

Coping with BIG DATA Image Formats: Integration of CBF, NeXus and HDF5, A Progress Report

Herbert J. Bernstein,[†] Jonathan M. Sloan,^{*} Graeme Winter,^{*} Tobias S. Richter,^{*} NeXus International Advisory Committee,[‡] Committee on the Maintenance of the CIF Standard[§]



Abstract 1.2.4 1995 24 – 28 May 2014 at American Cryst. Assoc. meeting, Albuquerque, NM USA.
Work supported in part by NIGMS, DOE, NSF, PaNData ODI (EU 7th Framework Programme)

The BIG DATA demands of the new generation of X-ray pixel array detectors necessitate the use of new storage technologies as we meet the limitations of existing file systems. Fast Dectris Pilatus detectors and FEL detectors such as the Cornell-SLAC pixel array detector (CSPAD) are already straining file systems, and the new generation of even faster detectors, such as the Dectris Eiger shown in Fig. 1, will bring this issue to a head. In addition, the modular nature of these detectors provides the opportunity to construct more complex detector arrays (e.g. the Dectris Pilatus detector at I23 at DLS shown in Fig. 3), which in turn requires a more complete description of the detector geometry. Taken together these give rise to a need to combine the best issues of CBF/imgCIF (the Crystallographic Binary File, which has a complete description of the experiment), NeXus (a common data framework for neutron, X-ray and muon science, which gracefully handles large data sets) and HDF5 (Hierarchical Data Format, version 5, the high-performance data format used by NeXus) for the management of such data at synchrotrons. In July 2013, discussions were in progress between COMCIFS (the IUCr Committee for the Maintenance of the CIF Standard) and NIAC (the NeXus International Advisory Committee) on an integrated ontology. Those discussions have progressed. A proof-of-concept API based on CBFlib and the HDF5 API that was being developed in a collaboration among Dowling College, Brookhaven National Laboratory and Diamond Light Source is now in use. The mapping and combined API continue to develop [1]. Releases of CBFlib since CBFlib 0.9.2.12 can store arbitrary CBF files in HDF5 and recover them, support use of all CBFlib compressions in HDF5 files, and can convert sets of minibcf files to a single NeXus file. The latest release, CBFlib 0.9.5, is operational for HDF5 handling of single detector module, monochromatic MX data compatibly with imgCIF, and similar multiple detector module support in HDF5 should be operational in Fall 2014. Here we present the new format, with examples, alongside the implications of the use of this format for software developers and for beamline users.



Figure 1: Dectris Eiger 1M, an example of the new generation of very high data rate detectors that raise BIG DATA issues.

- The new generation of high performance x-ray detectors requires integration of HDF5, NeXus and CBF.
- The DECTRIS workshop in Baden, Switzerland in January 2013 established the parameters of the integration.
- NIAC and COMCIFS are working together to ensure interoperability.
- Use of compressions helps to control storage volumes.
- Use of HDF5 helps to reduce high file-count burdens on facility file systems.
- CBFlib 0.9.5
 - Can store arbitrary CBF files in HDF5 and recover them.
 - Supports the NeXus NXmx Application Definition for single crystal MX data.
 - Supports use of all CBFlib compressions in HDF5 files.
 - Provides minibcf2nexus to convert sets of minibcf files to a single NeXus file.
 - Provides cb2nexus to convert a CBF file describing a single scan to a single NeXus file containing the same data.
 - Provides nexus2cbf to convert back from a NeXus file to a CBF file.
- A simplified functional mapping for single crystal MX has been prepared.
- A full mapping extending the functional mapping to the general case is being finished.
- Updated CBF dictionary has been prepared.
- There is much work still to be done – collaborators welcome.

Data Rates, Formats and High Performance X-ray Detectors

CCD X-ray detectors provide images at a moderate data rate of one every few to several seconds. Current higher performance X-ray detectors, such as the DECTRIS Pilatus, are capable of collecting six-megapixel images at 10 – 25 frames per second [2], while the newest Pilatus 3 6M instruments can operate at 100 frames per second. The coming next generation of high performance X-ray detectors for MX such as the DECTRIS Eiger will be capable of collecting 16+ megapixel images at more than 125 frames per second [3, page 6] [4]. The ADSC DAMPAD [5] is also expected to produce 900 fine-sliced images in steps of two-tenths of a degree at 125 frames per second.

Typical Sustained Data Rates

Detector	Raw Image size (MB)	Frame Rate (Hz)	Compressed Rate (Gb/sec)	USB Disk Data Rate (%)
ADSC Q315 (2x2 binned)	18	0.37	.013	7
Pilatus 2 6M	24	10	.48	240
Pilatus 2 Fast 6M	24	25	1.2	600
Pilatus 3 6M	24	100	4.8	2400
Eiger 16M	72	125	18	9000

Typical sustained data rates for detectors used for MX at NSLS, Diamond Light Source, etc. compared to expected rates from Eiger, expressed in terms of the typical data rate for an inexpensive USB disk of 25 MB/sec = 200 Mb/sec.

Today for MX alone Diamond Light Source employs three Pilatus 6M fast and two Pilatus 3 6M, giving a combined data rate of over 1 GB/sec and over 200 files/sec. These new detectors are creating the need to manage hundreds of thousands of images being received at rates from sixty megapixels to 2.5 gigapixels per second and beyond. For the Advanced Beamlines for Biological Investigations with X-rays (ABBIIX) that are being built for NSLS-II [6], just two of the beam lines, the Frontier Macromolecular Crystallography (FMX) beamline and the Automated Macromolecular Crystallography (AMX) beamline [7], are expected to produce an aggregate of more than 94 terabytes per operational half day, 660 terabytes per week or 38 petabytes per year. The anticipated beamline flux is 10^{13} photons per second for FMX and 2×10^{13} photons per second for AMX, approximately 50 times the NSLS X25 and X29 fluxes. One subtle effect of these high fluxes is that there will be more photons per pixel in images, making them more difficult to compress.

Compression

- Long-standing issues in Crystallography
 - High speed, high compression ratio compression is a critical issue for the next generation of detectors.
 - Some compressions raise license issues.
 - Some popular compressions are slow or inefficient or both.
 - Some compressions can be handled in processing programs such as XDS if license and language issues can be addressed.
- Low pixel density fine-slicing with clean backgrounds makes some compressions more effective.
- CBFlib provides useful compressions.
- A plugin has been written to allow HDF5 to read and write CBFlib compressions.

For the DECTRIS Pilatus 300K image shown in Fig 2, the compressions were

Compression	CBF size (MB)	HDF5 size (MB)
raw binary	1.212	1.296
byte offset	0.309	0.393
HDF5 zlib	n/a	0.370
nibble offset	0.207	0.290
packed	0.184	0.267
canonical	0.178	0.262
external bzip2	0.164	0.169

All the HDF5 presentations of the data see a modest increase in size due to the overhead of the more complex format. For larger images this would not be as significant a percentage. This particular data, having a noisy background and significant spots, does not compress well. For many experiments using fast detectors, it is now feasible to take very large numbers of fine-sliced images that have very few photons per image, resulting in images that consist primarily of pixels containing zero with a small number of pixels with very few counts. Fortunately, such images often can be faithfully compressed by factors of 10 to 60. In one recent case, a compression by a factor of more than 1000 was achieved with bzip compression.

Software and Documentation

- Draft imgCIF/CBF version 1.7 dictionary that now includes information on going from CBF to NeXus:
<https://www.sites.google.com/site/nexuscbf/home/cbf-dictionary>
- PDF summary of the concordance:
<https://www.sites.google.com/site/nexuscbf/mapping-draft>
- CBFlib kit:
<http://downloads.sf.net/cbflib/CBFlib-0.9.5.tar.gz>

[†]Department of Mathematics and Computer Science, Dowling College, Oakdale, NY 11769 (USA)

^{*}Diamond Light Source, Harwell Science and Innovation Campus, OX11 0DE (UK)

[‡]<http://wiki.nexusformat.org/NIAC>

[§]<http://www.iucr.org/resources/cif/comcifs>



Cornell-SLAC pixel array detector (CSPAD)

Comparison: A CBF MX Data Frame File

A DECTRIS Pilatus 300K image (Fig 2) is shown both as a CBF and as the equivalent HDF5/NeXus file conforming to the NXmx application definition. The mapping between them is a matter of mapping CBF table entries into HDF5 groups, datasets and attributes in a tree organization. ###CBF: VERSION 1.7.10

```
# CIF file written by CBFlib v0.9.5
data_1191_00005
.array data.data
;
-CIF-BINARY-FORMAT-SECTION-
... B000000 B000000 D000000 D000000 ... F000000 9000000 D000000 9000000
12000000
-CIF-BINARY-FORMAT-SECTION-
;
.diffrr.id DLS_I19 .diffrr.crystal.id xtal001
.diffrr.source.diffrr.id DLS_I19 .diffrr.source.source synchrotron
.diffrr.source.type "Diamond Light Source Beamline I19"
.diffrr.radiation.diffrr.id DLS_I19 .diffrr.radiation.wavelength.id WAVELENGTH1
.diffrr.radiation.monochromator "Si 111"
.diffrr.radiation.polarizn.source_ratio 0.8 .diffrr.radiation.polarizn.source_norm 0.0
.diffrr.radiation.div_x.source 0.08 .diffrr.radiation.div_y.source 0.01
.diffrr.radiation.div_x.y.source 0.00
.diffrr.detector.diffrr.id DLS_I19 .diffrr.detector.id None
.diffrr.detector.type 'pilatus' .diffrr.detector.number_of_axes 4
```

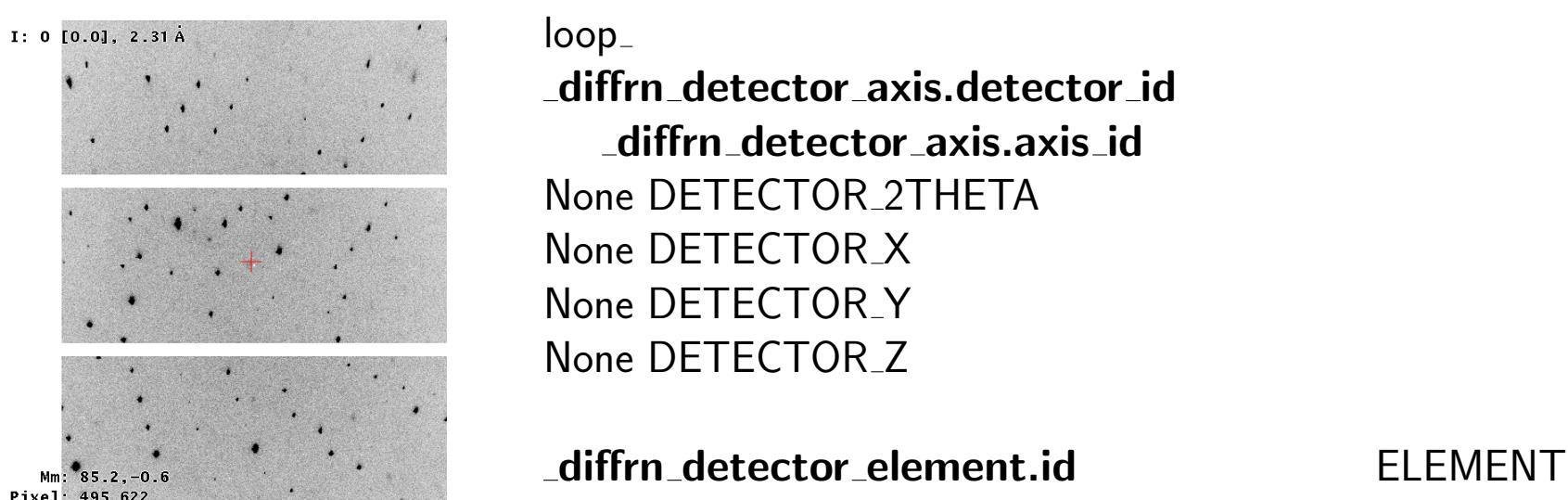


Figure 2: Sample Pilatus 300K diffraction image from DLS I19. Image produced by adxv

```
.loop_
.diffrr.detector.axis.detector.id
._diffrr.detector.axis.axis_id
None DETECTOR_2THETA
None DETECTOR_X
None DETECTOR_Y
None DETECTOR_Z

.diffrr.detector.element.id ELEMENT1
.diffrr.detector.element.detector.id None
;

.diffrr.data.frame.id FRAME1
.diffrr.data.frame.detector.element.id ELEMENT1

.diffrr.data.frame.array.id ARRAY1 .diffrr.data.frame.binary.id 1
.diffrr.scan.id SCAN1 .diffrr.scan.frame.id.start FRAME1
.diffrr.scan.frame.id.end FRAME1 .diffrr.scan.frames 1
.diffrr.measurement.diffrr.id DLS_I19 .diffrr.measurement.id GONIOMETER
.diffrr.measurement.number_of_axes 3 .diffrr.measurement.method rotation
.diffrr.measurement.sample.detector.distance 104.00

loop_
.diffrr.measurement.axis.measurement.id .diffrr.measurement.axis.axis_id
GONIOMETER GONIOMETER_OMEKA
GONIOMETER GONIOMETER_KAPPA
GONIOMETER GONIOMETER_PHI

.diffrr.radiation.wavelength.id WAVELENGTH1 .diffrr.radiation.wavelength.wavelength
0.68890 .diffrr.radiation.wavelength.wt 1

loop_
.diffrr.scan.axis.scan.id .diffrr.scan.axis.axis_id
.diffrr.scan.axis.angle.start .diffrr.scan.axis.angle.range .diffrr.scan.axis.angle.increment
.diffrr.scan.axis.displacement.start .diffrr.scan.axis.displacement.range
.diffrr.scan.axis.displacement.increment
;

SCAN1 GONIOMETER_OMEKA 23.0000 1.0000 1.0000 0.0 0.0 0.0
SCAN1 GONIOMETER_KAPPA 70.0000 0.0000 0.0000 0.0 0.0 0.0
SCAN1 GONIOMETER_PHI -179.0000 0.0000 0.0000 0.0 0.0 0.0
SCAN1 DETECTOR_2THETA 119.0000 0.0 0.0 0.0 0.0 0.0
SCAN1 DETECTOR_Z 0.0 0.0 0.0 104.00 0.0 0.0
SCAN1 DETECTOR_Y 0.0 0.0 0.0 0.0 0.0 0.0
SCAN1 DETECTOR_X 0.0 0.0 0.0 0.0 0.0 0.0

.diffrr.scan.frame.frame.id FRAME1 .diffrr.scan.frame.frame.number 1
.diffrr.scan.frame.integration_time 0.997000 .diffrr.scan.frame.exposure_time
1.000000
.diffrr.scan.frame.scan.id SCAN1 .diffrr.scan.frame.date 2013-08-15T13:21:53.634
```

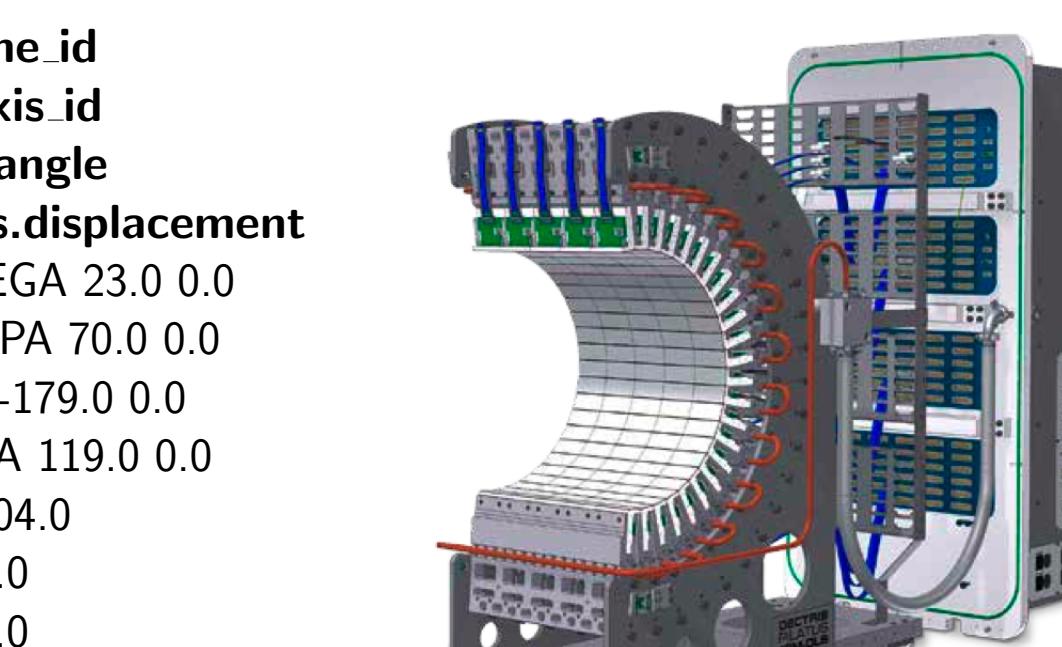


Figure 3: Curved DECTRIS detector for DLS beamline I23, an example of a detector with a complex geometry best described using the imgCIF/CBF ontology. NeXus has now adopted a similar approach to axis definitions.

```
.loop_
.axis.id axis.type
.axis.equipment axis.depends_on
.axis.vector[1] axis.vector[2]
.axis.vector[3]
.axis.offset[1] axis.offset[2]
.axis.offset[3]

GONIOMETER_OMEKA.rotation.goniometer .1 0 0 . .
GONIOMETER_KAPPA.rotation.goniometer .1 0 0 . .
GONIOMETER_PHI.rotation.goniometer .1 0 0 . .

SOURCE general.source .0 0 1 . .
GRAVITY general.gravity .0 -1 0 . .

DETECTOR_2THETA.rotation.detector .1 0 0 . .
DETECTOR_Z.translation.detector DETECTOR_2THETA 0 0 -1 0 0 0
DETECTOR_Y.translation.detector DETECTOR_2THETA 0 0 0 1 0 0
DETECTOR_X.translation.detector DETECTOR_2THETA 0 0 0 0 1 0
ELEMENT_X.translation.detector DETECTOR_2THETA 0 0 0 0 0 1
ELEMENT_Y.translation.detector ELEMENT_X 0 -1 0 20 ELEMENT_Y 0 0 1 0
ELEMENT_Z.translation.detector ELEMENT_X 0 0 0 0 1 0

loop_
.array.structure.list.array.id .array.structure.list.index
.array.structure.list.dimension .array.structure.list.precedence
.array.structure.list.direction .array.structure.list.axis.set.id
ARRAY1 1 487 1 increasing ELEMENT_X
ARRAY1 2 619 2 increasing ELEMENT_Y

loop_
.array.structure.list.axis.set.id .array.structure.list.axis.axis_id
.array.structure.list.axis.displacement .array.structure.list.axis.displacement.increment
ELEMENT_X ELEMENT_X 0.0 0.1720 ELEMENT_Y ELEMENT_Y 0.0 0.1720

loop_
.array.element.size.array.id .array.element.size.index .array.element.size.size
ARRAY1 1 0.000172 ARRAY1 2 0.000172

.array.intensities.array.id ARRAY1 .array.intensities.binary.id 1
.array.intensities.linearity linear .array.intensities.gain 1.0 .array.intensities.gain.esd
.array.intensities.overload 129889 .array.intensities.undefined.value -1

.array.structure.id ARRAY1 .array.structure.encoding.type "signed 32-bit integer"
.array.structure.compression.type none .array.structure.byte.order little_endian
```

Equivalent HDF5/NeXus File

```
/:NXroot
@creator="CBFlib"
@creator.version="0.9.5 (r492) 2014-04-27 12:00:40 -0400 (Sun, 27 Apr 2014)"
/entry:NXentry
@/data[NXdata]
@axes= [ "x", "y", "z" ]; @signal="data"
@x_indices=2; @y_indices=1
/@data = [11, 11, 13, 13, 11, ... 9, 13, 9, 18
@signal=1
@offset =0
@scaling.factor=1
/x=[0, 0.172, 0.344, 0.516, ... 83.248, 83.42, 83.592]
@depends.on=""/entry/instrument/detector/transformations/DETECTOR_X"
@equipment="detector"; @offset = [41.05, 52.32, 0]; @offset.units = "mm"
@transformation.type = "translation"; @units = "mm"
@vector= [-1, 0, 0]
/y=[0, 0.172, 0.344, 0.516, ... 105.952, 106.124, 106.296]
@depends.on=""/entry/instrument/detector/x.pixel.offset"
@equipment = "detector"; @offset = [0, 0, 0]; @offset.units = "mm"
@transformation.type = "translation"; @units = "mm"
@vector= [-1, 0, 0]
/definition="NXmx"
@version="1.2"
/instrument:NXinstrument
@/detector:NXdetector
@/count.time=1; @units = "s"
/@data → "/entry/data/data"
@depends.on=""/entry/instrument/detector/transformations/DETECTOR_X"
@description="pilatus"
@distance=104; @units = "mm"
@frame.start.time = "2013-08-15T13:21:53.634"
@offset → "/entry/data/offset"
@saturation.value=129883
@scaling.factor → "/entry/data/scaling.factor"
@transformations:NXtransformations
@/DETECTOR_2THETA=[119]
@depends.on=""; @equipment = "detector"; @offset = [0, 0, 0]; @offset.units = "mm"
@transformation.type = "rotation"; @units = "degrees"
@vector= [-1, 0, 0]
@/DETECTOR_X = [0]
@depends.on=""/entry/instrument/detector/transformations/DETECTOR_Y"
@equipment = "detector"; @offset = [0, 0, 0]; @offset.units = "mm"
@transformation.type = "translation"; @units = "mm"
@vector= [-1, 0, 0]
@/DETECTOR_Z = [104]
@depends.on=""/entry/instrument/detector/transformations/DETECTOR_2THETA"
@equipment = "detector"; @offset = [0, 0, 0]; @offset.units = "mm"
@transformation.type = "rotation"; @units = "degrees"
@vector= [0, 0, 1]
@undefined.value= [-1]
/x.pixel.offset → "/entry/data/x"
/x.pixel.size=0.172; @units = "mm"
/y.pixel.offset → "/entry/data/y"
/y.pixel.size=0.172; @units = "mm"
/monochromator:NXmonochromator
@/description="Si 111"
/source:NXsource
@/name="Diamond Light Source Beamline I19"
@/type = "synchrotron"
/transformations:NXtransformations
@/BEAM = []
@depends.on=""; @equipment = "general"; @system= "McStas.absolute"
@transformation.type = "general"
@vector= [0, 0, 1]
@/GRAVITY = []
@depends.on=""; @equipment = "general"; @system= "McStas.absolute"
@transformation.type = "general"
@vector= [0, -1, 0]
@/SOURCE= []
@depends.on=""; @equipment = "general"; @system= "McStas.absolute"
@transformation.type = "general"
@vector= [0, 0, -1]
@/UP= []
@depends.on=""; @equipment = "general"; @system= "McStas.absolute"
@transformation.type = "general"
@vector= [0, 1, 0]
/method="rotation"
/sample:NXsample
@/beam:NXbeam
@/incident.divergence.x= [0.08]; @units = "degrees"
@/incident.divergence.xy= [0]; @units = "degrees2"
@/incident.divergence.y= [0.01]; @units = "degrees"
@/incident.polarisation.stokes= [1, 0.8, 0, 0]
@/incident.wavelength=0.6889, @units = "angstroms"
@/weight=1
@depends.on=""/entry/sample/transformations/GONIOMETER_PHI"
@/transformations:NXtransformations
@/GONIOMETER_KAPPA = [70]
@depends.on=""/entry/sample/transformations/GONIOMETER_OMEKA"
@equipment = "goniometer"; @offset = [0, 0, 0]; @offset.units = "mm"
@transformation.type = "rotation"; @units = "degrees"
@vector= [-0.642788, -0.766044, 0]
@/GONIOMETER_OMEKA = [23]
@depends.on=""; @equipment = "goniometer"; @offset = [0, 0, 0]; @offset.units = "mm"
@transformation.type = "rotation"; @units = "degrees"
@vector= [-1, 0, 0]
@/GONIOMETER_PHI = [-179]
@depends.on=""/entry/sample/transformations/GONIOM
```