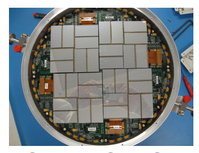
Coping with BIG DATA Image Formats: Integration of CBF, NeXus and HDF5, A Progress Report

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

[†]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
[Ψ]http://www.iucr.org/resources/cif/comcifs

Dectris Pilatus3 6M



Cornell-SLAC pixel array detector (CSpad)

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 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 miniCBF 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 minicbf2nexus to convert sets of minicbf files to a single NeXus file.
  - Provides cbf2nexus 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 Pilatus3 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 DMPAD [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 (ABBIX) 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`

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 B000000 ... F000000 9000000 D000000 9000000 12000000
–CIF-BINARY-FORMAT-SECTION—-
;
_diffrn.id DLS_I19 **_diffrn.crystal_id** xtal001
_diffrn_source.diffrn_id DLS_I19 **_diffrn_source.source** synchrotron
_diffrn_source.type 'Diamond Light Source Beamline I19'
_diffrn_radiation.diffrn_id DLS_I19 **_diffrn_radiation.wavelength_id** WAVELENGTH1
_diffrn_radiation.monochromator 'Si 111'
_diffrn_radiation.polarizn_source_ratio 0.8 **_diffrn_radiation.polarizn_source_norm** 0.0
_diffrn_radiation.div_x_source **0.08** **_diffrn_radiation.div_y_source** 0.01
_diffrn_radiation.div_x_y_source 0.00
_diffrn_detector.diffrn_id DLS_I19 **_diffrn_detector.id** None
_diffrn_detector.type 'pilatus' **_diffrn_detector.number_of_axes** 4

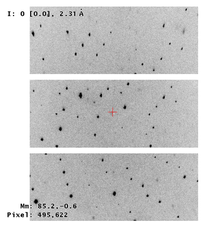


Figure 2: Sample Pilatus 300K diffraction image from DLS I19. Image produced by adxv

loop_
_diffrn_detector_axis.detector_id
_diffrn_detector_axis.axis_id
None DETECTOR_2THETA
None DETECTOR_X
None DETECTOR_Y
None DETECTOR_Z

_diffrn_detector_element.id ELEMENT1
_diffrn_detector_element.detector_id None -

_diffrn_data_frame.id FRAME1
_diffrn_data_frame.detector_element_id ELEMENT1

_diffrn_data_frame.array_id ARRAY1 **_diffrn_data_frame.binary_id** 1
_diffrn_scan.id SCAN1 _diffrn_scan.frame_id_start FRAME1
_diffrn_scan.frame_id_end FRAME1 _diffrn_scan.frames 1
_diffrn_measurement.diffrn_id DLS_I19 _diffrn_measurement.id GONIOMETER
_diffrn_measurement.number_of_axes 3 **_diffrn_measurement.method** rotation
_diffrn_measurement.sample_detector_distance 104.00

loop_
_diffrn_measurement_axis.measurement_id _diffrn_measurement_axis.axis_id
GONIOMETER GONIOMETER_OMEGA
GONIOMETER GONIOMETER_KAPPA
GONIOMETER GONIOMETER_PHI

_diffrn_radiation_wavelength.id WAVELENGTH1 **_diffrn_radiation_wavelength.wavelength** 0.68890 **_diffrn_radiation_wavelength.wt** 1

loop_
_diffrn_scan_axis.scan_id _diffrn_scan_axis.axis_id
_diffrn_scan_axis.angle_start _diffrn_scan_axis.angle_range _diffrn_scan_axis.angle_increment
_diffrn_scan_axis.displacement_start _diffrn_scan_axis.displacement_range
_diffrn_scan_axis.displacement_increment
SCAN1 GONIOMETER_OMEGA 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

_diffrn_scan_frame.frame_id FRAME1 **_diffrn_scan_frame.frame_number** 1
_diffrn_scan_frame.integration_time 0.997000 **_diffrn_scan_frame.exposure_time** 1.000000
_diffrn_scan_frame.scan_id SCAN1 **_diffrn_scan_frame.date** 2013-08-15T13:21:53.634

loop_
_diffrn_scan_frame_axis.frame_id
_diffrn_scan_frame_axis.axis_id
_diffrn_scan_frame_axis.angle
_diffrn_scan_frame_axis.displacement
FRAME1 GONIOMETER_OMEGA 23.0 0.0
FRAME1 GONIOMETER_KAPPA 70.0 0.0
FRAME1 GONIOMETER_PHI -179.0 0.0
FRAME1 DETECTOR_2THETA 119.0 0.0
FRAME1 DETECTOR_Z 0.0 104.0
FRAME1 DETECTOR_Y 0.0 0.0
FRAME1 DETECTOR_X 0.0 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]

GONIOMETER_OMEGA rotation goniometer . 1 0 0 . . .
GONIOMETER_KAPPA rotation goniometer
GONIOMETER_OMEGA 0.642788 -0.766044 0 . . .
GONIOMETER_PHI rotation goniometer GONIOMETER_KAPPA 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_Z 0 -1 0 0 0 0
DETECTOR_X translation detector DETECTOR_Y 1 0 0 0 0 0
ELEMENT_X translation detector DETECTOR_X 1 0 0 -41.05 52.32 0
ELEMENT_Y translation detector ELEMENT_X 0 -1 0 0 0 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.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 1298839 **_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

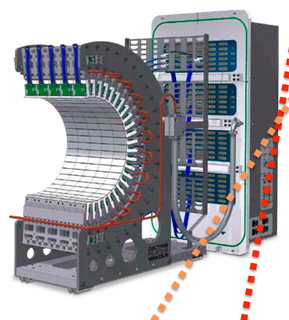


Figure 3. Curved DECTRIS detector for DLS beamline I23, an example of a detector with a complex geometry best described using the imgCIF/CBF ontology. NeXus has now adopted a similar approach to axis definitions.

Complex Mapping between imgCIF Tables and NeXus Trees

Polarization is moving to Stokes parameters

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= ["", "y", "x"]; @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= [0, -1, 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=1298839
/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_Y = [0]
@depends_on="/entry/instrument/detector/transformations/DETECTOR_Z"
@equipment = "detector"; @offset = [0, 0, 0]; @offset_units = "mm"
@transformation_type = "translation"; @units = "mm"
@vector = [0, -1, 0]
/DETECTOR_Z = [104]
@depends_on="/entry/instrument/detector/transformations/DETECTOR_2THETA"
@equipment ="detector"; @offset = [0, 0, 0]; @offset_units ="mm"
@transformation_type ="translation"; @units ="mm"
@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 ="degrees^2"
/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_OMEGA"
@equipment ="goniometer"; @offset = [0, 0, 0]; @offset_units ="mm"
@transformation_type ="rotation"; @units ="degrees"
@vector= [-0.642788, -0.766044, 0]
/GONIOMETER_OMEGA= [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/GONIOMETER_KAPPA"
@equipment ="goniometer"; @offset = [0, 0, 0]; @offset_units ="mm"
@transformation_type ="rotation"; @units ="degrees"
@vector= [-1, 0, 0]

CBF compressions are available in NeXus via an HDF5 filter plugin In CBFlib

NeXus axes are now handled similarly to CBF, but are distributed in the NeXus tree to the appropriate components

- **The minimum required NeXus application definition for single crystal monochromatic macromolecular crystallography is given in the NXmx application definition**:
  `http://download.nexusformat.org/sphinx/classes/contributed_definitions/NXmx.html`

References

[1] Bernstein, H. J. et al., Computational Crystallography Newsletter **5** (2014) 12 .

[2] Trueb, P. et al., Journal of Synchrotron Radiation **19** (2012) 347 .

[3] Willmott, P., *An Introduction to Synchrotron Radiation: Techniques and Applications*, John Wiley and Sons, Chichester, UK, 2011, page 6.

[4] Johnson, I. et al., Journal of Synchrotron Radiation **19** (2012) 0.

[5] Hamlin, R., Hontz, T., and Nielsen, C., The new dual mode pixel array detector, in *Meeting of the American Crystallographic Association, Boston, MA, 28 July – 1 August 2012*, American Crystallographic Association, 2012, Abstract 11.01.1151, see also WO Patent App. PCT/US2012/044,458.

[6] Hendrickson, W. A., NSLS-II - Status of the Life Sciences Program, Technical report, Brookhaven National Laboratory, X6A Science Advisory Committee, 2012, `protein.nsls.bnl.gov/mediawiki/images/e/e3/Hendrickson_2012.pdf`.

[7] Schneider, D. K., Sweet, B. M., and Skinner, J., Projection of MX ABBIX needs at AMX and FMX for data acquisition, data processing, software, data archiving and networking, Private Communication, 2012.

Acknowledgements

Our thanks to James Hester, Chair of COMCIFS and Mark Koennecke, Chair of NIAC, for supporting and encouraging this effort and to all participants in COMCIFS and NIAC for helpful comments. Our thanks to Nick Sauter and Aaron Brewster for extending this work to CSPAD. Our thanks for years of supporting efforts at the BNL PXRR Group: Robert M. Sweet, Dieter Schneider, Howard Robinson, John Skinner, Matt Cowan, Leonid Flaks, Richard Buono; at DLS: Alun Ashton, Bill Pulford; and at the Dowling College ARCiB Lab Group: Mojgan Asadi, Kostandina Bardhi, Keti Bardhi Limone Rosa, Our thanks to DECTRIS, BIOIHDF and the HDF Group. Our thanks to Frances C. Bernstein. This work was funded in part by NIGMS, DOE, NSF, PaNdata ODI (EU 7th Framework Programme).