:NX_FLOAT=FLUX →
  _NXsource.flux FLUX
- frequency:NX_FLOAT=FREQUENCY →
  _NXsource.frequency FREQUENCY
- last_fill:NX_NUMBER=LAST_FILL →
  _NXsource.last_fill LAST_FILL
- @time=TIME →
  _NXsource.last_fill__time TIME
- mode:NX_CHAR=MODE →
  _NXsource.mode MODE
- name:NX_CHAR=NAME →
  _NXsource.name NAME
- @short_name=SHORT_NAME →
  _NXsource.name__short_name SHORT_NAME
- number_of_bunches:NX_INT=NUMBER_OF_BUNCHES →
  _NXsource.number_of_bunches NUMBER_OF_BUNCHES

- period:NX_FLOAT=PERIOD →
  _NXsource.period PERIOD
- power:NX_FLOAT=POWER →
  _NXsource.power POWER
- probe:NX_CHAR=PROBE →
  _NXsource.probe PROBE
- pulse_width:NX_FLOAT=PULSE_WIDTH →
  _NXsource.pulse_width PULSE_WIDTH
- sigma_x:NX_FLOAT=SIGMA_X →
  _NXsource.sigma_x SIGMA_X
- sigma_y:NX_FLOAT=SIGMA_Y →
  _NXsource.sigma_y SIGMA_Y
- target_material:NX_CHAR=TARGET_MATERIAL →
  _NXsource.target_material TARGET_MATERIAL
- top_up:NX_BOOLEAN=TOP_UP →
  _NXsource.top_up TOP_UP
- type:NX_CHAR=TYPE →
  _NXsource.type TYPE
- voltage:NX_FLOAT=VOLTAGE →
  _NXsource.voltage VOLTAGE

- bunch_pattern:NXdata →
  _NXsource.NXdata_id bunch_pattern
- title:NX_CHAR=TITLE →
  _NXsource.title TITLE
- pulse_shape:NXdata →
  _NXsource.NXdata_id pulse_shape
- distribution:NXdata →
  _NXsource.NXdata_id distribution
- geometry:NXgeometry →
  _NXsource.NXgeometry_id geometry
- notes:NXnote →
  _NXsource.NXnote_id notes

6.41. NXsubentry

```
NXsubentry (base class, version 1.0)
  @IDF_Version
  collection_description:NX_CHAR
  collection_identifier:NX_CHAR
  collection_time:NX_FLOAT
  definition:NX_CHAR
    @version
    @URL
  definition_local:NX_CHAR
    @version
    @URL
  duration:NX_INT
  end_time:NX_DATE_TIME
  entry_identifier:NX_CHAR
  experiment_description:NX_CHAR
  experiment_identifier:NX_CHAR
  pre_sample_flightpath:NX_FLOAT
  program_name:NX_CHAR
    @version
    @configuration
  revision:NX_CHAR
    @comment
  run_cycle:NX_CHAR
  start_time:NX_DATE_TIME
  title:NX_CHAR
  NXcharacterization
  NXdata
  NXinstrument
  NXmonitor
  experiment_documentation:NXnote
  notes:NXnote
  thumbnail:NXnote
    @mime_type
  NXprocess
  NXsample
  NXuser
```

- SUBENTRY:NXsubentry →
  _NXsubentry.NX_tree_path NEXUSTREEPATH
  _NXsubentry.NX_id SUBENTRY
  _NXsubentry.NX_scan_id SCANID
  _NXsubentry.NX_diffrn_id DIFFRNID
  _NXsubentry.NX_entry_id ENTRYID

where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the NeXus class instance, ending with "/NXsubentry__SUBENTRY" where SUBENTRY is the name of this NeXus class instance, typically "subentry". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.

- @IDF_Version=IDF_VERSION →
  _NXsubentry.NX_class_attribute__IDF_Version IDF_VERSION
- collection_description:NX_CHAR=COLLECTION_DESCRIPTION →
  _NXsubentry.collection_description COLLECTION_DESCRIPTION
- collection_identifier:NX_CHAR=COLLECTION_IDENTIFIER →
  _NXsubentry.collection_identifier COLLECTION_IDENTIFIER
- collection_time:NX_FLOAT=COLLECTION_TIME →
  _NXsubentry.collection_time COLLECTION_TIME
- definition:NX_CHAR=DEFINITION →
  _NXsubentry.definition DEFINITION
- @version=VERSION →
  _NXsubentry.definition__version VERSION
- @URL=URL →
  _NXsubentry.definition__URL URL
- definition_local:NX_CHAR=DEFINITION_LOCAL →
  _NXsubentry.definition_local DEFINITION_LOCAL
- @version=VERSION →
  _NXsubentry.definition_local__version VERSION
- @URL=URL →
  _NXsubentry.definition_local__URL URL
- duration:NX_INT=DURATION →
  _NXsubentry.duration DURATION
- end_time:NX_DATE_TIME=END_TIME →
  _NXsubentry.end_time END_TIME
- entry_identifier:NX_CHAR=ENTRY_IDENTIFIER →
  _NXsubentry.entry_identifier ENTRY_IDENTIFIER
- experiment_description:NX_CHAR=EXPERIMENT_DESCRIPTION →
  _NXsubentry.experiment_description EXPERIMENT_DESCRIPTION
- experiment_identifier:NX_CHAR=EXPERIMENT_IDENTIFIER →
  _NXsubentry.experiment_identifier EXPERIMENT_IDENTIFIER
- pre_sample_flightpath:NX_FLOAT=PRE_SAMPLE_FLIGHTPATH →
  _NXsubentry.pre_sample_flightpath PRE_SAMPLE_FLIGHTPATH
- program_name:NX_CHAR=PROGRAM_NAME →
  _NXsubentry.program_name PROGRAM_NAME
- @version=VERSION →
  _NXsubentry.program_name__version VERSION
- @configuration=CONFIGURATION →
  _NXsubentry.program_name__configuration CONFIGURATION
- revision:NX_CHAR=REVISION →
  _NXsubentry.revision REVISION
- @comment=COMMENT →
  _NXsubentry.revision__comment COMMENT
- run_cycle:NX_CHAR=RUN_CYCLE →
  _NXsubentry.run_cycle RUN_CYCLE
- start_time:NX_DATE_TIME=START_TIME →
  _NXsubentry.start_time START_TIME
- title:NX_CHAR=TITLE →
  _NXsubentry.title TITLE
- characterization1:NXcharacterization →
  _NXsubentry.NXcharacterization_id characterization1
- data1:NXdata →
  _NXsubentry.NXdata_id data1
- instrument1:NXinstrument →
  _NXsubentry.NXinstrument_id instrument1
- monitor1:NXmonitor →
  _NXsubentry.NXmonitor_id monitor1
- experiment_documentation:NXnote →
  _NXsubentry.NXnote_id experiment_documentation
- notes:NXnote →
  _NXsubentry.NXnote_id notes
- thumbnail:NXnote →
  _NXsubentry.NXnote_id thumbnail
- @mime_type=MIME_TYPE →
  _NXsubentry.title__mime_type MIME_TYPE
- process1:NXprocess →
  _NXsubentry.NXprocess_id process1
- sample1:NXsample →
  _NXsubentry.NXsample_id sample1
- user1:NXuser →
  _NXsubentry.NXuser_id user1

6.42. NXtranslation

`NXtranslation (base class, version 1.0)`

```
  distances:NX_FLOAT[numobj,3]
  geometry:NXgeometry
```

- TRANSLATION:NXtranslation →
  _NXtranslation.NX_tree_path NEXUSTREEPATH
  _NXtranslation.NX_id TRANSLATION
  _NXtranslation.NX_scan_id SCANID
  _NXtranslation.NX_diffrn_id DIFFRNID
  _NXtranslation.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the NeXus class instance, ending with "/NXtranslation__TRANSLATION" where TRANSLATION is the name of this NeXus class instance, typically "translation". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- distances:NX_FLOAT[numobj,3]=DISTANCES →
  _NXtranslation.distances DISTANCES
- geometry:NXgeometry →
  _NXtranslation.NXgeometry_id geometry

6.43. NXuser

```
NXuser (base class, version 1.0)
  address:NX_CHAR
  affiliation:NX_CHAR
  email:NX_CHAR
  facility_user_id:NX_CHAR
  fax_number:NX_CHAR
  name:NX_CHAR
  role:NX_CHAR
  telephone_number:NX_CHAR
```

- USER:NXuser →
  _NXuser.NX_tree_path NEXUSTREEPATH
  _NXuser.NX_id USER
  _NXuser.NX_scan_id SCANID
  _NXuser.NX_diffrn_id DIFFRNID
  _NXuser.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the NeXus class instance, ending with "/NXuser__USER" where USER is the name of this NeXus class instance, typically "user". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- address:NX_CHAR=ADDRESS →
  _NXuser.address ADDRESS
- affiliation:NX_CHAR=AFFILIATION →
  _NXuser.affiliation AFFILIATION
- email:NX_CHAR=EMAIL →
  _NXuser.email EMAIL
- facility_user_id:NX_CHAR=FACILITY_USER_ID →
  _NXuser.facility_user_id FACILITY_USER_ID
- fax_number:NX_CHAR=FAX_NUMBER →
  _NXuser.fax_number FAX_NUMBER
- name:NX_CHAR=NAME →
  _NXuser.name NAME
- role:NX_CHAR=ROLE →
  _NXuser.role ROLE
- telephone_number:NX_CHAR=TELEPHONE_NUMBER →
  _NXuser.telephone_number TELEPHONE_NUMBER

6.44. NXvelocity_selector

```
NXvelocity_selector (base class, version 1.0)
  height:NX_FLOAT
  length:NX_FLOAT
  num:NX_INT
  radius:NX_FLOAT
  rotation_speed:NX_FLOAT
  spwidth:NX_FLOAT
  table:NX_FLOAT
  twist:NX_FLOAT
  type:NX_CHAR
  wavelength:NX_FLOAT
  wavelength_spread:NX_FLOAT
  width:NX_FLOAT
  geometry:NXgeometry
```

- VELOCITY_SELECTOR:NXvelocity_selector →
  _NXvelocity_selector.NX_tree_path NEXUSTREEPATH
  _NXvelocity_selector.NX_id VELOCITY_SELECTOR
  _NXvelocity_selector.NX_scan_id SCANID
  _NXvelocity_selector.NX_diffrn_id DIFFRNID

_NXvelocity_selector.NX_entry_id ENTRYID
where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the NeXus class instance, ending with "/NXvelocity_selector__VELOCITY_SELECTOR" where VELOCITY_SELECTOR is the name of this NeXus class instance, typically "velocity_selector". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.

- height:NX_FLOAT=HEIGHT →
  _NXvelocity_selector.height HEIGHT
- length:NX_FLOAT=LENGTH →
  _NXvelocity_selector.length LENGTH
- num:NX_INT=NUM →
  _NXvelocity_selector.num NUM
- radius:NX_FLOAT=RADIUS →
  _NXvelocity_selector.radius RADIUS
- rotation_speed:NX_FLOAT=ROTATION_SPEED →
  _NXvelocity_selector.rotation_speed ROTATION_SPEED
- spwidth:NX_FLOAT=SPWIDTH →
  _NXvelocity_selector.spwidth SPWIDTH
- table:NX_FLOAT=TABLE →
  _NXvelocity_selector.table TABLE
- twist:NX_FLOAT=TWIST →
  _NXvelocity_selector.twist TWIST
- type:NX_CHAR=TYPE →
  _NXvelocity_selector.type TYPE
- wavelength:NX_FLOAT=WAVELENGTH →
  _NXvelocity_selector.wavelength WAVELENGTH
- wavelength_spread:NX_FLOAT=WAVELENGTH_SPREAD →
  _NXvelocity_selector.wavelength_spread WAVELENGTH_SPREAD
- width:NX_FLOAT=WIDTH →
  _NXvelocity_selector.width WIDTH
- geometry:NXgeometry →
  _NXvelocity_selector.NXgeometry_id geometry

6.45. NXxraylens

```
NXxraylens (base class, version 1.0)
  aperture:NX_FLOAT
  curvature:NX_FLOAT
  cylindrical:NX_BOOLEAN
  focus_type:NX_CHAR
  gas:NX_CHAR
  gas_pressure:NX_FLOAT
  lens_geometry:NX_CHAR
  lens_length:NX_FLOAT
  lens_material:NX_CHAR
  lens_thickness:NX_FLOAT
  number_of_lenses:NX_INT
  symmetric:NX_BOOLEAN
  cylinder_orientation:NXnote
```

- XRAYLENS:NXxraylens →
  _NXxraylens.NX_tree_path NEXUSTREEPATH
  _NXxraylens.NX_id XRAYLENS
  _NXxraylens.NX_scan_id SCANID
  _NXxraylens.NX_diffrn_id DIFFRNID
  _NXxraylens.NX_entry_id ENTRYID

  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the NeXus class instance, ending with "/NXxraylens__XRAYLENS" where XRAYLENS is the name of this NeXus class instance, typically "xraylens". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- aperture:NX_FLOAT=APERTURE →
  _NXxraylens.aperture APERTURE
- curvature:NX_FLOAT=CURVATURE →
  _NXxraylens.curvature CURVATURE
- cylindrical:NX_BOOLEAN=CYLINDRICAL →
  _NXxraylens.cylindrical CYLINDRICAL
- focus_type:NX_CHAR=FOCUS_TYPE →
  _NXxraylens.focus_type FOCUS_TYPE
- gas:NX_CHAR=GAS →
  _NXxraylens.gas GAS
- gas_pressure:NX_FLOAT=GAS_PRESSURE →
  _NXxraylens.gas_pressure GAS_PRESSURE
- lens_geometry:NX_CHAR=LENS_GEOMETRY →
  _NXxraylens.lens_geometry LENS_GEOMETRY
- lens_length:NX_FLOAT=LENS_LENGTH →
  _NXxraylens.lens_length LENS_LENGTH
- lens_material:NX_CHAR=LENS_MATERIAL →
  _NXxraylens.lens_material LENS_MATERIAL

- lens_thickness:NX_FLOAT=LENS_THICKNESS → _NXxraylens.lens_thickness LENS_THICKNESS
- number_of_lenses:NX_INT=NUMBER_OF_LENSES → _NXxraylens.number_of_lenses NUMBER_OF_LENSES
- symmetric:NX_BOOLEAN=SYMMETRIC → _NXxraylens.symmetric SYMMETRIC
- cylinder_orientation:NXnote → _NXxraylens.NXnote_id cylinder_orientation

7. Proposed Pixel Array Detector Application Definitions

The following has been derived from the current Dectris Eiger test data and presentations and the Dectris web site:
`https://www.dectris.com/nexus.html#main_head_navigation`

In this version we have noted the multi-NXDATA use of NXentry, and note the extensions to NXdectector by including what Dectris has called detectorSpecific:DetectorSpecific detectorSpecific:DECTRIS_dectector_specific as a recommended placeholders for such information in an update to the NXdetector base class. We have also called detectorModule_xxx:DetectorModule detectorModule_xxx:DECTRIS_dectector_module and detectorChip_xxx:DetectorChip detectorChip_xxx:DECTRIS_detector_chip. These changes in class names have no impact on the relevant HDF5 paths.

7.1. NXentry

The base class NXentry currently has the following structure:

```
NXentry (base class, version 1.0)
  @IDF_Version
  collection_description:NX_CHAR
  collection_identifier:NX_CHAR
  collection_time:NX_FLOAT
  definition:NX_CHAR
    @version
    @URL
  definition_local:NX_CHAR
    @version
    @URL
  duration:NX_INT
  end_time:NX_DATE_TIME
  entry_identifier:NX_CHAR
  experiment_description:NX_CHAR
  experiment_identifier:NX_CHAR
  pre_sample_flightpath:NX_FLOAT
  program_name:NX_CHAR
    @version
    @configuration
  revision:NX_CHAR
    @comment
  run_cycle:NX_CHAR
  start_time:NX_DATE_TIME
  title:NX_CHAR
  NXcharacterization
  NXdata
  NXinstrument
  NXmonitor
  experiment_documentation:NXnote
  notes:NXnote
  thumbnail:NXnote
    @mime_type
  NXprocess
  NXsample
  NXsubentry
  NXuser
```

There is no conflict between the Dectris proposal and this class. We recommend that NIAC formally adopt the use of multiple NXdata NeXus class instances in NXentry and note that in the NXentry base class definition, not just imply that possibility in the NXdata base class definition as is now the case, so that the Detris proposal

```
NXentry_pad (application definition, version 0.1) (overlays NXentry)
  data_000001:NXDATA
  data_000002:NXDATA
  ...
```

```
  data_nnnnnn:NXDATA
  instrument:NXinstrument
  ...
```

will not raise any questions. The CIF mapping of the Dectris proposal would then be:

- ENTRY:NXentry →
  _NXentry.NX_tree_path NEXUSTREEPATH
  _NXentry.NX_id ENTRY
  _NXentry.NX_scan_id SCANID
  _NXentry.NX_diffrn_id DIFFRNID
  _NXentry.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the NeXus class instance, ending with "/NXentry__ENTRY" where ENTRY is the name of this NeXus class instance, typically "entry". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- data_000001:NXDATA →
  _NXentry.NXDATA_id data_000001
- data_000002:NXDATA →
  _NXentry.NXDATA_id data_000002
  ...
- data_nnnnnn:NXDATA →
  _NXentry.NXDATA_id data_nnnnnn
- instrument:NXinstrument →
  _NXentry.NXinstrument_id instrument

7.2. NXinstrument

There is no conflict between the Dectris proposal and this class. The Dectris use is

```
NXinstrument
  dectector:NXdetector
```

which is a standard use and would map in CIF to

- INSTRUMENT:NXinstrument →
  _NXinstrument.NX_tree_path NEXUSTREEPATH
  _NXinstrument.NX_id INSTRUMENT
  _NXinstrument.NX_scan_id SCANID
  _NXinstrument.NX_diffrn_id DIFFRNID
  _NXinstrument.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the NeXus class instance, ending with "/NXinstrument__INSTRUMENT" where INSTRUMENT is the name of this NeXus class instance, typically "instrument". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- dectector:NXdetector →
  _NXinstrument.NXdetector_id dectector

7.3. NXdetector

There are minor conflicts between the Dectris proposal and this class. Dectris has used the detector_number field as a character string, rather than as a number. We have added the "Dectris_" prefix for the character string version and changes the name to "Dectris_detector_serial_number". The use of character string serial "numbers" is a common practice for electronic equipment, so we recommend that a "detector_serial_number:NX_CHAR" field be added to the NXdetector base class to carry this information. The field detector_distance is not currently in this class. We have added the "Dectris_" prefix as a placeholder until a field with this information is added to the base class. Mark Koennecke has suggested "sample_detector_distance" as the new field name, so the name used here is "Dectris_sample_detector_distance"

Dectris has added one subclass, which is likely to be needed for almost all pixel array detectors. However, at the moment, what is proposed is a Dectris-specific class definition, so rather than proposing the subclass immediately as a new NeXus base class, we propose the subclass with a "DECTRIS_" prefix. The revised NXdetector base class would then be as follows. The fields explicitly used by DECTRIS are noted with [**]. Fields proposed by Dectris that are not currently in the NXdetector base class are prefixed with the "DECTRIS_" prefix

```
NXdetector (base class, version 1.1)
  acquisition_mode:NX_CHAR                          [**]
  angular_calibration:NX_FLOAT[i,j]                 [**]
  angular_calibration_applied:NX_BOOLEAN            [**]
  azimuthal_angle:NX_FLOAT[np,i,j]
  beam_center_x:NX_FLOAT                            [**]
  beam_center_y:NX_FLOAT                            [**]
  bit_depth_readout:NX_UINT                         [**]
  calibration_date:NX_DATE_TIME
  countrate_correction__applied:NX_BOOLEAN          [**]
  count_time:NX_NUMBER[np]                          [**]
  crate:NX_INT[i,j]
    @local_name
  data:NX_NUMBER[np,i,j,tof]
    @signal
    @axes
    @long_name
    @check_sum
```

```
  @link
data_error:NX_NUMBER[np,i,j,tof]
  @units
  @link
dead_time:NX_FLOAT[np,i,j]
description:NX_CHAR                                    [**]
DECTRIS_sample_detector_distance:NX_FLOAT              [**]
detection_gas_path:NX_FLOAT
detector_number:NX_INT[i,j]
DECTRIS_detector_serial_number:NX_CHAR                 [**]
detector_readout_time:NX_FLOAT                         [**]
DECTRIS_efficiency_correction_applied:NX_BOOLEAN       [**]
diameter:NX_FLOAT
distance:NX_FLOAT[np,i,j]
flatfield:NX_FLOAT[i,j]
flatfield_applied:NX_BOOLEAN                           [**]
flatfield_error:NX_FLOAT[i,j]
frame_start_number:NX_INT
frame_time:NX_FLOAT[NP]                                [**]
gain_setting:NX_CHAR                                   [**]
gas_pressure:NX_FLOAT[i,j]
input:NX_INT[i,j]
  @local_name
layout:NX_CHAR
local_name:NX_CHAR
number_of_cycles:NX_INT                                [**]
pixel_mask:NX_FLOAT[i,j]
pixel_mask_applied:NX_BOOLEAN                          [**]
polar_angle:NX_FLOAT[np,i,j]
raw_time_of_flight:NX_INT[tof+1]
  @frequency
saturation_value:NX_INT
sensor_material:NX_CHAR                                [**]
sensor_thickness:NX_FLOAT                              [**]
sequence_number:NX_CHAR
slot:NX_INT[i,j]
  @local_name
solid_angle:NX_FLOAT[i,j]
threshold_energy:NX_FLOAT                              [**]
time_of_flight:NX_FLOAT[tof+1]
  @axis
  @primary
  @long_name
  @link
trigger_dead_time:NX_FLOAT
trigger_delay_time:NX_FLOAT
type:NX_CHAR
DECTRIS_virtual_pixel_correction_applied:NX_BOOLEAN    [**]
x_pixel_offset:NX_FLOAT[i,j]
  @axis
  @primary
  @long_name
  @link
x_pixel_size:NX_FLOAT[i,j]                             [**]
y_pixel_offset:NX_FLOAT[i,j]
  @axis
  @primary
  @long_name
y_pixel_size:NX_FLOAT[i,j]                             [**]
characterization:NXcharacterization
detectorSpecific:DECTRIS\_detector\_specific           [**]
efficiency:NXdata
  efficiency:NX_FLOAT[i,j,k]
  real_time:NX_NUMBER[i,j,k]
  wavelength:NX_FLOAT[i,j,k]
geometry:NXgeometry
calibration_method:NXnote
data_file:NXnote
```

adding

```
DECTRIS_sample_detector_distance:NX_FLOAT
DECTRIS_detector_serial_number:NX_CHAR
DECTRIS_efficiency_correction_applied:NX_BOOLEAN
DECTRIS_virtual_pixel_correction_applied:NX_BOOLEAN
```

and giving explicit suggested names to the NXcharacterization and NXgeometry instances.

```
NXdetector_detris (application definition, version 0.1)
  (overlays NXDetector)
NXdetector
  acquisition_mode:NX_CHAR
  angular_calibration_applied:NX_BOOLEAN
  beam_center_x:NX_FLOAT
    @units
  beam_center_y:NX_FLOAT
    @units
  bit_depth_readout:NX_UINT
  count_time:NX_FLOAT[np]
    @units
  countrate_correction_applied:NX_BOOLEAN
  description:NX_CHAR
  DECTRIS_sample_detector_distance:NX_FLOAT
    @units
  DECTRIS_detector_serial_number:NX_CHAR
  detector_number:NX_CHAR
  detector_readout_time:NX_FLOAT[np]
    @units
  detectorSpecific:DECTRIS_detector_specific
  DECTRIS_efficiency_correction_applied:NX_BOOLEAN
  flatfield_correction_applied:NX_BOOL
  frame_time:NX_FLOAT[np]
    @units
  gain_setting:NX_CHAR
  number_of_cycles:NX_UINT
  pixel_mask_applied:NX_BOOL
  sensor_material:NX_STRING
  sensor_thickness:NX_FLOAT
    @units
  threshold_energy:NX_FLOAT
    @units
  DECTRIS_virtual_pixel_correction_applied:NX_BOOL
  x_pixel_size:NX_FLOAT
    @units
  y_pixel_size:NX_FLOAT
    @units
```

- DETECTOR:NXdetector →
  _NXdetector.NX_tree_path NEXUSTREEPATH
  _NXdetector.NX_id DETECTOR
  _NXdetector.NX_scan_id SCANID
  _NXdetector.NX_diffrn_id DIFFRNID
  _NXdetector.NX_entry_id ENTRYID

  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the NeXus class instance, ending with "/NXdetector__DETECTOR" where DETECTOR is the name of this NeXus class instance, typically "detector". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- acquisition_mode:NX_CHAR=ACQUISITION_MODE →
  _NXdetector.acquisition_mode ACQUISITION_MODE
- angular_calibration_applied:NX_BOOLEAN=ANGULAR_CALIBRATION_APPLIED →
  _NXdetector.angular_calibration_applied ANGULAR_CALIBRATION_APPLIED
- beam_center_x:NX_FLOAT=BEAM_CENTER_X →
  _NXdetector.beam_center_x BEAM_CENTER_X
- @units=UNITS →
  _NXdetector.beam_center_x__units UNITS
- beam_center_y:NX_FLOAT=BEAM_CENTER_Y →
  _NXdetector.beam_center_y BEAM_CENTER_Y
- @units=UNITS →
  _NXdetector.beam_center_y__units UNITS
- bit_depth_readout:NX_UINT=BIT_DEPTH_READOUT →
  _NXdetector.bit_depth_readout BIT_DEPTH_READOUT
- count_time:NX_FLOAT[np]=COUNT_TIME →
  _NXdetector.count_time COUNT_TIME
- @units=UNITS →
  _NXdetector.count_time__units UNITS
- countrate_correction_applied:NX_BOOLEAN=COUNTRATE_CORRECTION_APPLIED →
  _NXdetector.countrate_correction_applied COUNTRATE_CORRECTION_APPLIED
- description:NX_CHAR=DESCRIPTION →
  _NXdetector.description DESCRIPTION
- detector_number:NX_CHAR=DETECTOR_NUMBER →
  _NXdetector.detector_number DETECTOR_NUMBER
- detectorSpecific:DECTRIS_detector_specific →
  _NXdetector.DECTRIS_detector_specific_id detectorSpecific
- DECTRIS_detector_number:NX_CHAR=DETECTOR_NUMBER →
  _NXdetector.DECTRIS_detector_number DETECTOR_NUMBER
- detector_readout_time:NX_FLOAT[np]=DETECTOR_READOUT_TIME →

_NXdetector.detector_readout_time DETECTOR_READOUT_TIME
- @units=UNITS →
  _NXdetector.detector_readout_time__units UNITS
- efficiency_correction_applied:NX_BOOL=EFFICIENCY_CORRECTION_APPLIED →
  _NXdetector.efficiency_correction_applied EFFICIENCY_CORRECTION_APPLIED
- flatfield_correction_applied:NX_BOOL=FLATFIELD_CORRECTION_APPLIED →
  _NXdetector.flatfield_correction_applied FLATFIELD_CORRECTION_APPLIED
- frame_time:NX_FLOAT[np]=FRAME_TIME →
  _NXdetector.frame_time FRAME_TIME
- @units=UNITS →
  _NXdetector.frame_time__units UNITS
- gain_setting:NX_CHAR=GAIN_SETTING →
  _NXdetector.gain_setting GAIN_SETTING
- number_of_cycles:NX_UINT=NUMBER_OF_CYCLES →
  _NXdetector.number_of_cycles NUMBER_OF_CYCLES
- pixel_mask_applied:NX_BOOL=PIXEL_MASK_APPLIED →
  _NXdetector.pixel_mask_applied PIXEL_MASK_APPLIED
- sensor_material:NX_STRING=SENSOR_MATERIAL →
  _NXdetector.sensor_material SENSOR_MATERIAL
- sensor_thickness:NX_FLOAT=SENSOR_THICKNESS →
  _NXdetector.sensor_thickness SENSOR_THICKNESS
- @units=UNITS →
  _NXdetector.sensor_thickness__units UNITS
- threshold_energy:NX_FLOAT=THRESHOLD_ENERGY →
  _NXdetector.threshold_energy THRESHOLD_ENERGY
- @units=UNITS →
  _NXdetector.threshold_energy__units UNITS
- DECTRIS_virtual_pixel_correction_applied:NX_BOOL=VIRTUAL_PIXEL_CORRECTION_APPLIED →
  _NXdetector.DECTRIS_virtual_pixel_correction_applied VIRTUAL_PIXEL_CORRECTION_APPLIED
- x_pixel_size:NX_FLOAT=X_PIXEL_SIZE →
  _NXdetector.x_pixel_size X_PIXEL_SIZE
- @units=UNITS →
  _NXdetector.x_pixel_size__units UNITS
- y_pixel_size:NX_FLOAT=Y_PIXEL_SIZE →
  _NXdetector.y_pixel_size Y_PIXEL_SIZE
- @units=UNITS →
  _NXdetector.y_pixel_size__units UNITS

7.4. detectorSpecific:DECTRIS_detector_specific

These are some additions added by Dectris, that they designated as a new detectorSpecific:DectectorSpecific NeXus class instance.

```
DECTRIS\_detector\_specific (application definition, version 0.1)
  countrate_correction_bunch_mode:NX_CHAR
  countrate_correction_count_cutoff:NX_UINT
  countrate_correction_lookup_table:NX_FLOAT[1000000]
  data_collection_date:NX_CHAR
  detectorModule_000:DECTRIS_detector_module
  detectorModule_001:DECTRIS_detector_module
  detectorModule_002:DECTRIS_detector_module
  detectorModule_003:DECTRIS_detector_module
  detectorModule_004:DECTRIS_detector_module
  detectorModule_005:DECTRIS_detector_module
  detectorModule_006:DECTRIS_detector_module
  detectorModule_007:DECTRIS_detector_module
  detectorModule_008:DECTRIS_detector_module
  detectorModule_009:DECTRIS_detector_module
  detectorModule_010:DECTRIS_detector_module
  detectorModule_011:DECTRIS_detector_module
  detectorModule_012:DECTRIS_detector_module
  detectorModule_013:DECTRIS_detector_module
  detectorModule_014:DECTRIS_detector_module
  detectorModule_015:DECTRIS_detector_module
  detectorModule_016:DECTRIS_detector_module
  detectorModule_017:DECTRIS_detector_module
  detectorModule_018:DECTRIS_detector_module
  detectorModule_019:DECTRIS_detector_module
  detectorModule_020:DECTRIS_detector_module
  detectorModule_021:DECTRIS_detector_module
  detectorModule_022:DECTRIS_detector_module
  detectorModule_023:DECTRIS_detector_module
  detectorModule_024:DECTRIS_detector_module
  detectorModule_025:DECTRIS_detector_module
  detectorModule_026:DECTRIS_detector_module
  detectorModule_027:DECTRIS_detector_module
  detectorModule_028:DECTRIS_detector_module
  detectorModule_029:DECTRIS_detector_module
```

```
detectorModule_030:DECTRIS_detector_module
detectorModule_031:DECTRIS_detector_module
detectorModule_032:DECTRIS_detector_module
detectorModule_033:DECTRIS_detector_module
detectorModule_034:DECTRIS_detector_module
detectorModule_035:DECTRIS_detector_module
detectorModule_036:DECTRIS_detector_module
detectorModule_037:DECTRIS_detector_module
detectorModule_038:DECTRIS_detector_module
detectorModule_039:DECTRIS_detector_module
detectorModule_040:DECTRIS_detector_module
detectorModule_041:DECTRIS_detector_module
detectorModule_042:DECTRIS_detector_module
detectorModule_043:DECTRIS_detector_module
detectorModule_044:DECTRIS_detector_module
detectorModule_045:DECTRIS_detector_module
detectorModule_046:DECTRIS_detector_module
detectorModule_047:DECTRIS_detector_module
detectorModule_048:DECTRIS_detector_module
detectorModule_049:DECTRIS_detector_module
detectorModule_050:DECTRIS_detector_module
detectorModule_051:DECTRIS_detector_module
detectorModule_052:DECTRIS_detector_module
detectorModule_053:DECTRIS_detector_module
detectorModule_054:DECTRIS_detector_module
detectorModule_055:DECTRIS_detector_module
detectorModule_056:DECTRIS_detector_module
detectorModule_057:DECTRIS_detector_module
detectorModule_058:DECTRIS_detector_module
detectorModule_059:DECTRIS_detector_module
detector_origin:NX_FLOAT
    @depends_on
    @transformation
    @units
    @vector
transformLabToDetector:NX_FLOAT
    @rotation:NX_FLOAT[9]
    @translation:NX_FLOAT[3]
    @units
flat field:NX_FLOAT[number of x pixels,number of y pixels]
mode_register:NX_UINT
nimages:NX_UINT
number_of_excluded_pixels:NX_UINT
photon_energy:NX_FLOAT
pixel_mask:NX_UINT[number of x pixels,number of y pixels]
readout_mode:NX_CHAR
software_version:NX_CHAR
sub_image_exposure_time:NX_FLOAT
summation_mode:NX_CHAR
summation_nimages:NX_UINT
trigger_mode:NX_CHAR
x_pixels_in_detector:NX_UINT
y_pixels_in_detector:NX_UINT
```

- DETECTOR_SPECIFIC:DECTRIS_detector_specific →
  _DECTRIS_detector_specific.NX_tree_path NEXUSTREEPATH
  _DECTRIS_detector_specific.NX_id DETECTOR_SPECIFIC
  _DECTRIS_detector_specific.NX_scan_id SCANID
  _DECTRIS_detector_specific.NX_diffrn_id DIFFRNID
  _DECTRIS_detector_specific.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the NeXus class instance, ending with "/NXcollection- __DETECTOR_SPECIFIC" where _DETECTOR_SPECIFIC is the name of this NeXus class instance, typically "collection". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- countrate_correction_bunch_mode:NX_CHAR=COUNTRATE_CORRECTION_BUNCH_MODE →
  _DECTRIS_detector_specific.countrate_correction_bunch_mode
  COUNTRATE_CORRECTION_BUNCH_MODE
- countrate_correction_count_cutoff:NX_UINT=
  COUNTRATE_CORRECTION_COUNT_CUTOFF →
  _DECTRIS_detector_specific.countrate_correction_count_cutoff
  COUNTRATE_CORRECTION_COUNT_CUTOFF
- countrate_correction_lookup_table:NX_FLOAT[1000000]
  =COUNTRATE_CORRECTION_LOOKUP_TABLE →
  _DECTRIS_detector_specific.countrate_correction_lookup_table
  COUNTRATE_CORRECTION_LOOKUP_TABLE
- data_collection_date:NX_CHAR=DATA_COLLECTION_DATE →
  _DECTRIS_detector_specific.data_collection_date DATA_COLLECTION_DATE

- detectorModule_000:DECTRIS_detector_module
  detectorModule_001:DECTRIS_detector_module
  ...
  detectorModule_059:DECTRIS_detector_module →
  _DECTRIS_detector_specific.DECTRIS_detector_module_id
  ["detectorModule_000","detectorModule_001",...,"detectorModule_059"]
- detector_origin:NX_FLOAT64=DETECTOR_ORIGIN →
  _DECTRIS_detector_specific.detector_origin DETECTOR_ORIGIN
- @depends_on=DEPENDS_ON →
  _DECTRIS_detector_specific.detector_origin__depends_on DEPENDS_ON
- @transformation=TRANSFORMATION →
  _DECTRIS_detector_specific.detector_origin__transformation TRANSFORMATION
- @units=UNITS →
  _DECTRIS_detector_specific.detector_origin__units UNITS
- @vector=VECTOR →
  _DECTRIS_detector_specific.detector_origin__vector VECTOR
- transformLabToDetector:NX_FLOAT64=TRANSFORMLABTODETECTOR →
  _DECTRIS_detector_specific.transformLabToDetector TRANSFORMLABTODETECTOR
- @rotation=ROTATION →
  _DECTRIS_detector_specific.transformLabToDetector__rotation ROTATION
- @translation=TRANSLATION →
  _DECTRIS_detector_specific.transformLabToDetector__translation TRANSLATION
- @units=UNITS →
  _DECTRIS_detector_specific.transformLabToDetector__units UNITS
- flatfield:NX_FLOAT[numberofxpixels,numberofypixels]=FLATFIELD →
  _DECTRIS_detector_specific.flatfield FLATFIELD
- mode_register:NX_UINT=MODE_REGISTER →
  _DECTRIS_detector_specific.mode_register MODE_REGISTER
- nimages:NX_UINT=NIMAGES →
  _DECTRIS_detector_specific.nimages NIMAGES
- number_of_excluded_pixels:NX_UINT=NUMBER_OF_EXCLUDED_PIXELS →
  _DECTRIS_detector_specific.number_of_excluded_pixels NUMBER_OF_EXCLUDED_PIXELS
- photon_energy:NX_FLOAT=PHOTON_ENERGY →
  _DECTRIS_detector_specific.photon_energy PHOTON_ENERGY
- pixel_mask:NX_UINT[numberofxpixels,numberofypixels]=PIXEL_MASK →
  _DECTRIS_detector_specific.pixel_mask PIXEL_MASK
- readout_mode:NX_CHAR=READOUT_MODE →
  _DECTRIS_detector_specific.readout_mode READOUT_MODE
- software_version:NX_CHAR=SOFTWARE_VERSION →
  _DECTRIS_detector_specific.software_version SOFTWARE_VERSION
- sub_image_exposure_time:NX_FLOAT=SUB_IMAGE_EXPOSURE_TIME →
  _DECTRIS_detector_specific.sub_image_exposure_time SUB_IMAGE_EXPOSURE_TIME
- summation_mode:NX_CHAR=SUMMATION_MODE →
  _DECTRIS_detector_specific.summation_mode SUMMATION_MODE
- summation_nimages:NX_UINT=SUMMATION_NIMAGES →
  _DECTRIS_detector_specific.summation_nimages SUMMATION_NIMAGES
- trigger_mode:NX_CHAR=TRIGGER_MODE →
  _DECTRIS_detector_specific.trigger_mode TRIGGER_MODE
- x_pixels_in_detector:NX_UINT=X_PIXELS_IN_DETECTOR →
  _DECTRIS_detector_specific.x_pixels_in_detector X_PIXELS_IN_DETECTOR
- y_pixels_in_detector:NX_UINT=Y_PIXELS_IN_DETECTOR →
  _DECTRIS_detector_specific.y_pixels_in_detector Y_PIXELS_IN_DETECTOR

The specific number of modules may vary.

7.5. detectorModule_nnn:DECTRIS_detector_module

```
DECTRIS\_detector\_module (application definition, version 0.1)
  dac_names:NX_CHAR[6]
  dac_values:NX_UINT[6]
  data_origin:NX_UINT[2]
  data_size:NX_UINT[2]
  detectorChip_000:DECTRIS_detector_chip
  detectorChip_001:DECTRIS_detector_chip
  detectorChip_002:DECTRIS_detector_chip
  detectorChip_003:DECTRIS_detector_chip
  detectorChip_004:DECTRIS_detector_chip
  detectorChip_005:DECTRIS_detector_chip
  detectorChip_006:DECTRIS_detector_chip
  detectorChip_007:DECTRIS_detector_chip
  detectorChip_008:DECTRIS_detector_chip
  detectorChip_009:DECTRIS_detector_chip
  detectorChip_010:DECTRIS_detector_chip
  detectorChip_011:DECTRIS_detector_chip
  detectorChip_012:DECTRIS_detector_chip
  detectorChip_013:DECTRIS_detector_chip
  detectorChip_014:DECTRIS_detector_chip
  detectorChip_015:DECTRIS_detector_chip
  fast_pixel_direction:NX_FLOAT64
```

```
  @depends_on
  @transformation
  @units
  @vector
firmware_version:NX_CHAR
module_offset:NX_FLOAT64
  @depends_on
  @transformation
  @units
  @vector
nbits:NX_UINT
nchips:NX_UINT
readout_frequency:NX_FLOAT
  @units
region_of_interest:NX_UINT[4]
slow_pixel_direction:NX_FLOAT64
  @depends_on
  @transformation
  @units
  @vector
x_pixels_in_module:NX_UINT
y_pixels_in_module:NX_UINT
```

- DETECTORMODULE:DECTRIS_detector_module →
  _DECTRIS_detector_module.NX_tree_path NEXUSTREEPATH
  _DECTRIS_detector_module.NX_id DETECTORMODULE
  _DECTRIS_detector_module.NX_scan_id SCANID
  _DECTRIS_detector_module.NX_diffrn_id DIFFRNID
  _DECTRIS_detector_module.NX_entry_id ENTRYID
  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the NeXus class instance, ending with "/DECTRIS_detector_module- _DETECTORMODULE" where DETECTORMODULE is the name of this NeXus class instance, typically "detectorModule_nnn". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- dac_names:NX_CHAR[6]=DAC_NAMES →
  _DECTRIS_detector_module.dac_names DAC_NAMES
- dac_values:NX_UINT[7]=DAC_VALUES →
  _DECTRIS_detector_module.dac_values DAC_VALUES
- data_origin:NX_UINT[2]=DATA_ORIGIN →
  _DECTRIS_detector_module.data_origin DATA_ORIGIN
- data_size:NX_UINT[2]=DATA_SIZE →
  _DECTRIS_detector_module.data_size DATA_SIZE
- detectorChip_000:DECTRIS_detector_chip
  detectorChip_001:DECTRIS_detector_chip
  ...
  detectorChip_015:DECTRIS_detector_chip →
  _DECTRIS_detector_module.NXcollection_id
  ["detectorChip_000","detectorChip_001",...,"detectorChip_015"]
- fast_pixel_direction:NX_FLOAT64=FAST_PIXEL_DIRECTION →
  _DECTRIS_detector_module.fast_pixel_direction FAST_PIXEL_DIRECTION
- @depends_on=DEPENDS_ON →
  _DECTRIS_detector_module.fast_pixel_direction__depends_on DEPENDS_ON
- @transformation=TRANSFORMATION →
  _DECTRIS_detector_module.fast_pixel_direction__transformation TRANSFORMATION
- @units=UNITS →
  _DECTRIS_detector_module.fast_pixel_direction__units UNITS
- @vector=VECTOR →
  _DECTRIS_detector_module.fast_pixel_direction__vector VECTOR
- firmware_version:NX_CHAR=FIRMWARE_VERSION →
  _DECTRIS_detector_module.firmware_version FIRMWARE_VERSION
- module_offset:NX_FLOAT64=MODULE_OFFSET →
  _DECTRIS_detector_module.module_offset MODULE_OFFSET
- @depends_on=DEPENDS_ON →
  _DECTRIS_detector_module.module_offset__depends_on DEPENDS_ON
- @transformation=TRANSFORMATION →
  _DECTRIS_detector_module.module_offset__transformation TRANSFORMATION
- @units=UNITS →
  _DECTRIS_detector_module.module_offset__units UNITS
- @vector=VECTOR →
  _DECTRIS_detector_module.module_offset__vector VECTOR
- nbits:NX_UINT=NBITS →
  _DECTRIS_detector_module.nbits NBITS
- nchips:NX_UINT=NCHIPS →
  _DECTRIS_detector_module.nchips NCHIPS
- readout_frequency:NX_FLOAT=READOUT_FREQUENCY →
  _DECTRIS_detector_module.readout_frequency READOUT_FREQUENCY
- @units=UNITS →
  _DECTRIS_detector_module.readout_frequency__units UNITS

- region_of_interest:NX_UINT[4]=REGION_OF_INTEREST →
  _DECTRIS_detector_module.region_of_interest REGION_OF_INTEREST
- slow_pixel_direction:NX_FLOAT64=SLOW_PIXEL_DIRECTION →
  _DECTRIS_detector_module.slow_pixel_direction SLOW_PIXEL_DIRECTION
- @depends_on=DEPENDS_ON →
  _DECTRIS_detector_module.slow_pixel_direction__depends_on DEPENDS_ON
- @transformation=TRANSFORMATION →
  _DECTRIS_detector_module.slow_pixel_direction__transformation TRANSFORMATION
- @units=UNITS →
  _DECTRIS_detector_module.slow_pixel_direction__units UNITS
- @vector=VECTOR →
  _DECTRIS_detector_module.slow_pixel_direction__vector VECTOR
- x_pixels_in_module:NX_UINT=X_PIXELS_IN_MODULE →
  _DECTRIS_detector_module.x_pixels_in_module X_PIXELS_IN_MODULE
- y_pixels_in_module:NX_UINT=Y_PIXELS_IN_MODULE →
  _DECTRIS_detector_module.y_pixels_in_module Y_PIXELS_IN_MODULE

The specific number of chips may vary.

7.6. detectorChip_nn:DECTRIS_detector_chip

```
DECTRIS_detector_chip
  chip_type:NX_CHAR
  x_pixels_in_chip:NX_UINT
  x_position:NX_UINT
  y_pixels_in_chip:NX_UINT
  y_position:NX_UINT
```

- DETECTORCHIP:DECTRIS_detector_chip →
  _DECTRIS_detector_chip.NX_tree_path NEXUSTREEPATH
  _DECTRIS_detector_chip.NX_id DETECTORCHIP
  _DECTRIS_detector_chip.NX_scan_id SCANID
  _DECTRIS_detector_chip.NX_diffrn_id DIFFRNID
  _DECTRIS_detector_chip.NX_entry_id ENTRYID

  where components of NEXUSTREEPATH are composed of the relevant NeXus class, a double under score and, finally, the name of the NeXus class instance, ending with "/DECTRIS_detector_chip-__DETECTORCHIP" where DETECTORCHIP is the name of this NeXus class instance, typically "detectorChip_nnn". The SCANID, DIFFRNID and ENTRYID are optional keys for use when multiple scans, *etc.* are aggregated in the same CBF.
- chip_type:NX_CHAR=CHIP_TYPE →
  _NXcollection.chip_type CHIP_TYPE
- x_pixels_in_chip:NX_UINT=X_PIXELS_IN_CHIP →
  _NXcollection.x_pixels_in_chip X_PIXELS_IN_CHIP
- x_position:NX_UINT=X_POSITION →
  _NXcollection.x_position X_POSITION
- y_pixels_in_chip:NX_UINT=Y_PIXELS_IN_CHIP →
  _NXcollection.y_pixels_in_chip Y_PIXELS_IN_CHIP
- y_position:NX_UINT=Y_POSITION →
  _NXcollection.y_position Y_POSITION

7.7. Consolidated Dectris Eiger Application Definition

In addition to the above classes, the Dectris application definition draws on NXsample for the rotation_angle_step and NXmonochromator for the wavelength. The Dectris Eiger NeXus format as proposed by Dectris with the proposed changes in this document marked with "**[...]**" is as follows. However, rotation_angle_step is not in the NeXus base class dictionary. Therefore, we flag it with the DETRIS prefix until it is formally adopted as part of the NXsample base class.

```
NXdectris_eiger (application definition, version 0.1)
  (overlays NXentry)
  entry:NXentry
    data_000001:NXDATA
      data:NXINT[np_000001,number of x pixels,number of y pixels]
        @image_nr_low
        @image_nr_high
    data_000002:NXDATA
      data:NXINT[np_000001,number of x pixels,number of y pixels]
        @image_nr_low
        @image_nr_high
    ...
    data_nnnnnn:NXDATA
      data:NXINT[np_nnnnnn,number of x pixels,number of y pixels]
        @image_nr_low
        @image_nr_high
    instrument:NXinstrument
      detector:NXdetector
        acquisition_mode:NX_CHAR
        angular_calibration_applied:NX_FLOAT32[number of x pixels,number of y pixels]
        beam_center_x:NX_FLOAT
          @units
        beam_center_y:NX_FLOAT
          @units
```

```
bit_depth_readout:NX_UINT
countrate_correction_applied:NX_BOOLEAN
count_time:NX_FLOAT[np]
  @units
description:NX_CHAR
detector_number:NX_CHAR
detectorSpecific:DetectorSpecific           **[detectorSpecific:DECTRIS_detector_specific]**
  countrate_correction_bunch_mode:NX_CHAR
  countrate_correction_count_cutoff:NX_UINT
  countrate_correction_lookup_table:NX_FLOAT[1000000]
  data_collection_date:NX_CHAR
  detectorModule_000:detectorModule         **[detectorModule_000:DECTRIS_detector_module]**
    dac_names:NX_CHAR[6]
    dac_values:NX_UINT[7]
    data_origin:NX_UINT[2]
    data_size:NX_UINT[2]
    detectorChip_000:DetectorChip            **[detectorChip_000:DECTRIS_detector_chip]**
      chip_type:NX_CHAR
      x_pixels_in_chip:NX_UINT
      x_position:NX_UINT
      y_pixels_in_chip:NX_UINT
      y_position:NX_UINT
    detectorChip_001:DetectorChip            **[detectorChip_001:DECTRIS_detector_chip]**
      chip_type:NX_CHAR
      x_pixels_in_chip:NX_UINT
      x_position:NX_UINT
      y_pixels_in_chip:NX_UINT
      y_position:NX_UINT
    ...
    detectorChip_015:DetectorChip            **[detectorChip_015:DECTRIS_detector_chip]**
      chip_type:NX_CHAR
      x_pixels_in_chip:NX_UINT
      x_position:NX_UINT
      y_pixels_in_chip:NX_UINT
      y_position:NX_UINT
    fast_pixel_direction:NX_FLOAT
      @transformation
      @vector:NX_FLOAT[3]
      @units
      @depends_on
    firmware_version:NX_CHAR
    module_offset:NX_FLOAT
      @transformation
      @vector:NX_FLOAT[3]
      @units
      @depends_on
    nbits:NX_UINT
    nchips:NX_UINT
    readout_frequency:NX_FLOAT
      @units
    region_of_interest:NX_UINT[4]
    slow_pixel_direction:NX_FLOAT
      @transformation
      @vector:NX_FLOAT[3]
      @units
      @depends_on
    x_pixels_in_module:NX_UINT
    y_pixels_in_module:NX_UINT
  detectorModule_001:detectorModule         **[detectorModule_001:DECTRIS_detector_module]**
    dac_names:NX_CHAR[6]
    dac_values:NX_UINT[7]
    data_origin:NX_UINT[2]
    data_size:NX_UINT[2]
    detectorChip_000:DetectorChip            **[detectorChip_000:DECTRIS_detector_chip]**
      chip_type:NX_CHAR
      x_pixels_in_chip:NX_UINT
      x_position:NX_UINT
      y_pixels_in_chip:NX_UINT
      y_position:NX_UINT
    detectorChip_001:DetectorChip            **[detectorChip_001:DECTRIS_detector_chip]**
      chip_type:NX_CHAR
      x_pixels_in_chip:NX_UINT
      x_position:NX_UINT
      y_pixels_in_chip:NX_UINT
      y_position:NX_UINT
    ...
    detectorChip_015:DetectorChip            **[detectorChip_015:DECTRIS_detector_chip]**
```

```
      chip_type:NX_CHAR
      x_pixels_in_chip:NX_UINT
      x_position:NX_UINT
      y_pixels_in_chip:NX_UINT
      y_position:NX_UINT
    fast_pixel_direction:NX_FLOAT
      @transformation
      @vector:NX_FLOAT[3]
      @units
      @depends_on
    firmware_version:NX_CHAR
    module_offset:NX_FLOAT
      @transformation
      @vector:NX_FLOAT[3]
      @units
      @depends_on
    nbits:NX_UINT
    nchips:NX_UINT
    readout_frequency:NX_FLOAT
      @units
    region_of_interest:NX_UINT[4]
    slow_pixel_direction:NX_FLOAT
      @transformation
      @vector:NX_FLOAT[3]
      @units
      @depends_on
    x_pixels_in_module:NX_UINT
    y_pixels_in_module:NX_UINT
  ...
  detectorModule_059:detectorModule        **[detectorModule_059:DECTRIS_detector_module]**
    dac_names:NX_CHAR[6]
    dac_values:NX_UINT[7]
    data_origin:NX_UINT[2]
    data_size:NX_UINT[2]
    detectorChip_000:DetectorChip           **[detectorChip_000:DECTRIS_detector_chip]**
      chip_type:NX_CHAR
      x_pixels_in_chip:NX_UINT
      x_position:NX_UINT
      y_pixels_in_chip:NX_UINT
      y_position:NX_UINT
    detectorChip_001:DetectorChip           **[detectorChip_001:DECTRIS_detector_chip]**
      chip_type:NX_CHAR
      x_pixels_in_chip:NX_UINT
      x_position:NX_UINT
      y_pixels_in_chip:NX_UINT
      y_position:NX_UINT
    ...
    detectorChip_015:DetectorChip           **[detectorChip_015:DECTRIS_detector_chip]**
      chip_type:NX_CHAR
      x_pixels_in_chip:NX_UINT
      x_position:NX_UINT
      y_pixels_in_chip:NX_UINT
      y_position:NX_UINT
    fast_pixel_direction:NX_FLOAT
      @transformation
      @vector:NX_FLOAT[3]
      @units
      @depends_on
    firmware_version:NX_CHAR
    module_offset:NX_FLOAT
      @transformation
      @vector:NX_FLOAT[3]
      @units
      @depends_on
    nbits:NX_UINT
    nchips:NX_UINT
    readout_frequency:NX_FLOAT
      @units
    region_of_interest:NX_UINT[4]
    slow_pixel_direction:NX_FLOAT
      @transformation
      @vector:NX_FLOAT[3]
      @units
      @depends_on
    x_pixels_in_module:NX_UINT
    y_pixels_in_module:NX_UINT
  detector_origin:NX_FLOAT
```

```
        @transformation
        @vector:NX_FLOAT[3]
        @units
        @depends_on
      transformLabToDetector:NX_FLOAT
        @rotation:NX_FLOAT[9]
        @translation:NX_FLOAT[3]
        @units
      flatfield:NX_FLOAT[number of x pixels,number of y pixels]
      mode_register:NX_UINT
      nimages:NX_UINT
      number_of_excluded_pixels:NX_UINT
      photon_energy:NX_FLOAT
      pixel_mask:NX_UINT[number of x pixels,number of y pixels]
      readout_mode:NX_CHAR
      software_version:NX_CHAR
      sub_image_exposure_time:NX_FLOAT
      summation_mode:NX_CHAR
      summation_nimages:NX_UINT
      trigger_mode:NX_CHAR
      x_pixels_in_detector:NX_UINT
      y_pixels_in_detector:NX_UINT
    detector_number:NX_CHAR
    detector_readout_time:NX_FLOAT[np]
      @units
    efficiency_correction_applied:NX_BOOL
    flatfield_correction_applied:NX_BOOL
    frame_time:NX_FLOAT[np]
      @units
    gain_setting:NX_CHAR
    number_of_cycles:NX_UINT
    pixel_mask_applied:NX_BOOL
    sensor_material:NX_STRING
    sensor_thickness:NX_FLOAT
      @units
    threshold_energy:NX_FLOAT
      @units
    virtual_pixel_correction_applied:NX_BOOLEAN
    x_pixel_size:NX_FLOAT
      @units
    y_pixel_size:NX_FLOAT
      @units
sample:NXsample
  rotation_angle_step:NX_FLOAT[np]                    **[DECTRIS_rotation_angle_step]**
    @units
monochromator:NXmonchromator
  wavelength:NX_FLOAT32
    @units
```