DXfile Documentation

Release 0.6

Argonne National Laboratory

Dec 14, 2022

CONTENTS

Scientific Data Exchange [A1] is a set of guidelines for storing scientific data and metadata in a Hierarchical Data Format 5 [B7] file.

HDF5 [B7] has many important characteristics for scientific data storage. It offers platform-independent binary data storage with optional compression, hierarchical data ordering, and support for MPI-based parallel computing. Data are stored with alphanumeric tags, so that one can examine a HDF5 file's contents with no knowledge of how the file writing program was coded. Tools for this examination include the HDF5-supplied command-line utility [B6] to examine the contents of any HDF5 file, or the freely-available Java program [B8] to interactively examine the file.

At synchrotron facilities using the EPICS [B1] software for area detectors [B12] with the NDFileHDF5 plugin [B11], is possible to save Data Exchange files by properly configure the detector and the HDF schema attribute files .

This reference guide describes the basic design principles of Data Exchange, examples of their application, a core reference for guidelines common to most uses, and coding examples.

CHAPTER

ONE

FEATURES

- The definition of the scientific data exchange.
- A python interface for writing scientific data exchange files.
- XML attribute files for writers with the EPICS Area Detector HDF plug-in.

CHAPTER

TWO

HIGHLIGHTS

- Based on Hierarchical Data Format 5 (HDF5).
- Focuses on technique rather than instrument descriptions.
- Provenance tracking for understanding analysis steps and results.
- Ease of readability.

CHAPTER

THREE

CONTRIBUTE

- Documentation: https://github.com/data-exchange/dxfile/tree/master/doc
- Issue Tracker: https://github.com/data-exchange/dxfile/issues
- Source Code: https://github.com/data-exchange/dxfile

CHAPTER

FOUR

CONTENTS

4.1 Introduction

4.1.1 Root Level Structure

While HDF5 gives great flexibility in data storage, straightforward file readability and exchange requires adhering to an agreed-upon naming and organizational convention. To achieve this goal, Data Exchange adopts a layered approach by defining a set of **mandatory** and optional fields.

The general structure of a Data Exchange file is shown in table [tab:genrules]. The most basic file must have an **implements** string, and an **exchange** group at the root level/group of the HDF5 file. Optional *measurement* and *process* groups are also defined. Beyond this, additional groups may be added to meet individual needs, with guidelines suggesting the best structure.

Member	Type	Example
implements	string dataset	**exchange**:*measurement*:*process*
exchange	group	
measurement	group	
process	group	

implements

Mandatory scalar string dataset in the root of the HDF5 file whose value is a colon separated list that shows which components are present in the file. All components listed in the **implements** string are to be groups placed in the HDF5 file at the root level/group. In a minimal Data Exchange file, the only **mandatory** item in this list is **exchange**. A more general Data Exchange file also contain *measurement* and possibly *process*, in which case the **implements** string would be: **exchange**:*measurement*:*process*.

exchange

Mandatory group containing one or more arrays that represent the most basic version of the data, such as raw or normalized optical density maps or a elemental signal map. **Exchange_N** is used when more than one core dataset or derived datasets are saved in the file. The **exchange** implementation for specific techniques are defined in separate sections in the Reference Guide.

measurement

Optional group containing the *measurement* made on the sample; *measurement* contains information about the sample and the instrument; *measurement_N* is used when more than one *measurement* is stored in the same file.

process

The Process group describes all the "work" that has been done. This includes data processing steps that have been applied to the data as well as experimental steps (e.g. data collection strategy etc.) and sample preparation ahead of the experiment and during the measurement (e.g. environment conditions etc.).

In a Data Exchange file, each dataset has a unit defined using the units attribute. units is not **mandatory** - if omitted, the default unit as defined in Appendix [appendix:units] is used.

The detailed rules about how to store datasets within the exchange group are best shown through examples in the next section. Detailed reference information can be found in the section.

4.1.2 Definitions

Color code

All the diagrams in this section follow the color conventions shown in *Color Code*. The basic elements are HDF5 datasets, attributes, and groups. We also support internal references to elements in the file by a simple scalar string that holds the path of the dataset within the file. On the diagram, this is shown as a reference dataset that points to the referred-to dataset. Note that we use this mechanism rather than HDF5 hard or soft links

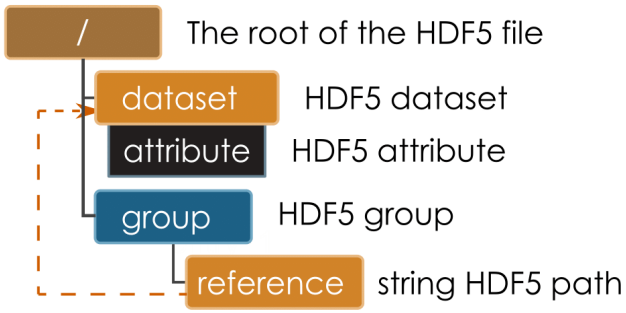


Fig. 1: Color Code
Explanation of the color code used in the diagrams

Multidimensional data

A multidimensional dataset should be described as fully as possible, with units for the dataset as well as dimension descriptors (that also have units defined). There are also additional descriptive fields available such as title and description. The order of dimensions in the dataset should put the slowest changing dimension first, and the fastest changing dimension last.

It is strongly encouraged that all datasets have a units attribute. The string value for units should preferably be an SI unit, however well understood non-SI units are acceptable, in particular *degrees*. The units strings should conform to those defined by UDUNITS [B2]. While UDUNITS is a software package, it contains simple XML files that describe units strings and acceptable aliases.

The axes of a multidimensional dataset are described through the use of additional one-dimensional datasets (dimension descriptors), one for each axis in the main dataset. Take for example a 3-dimensional cube of images, with axes of x,

y, and z where z represents the angle of the sample when each image was taken. There should be 3 additional one-dimensional datasets called x, y, and z where x and y contain an integer sequence, and z contains a list of angles. X and y have units of *counts* and z has units of *degree*. To simplify, it is acceptable to omit x and y, since the default interpretation will always be an integer sequence.

The dimension descriptors (x, y, z) can be associated with the main dataset through two mechanisms. The HDF5 libraries contain a function call H5DSattach_scale to *attach* a dimension descriptor dataset to a given dimension of the main dataset. HDF5 takes care of entering several attributes in the file that serve to keep track of this association. If the particular programming language you work in does not support this HDF5 function, then you can instead add a string attribute to your main dataset called axes. The axes attribute is simply a colon separated string naming the dimension descriptor datasets in order, so *z:y:x* in this case. Additional examples below show this in action.

4.1.3 Data Structure

A tomographic data set consists of a series of projections, dark and white field images. The dark and white fields must have the same projection image dimensions and can be collected at any time before, after or during the projection data collection. The angular position of the tomographic rotation axis, theta, can be used to keep track of when the dark and white images are collected. These examples show projection, dark, and white images saved in three 3D arrays as shown in *Basic Tomo A* and *Basic Tomo B* using, by default, the natural HDF5 order of the multidimensional array (rotation axis, ccd y, ccd x), i.e. with the fastest changing dimension being the last dimension, and the slowest changing dimension being the first dimension. If using the default dimension order, the axes attribute *theta:y:x* can be omitted. The attribute is **mandatory** if the 3D arrays use a different axes order. This could be the case when, for example, the arrays are optimized for sinogram read *y:theta:x*. As no units are specified the data is assumed to be in *counts* with the axes (x, y) in pixels. If the positions of the rotation axis for each projection, dark, and white images are not specified via theta dimension scale datasets, it is assumed that the raw projections are taken at equally spaced angular intervals between 0 and 180 degree, with white and dark field collected at the same time before or after the projection data collection.

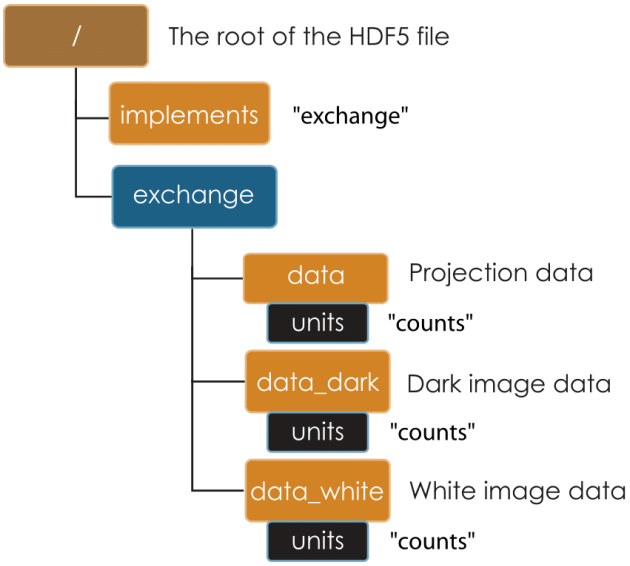


Fig. 2: Basic Tomo A
Diagram of a minimal Data Exchange file for a single tomographic data set including raw projections, dark, and white fields

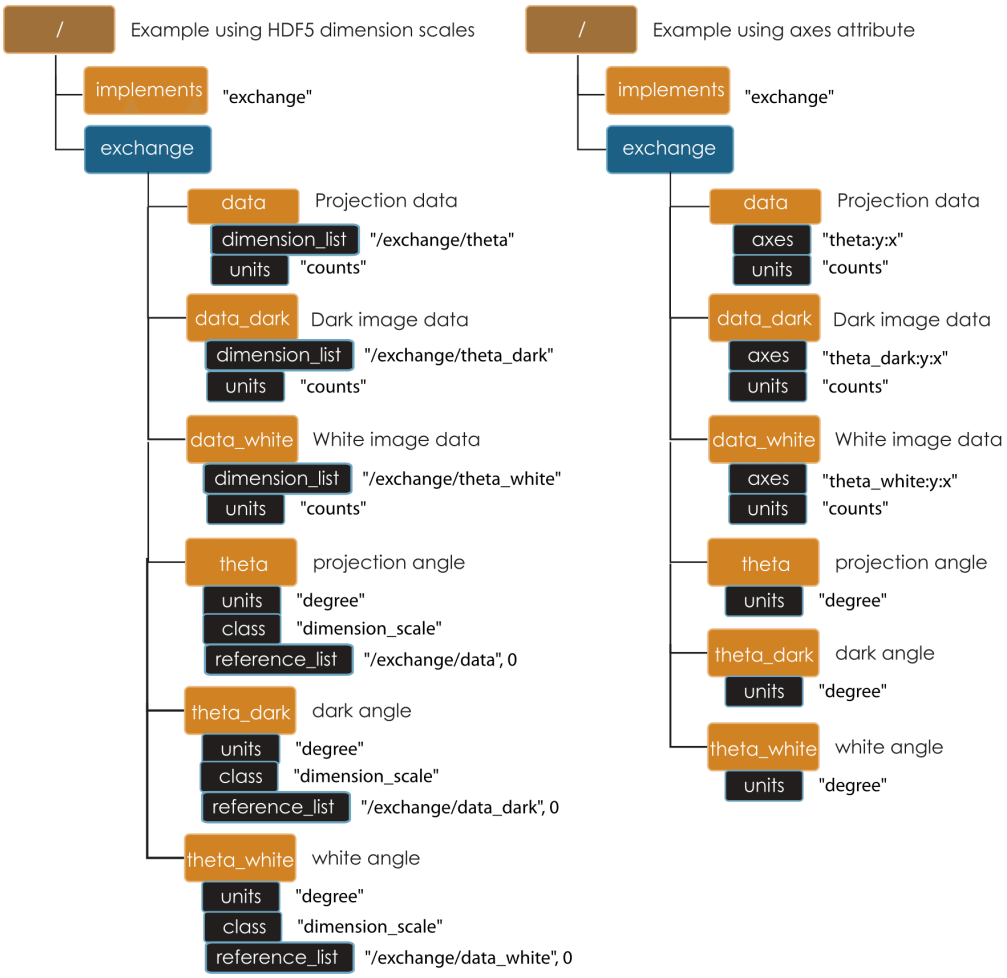


Fig. 3: Basic Tomo B

Diagram of a single tomographic data set including raw projections, dark and white fields. In this case, there are additional dimension descriptor datasets theta, theta_dark, and theta_white that contain the positions of the rotation axis for each projection, dark, and white image. The lefthand example shows this as it would appear using the HDF5 H5DSattach_scale function. The righthand example shows this as it would appear by manually adding an axes attribute (for cases where H5DSattach_scale is unavailable)

Imaging

The examples in this section show how one can store data for imaging experiments using the Data Exchange format. It is general enough, however, to show how Data Exchange can be extended or adapted to other techniques. These examples are meant to give a flavor for our approach. A complete reference to the core structure can be found in Section *Reference*. Technique specific extensions to the core structure can be found at the end of the Reference Guide.

Minimal DXfile shows a diagram of a minimal Data Exchange file to store a single projection image. It is strongly encouraged that all datasets shall have a units attribute. The axes of the dataset are not specified in this minimal case, and can be assumed to be x and y with a zero-based integer sequence, or more simply, pixels.

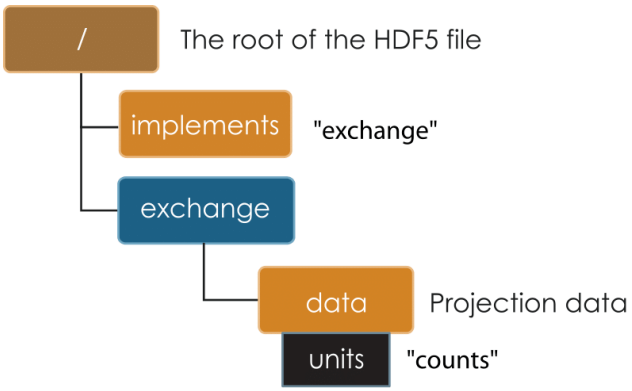


Fig. 4: Minimal DXfile
Diagram of a minimal Data Exchange file for a single image.

Series

A series of tomographic measurements, when relevant, can be stored in the same file appending _N to the measurement tag. A series of tomographic data sets are typically collected changing the instrument status (energy, detector or optics position); changing the sample status (position, environment etc.). Figure *Temperature*, *Energy* and *Distance* show the content of files changing the sample temperature, the X-ray source energy and detector-sample distance. In nano tomography experiments, for example, the detector field of view is often smaller than the sample. To collect a complete tomographic data set, it is necessary to raster the sample across the field of view moving its x and y location. Figure *Raster* shows a file from a nano tomography experiment when the sample rasters through the field of view.

There are limits to this approach, as one clearly does not want to have hundreds of measurement groups in a file (or multiple files) where most of the metadata is the same. For measurements where there are many “positioner” values (aka a “scan”), it is more sensible to add dimension(s) to the exchange dataset, and describe the “positioner” values as dimension scales. This is a judgement left to the user.

Temperature

Energy

Detector-sample distance

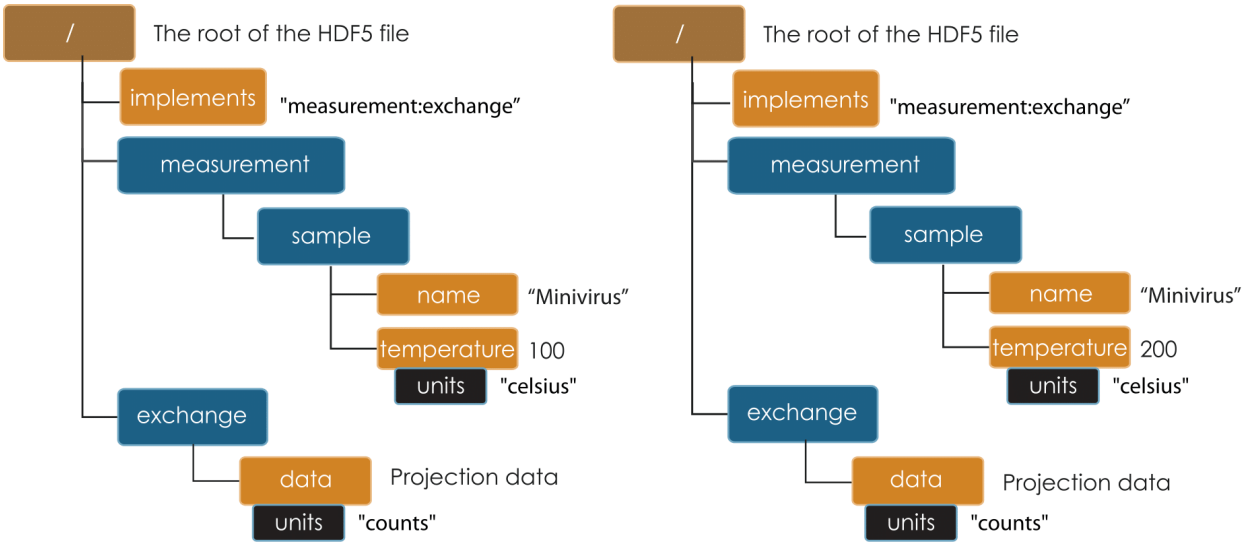


Fig. 5: Temperature
Diagram of two tomographic data sets taken at two different sample temperatures (100 and 200 Celsius)

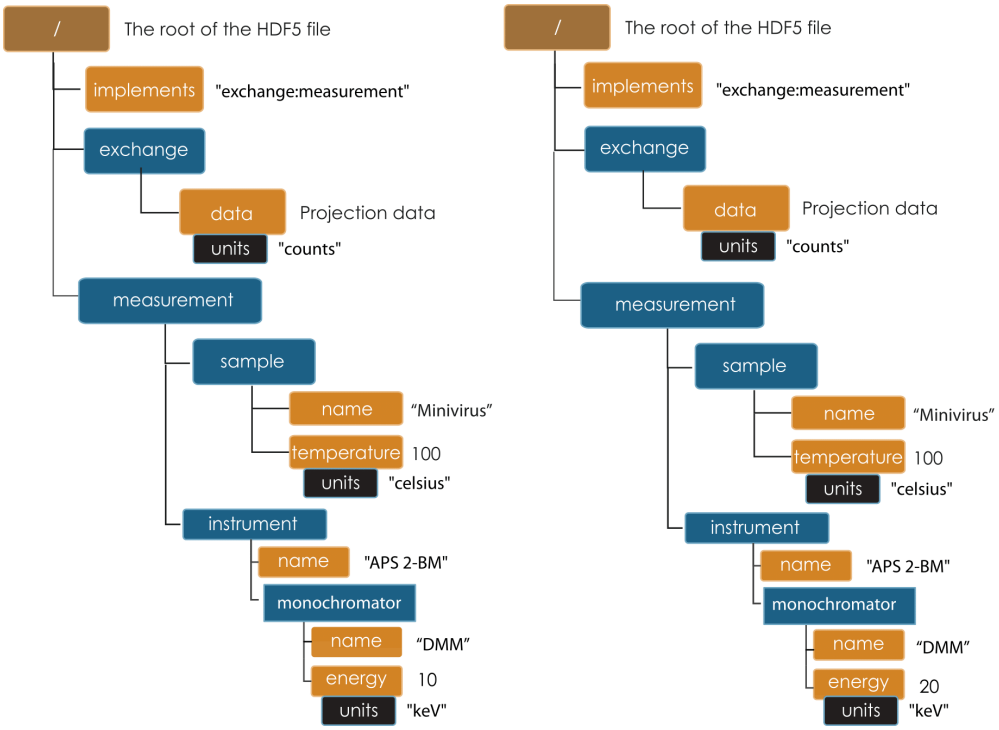


Fig. 6: Energy
Diagram of two tomographic data sets taken at two different energy (10 and 20 keV)

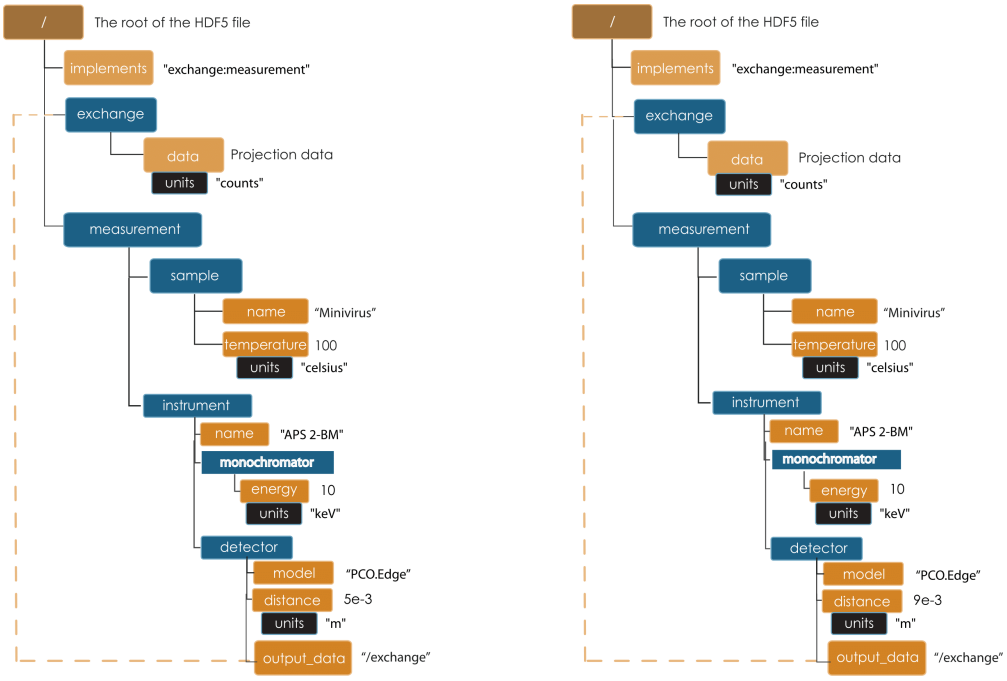


Fig. 7: Distance

Diagram of two tomographic data sets collected with two different detector-sample distances (5 and 9 mm). Note the use of output_data dataset to associate the detector with the exchange group generated from the acquisition

Raster

4.2 Core Reference

4.2.1 Top level (root)

This node represents the top level of the HDF5 file and holds some general information about the file.

Member	Type	Example
implements	string dataset	**exchange**:*measurement*:*process*
exchange	group	
measurement	group	
process	group	

implements

A colon separated list that shows which components are present in the file. The only **mandatory** component is **exchange**. A more general Data Exchange file also contains *measurement* and *process* information, if so these will be declared in **implements** as **exchange**:*measurement*:*process*

exchange or exchange_N

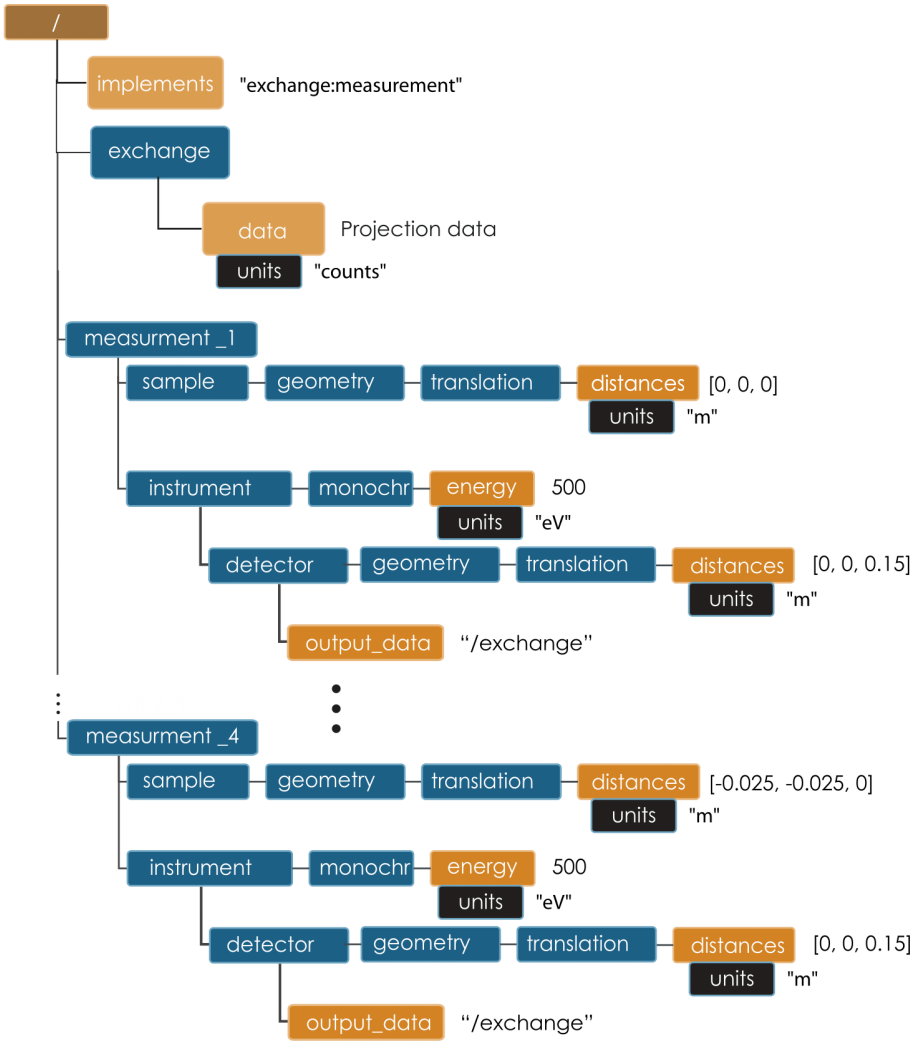


Fig. 8: Raster
Diagram of a file with 4 tomographic data sets from a nano tomography experiment

The data taken from measurements or processing. Dimension descriptors within the group may also serve to describe "positioner" values involved in a scan.

measurement or measurement_N

Description of the sample and instrument as configured for the measurement. This group is appropriate for relatively static metadata. For measurements where there are many "positioner" values (aka a "scan"), it is more sensible to add dimension(s) to the exchange dataset, and describe the "positioner" values as dimension scales rather than record the data via multiple matching *measurement* and **exchange** groups. This is a judgement left to the user.

process

The Process group describes all the "work" that has been done. This includes data processing steps that have been applied to the data as well as experimental steps (e.g. data collection strategy etc.) and sample preparation ahead of the experiment and during the measurement (e.g. environment conditions etc.).

4.2.2 exchange

The exchange group is where scientific datasets reside. This group contains one or more array datasets containing n-dimensional data and optional descriptions of the axes (dimension scale datasets). Exactly how this group is used is dependent on the application, however the general idea is that one exchange group contains one cohesive dataset. If, for example, the dataset is processed into some other form, then another exchange group is used to store the derived data.

Multiple exchange groups are numbered consecutively as **exchange_N**. At a minimum, each exchange group should have a primary dataset named **data**. The *title* is optional.

Member	Type	Example
name	string dataset	"absorption_tomography"
description	string dataset	"raw absorption tomo"
data	array dataset	n-dimensional dataset

Table: Exchange Group Members

name

Descriptive *name* for **data** dataset. Current types include: absorption_tomography, phase_tomography, dpc_tomography

description

Description.

data

The primary scientific dataset. Additional related datasets may have any arbitrary name. Each dataset should have a units and description attribute. Discussion of dimension descriptors and optional axes attribute is covered in Section [sec:multidims].

Attribute

Description and units can be added as attribute to any data, both array or values, inside a data exchange file. If units is omitted default is SI.

Member	Type	Example
description	string attribute	"transmission"
units	string attribute	*counts*

Table: data attributes

4.2.3 *measurement*

This group holds sample and instrument information. These groups are designed to hold relatively static data about the sample and instrument configuration at the time of the measurement. Rapidly changing *positioner* values (aka scan) are better represented in the exchange group dataset.

Member	Type	Example
instrument	group	
sample	group	

Table: Measurement Group Members

instrument

The instrument used to collect this data.

sample

The sample measured.

instrument

The instrument group stores all relevant beamline components status at the beginning of a measurement. While all these fields are optional, if you do intend to include them they should appear within this parentage of groups.

Member	Type	Example
name	string dataset	"XSD/2-BM"
component_1	group	
component_2	group	
component_n	group	
setup	group	

Table: Instrument

name

Name of the instrument.

component

List of components part of the instrument. Replace *component* with the actual item name, *source*, *mirror*, etc.

detector

The detectors that compose the instrument.

component

Class describing the component being used.

Member	Type	Example
name	string dataset	"APS"
description	string dataset	"APS"
arbitrary_label_1	string dataset	"what ever"
arbitrary_label_2	string dataset	"what ever"
arbitrary_label_n	string dataset	"what ever"
setup	group	
geometry	group	

Table: Component Description

name

Name.

arbitrary_label(s)

Date and time source was measured.

setup

Logging instrument and beamline component setup parameters (static setup values) is not defined by Data Exchange because is specific and different for each instrument and beamline. To capture this information Data Exchange requires to set a *setup* group under each beamline component and leaves each facility free to store what is relevant for each component (list of motor positions etc.). Ideally each component in the instrument list (source, shutter, attenuator etc.) should have included its setup group. For setup values not associated with a specific beamline component a *setup* group in the instrument group should be created.

Member	Type	Example
positioner_x	float	-10.107
positioner_y	float	-17.900
positioner_z	float	-5.950

Table: Setup Group Members

geometry

The geometry group is common to many of the subgroups under measurement. The intent is to describe the translation and rotation (orientation) of the sample or instrument component relative to some coordinate system. Since we believe it is not possible to determine all possible uses at this time, we leave the precise definition of geometry up to the technique. We do encourage the use of separate translation and orientation subgroups within geometry. As such, we do not describe geometry further here. This class holds the general position and orientation of a component.

Member	Type	Example
translation	group	
orientation	group	

translation

The position of the object with respect to the origin of your coordinate system.

orientation

The rotation of the object with respect to your coordinate system.

translation

This is the description for the general spatial location of a component for tomography.

Member	Type	Example
distances	3 float array dataset	(0, 0.001, 0)

distances

The x, y and z components of the translation of the origin of the object
relative to the origin of the global coordinate system (the place where
the X-ray beam meets the sample when the sample is first aligned in the beam).
If distances does not have the attribute units set then the units are in
meters.

orientation

This is the description for the orientation of a component for tomography.

Member	Type	Example
value	6 float array dataset	

value

Dot products between the local and the global unit vectors. Unitless

The orientation information is stored as direction cosines. The direction cosines will be between the local coordinate directions and the global coordinate directions. The unit vectors in both the local and global coordinates are right-handed and orthonormal.

Calling the local unit vectors (x', y',z') and the reference unit vectors (x, y, z) the six numbers will be

$$[x \cdot x, x' \cdot y, x' \cdot z, y' \cdot x, y' \cdot y, y' \cdot z]$$

where

' . '

is the scalar dot product (cosine of the angle between the unit vectors).

Notice that this corresponds to the first two rows of the rotation matrix that transforms from the global orientation to the local orientation. The third row can be recovered by using the fact that the basis vectors are orthonormal.

sample

This group holds basic information about the sample, its geometry, properties, the sample owner (user) and sample proposal information. While all these fields are optional, if you do intend to include them they should appear within this parentage of groups.

Member	Type	Example
name	string dataset	"cells sample 1"
description	string dataset	"malaria cells"
preparation_date	string dataset (ISO 8601)	"2012-07-31T21:15:22+0600"
chemical_formula	string dataset (abbr. CIF format)	"(Cd 2+)3, 2(H2 O)"
mass	float dataset	0.25
concentration	float dataset	0.4
environment	string dataset	"air"
temperature	float dataset	25.4
temperature_set	float dataset	26.0
pressure	float dataset	101325
thickness	float dataset	0.001
position	string dataset	"2D" APS robot coord.
geometry	group	
setup	group	
experiment	group	
experimenter	group	

Table: Sample Group Members

name

Descriptive name of the sample.

description

Description of the sample.

preparation_date

Date and time the sample was prepared.

chemical_formula

Sample chemical formula using the CIF format.

mass

Mass of the sample.

concentration

Mass/volume.

environment

Sample environment.

temperature

Sample temperature.

temperature_set

Sample temperature set point.

pressure

Sample pressure.

thickness

Sample thickness.

position

Sample position in the sample changer/robot.

geometry

Sample center of mass position and orientation.

experiment

Facility experiment identifiers.

experimenter

Experimenter identifiers.

experiment

This provides references to facility ids for the proposal, scheduled activity, and safety form.

Member	Type	Example
proposal	string dataset	"1234"
activity	string dataset	"9876"
safety	string dataset	"9876"

Table: Experiment Group Members

proposal

Proposal reference number. For the APS this is the General User
Proposal number.

activity

Proposal scheduler id. For the APS this is the beamline scheduler activity id.

safety

Safety reference document. For the APS this is the Experiment
Safety Approval Form number.

experimenter

Description of a single experimenter. Multiple experimenters can be represented through numbered entries such as experimenter_1, experimenter_2.

Member	Type	Example
name	string dataset	"John Doe"
role	string dataset	"Project PI"
affiliation	string dataset	"University of California, Berkeley"
address	string dataset	"EPS UC Berkeley CA 94720 4767 USA"
phone	string dataset	"+1 123 456 0000"
email	string dataset	"johndoe@berkeley.edu"
facility_user_id	string dataset	"a123456"

Table: Experimenter Group Members

name: User name.

role: User role.

affiliation: User affiliation.

address: User address.

phoen: User phone number.

email: User e-mail address

facility_user_id: User badge number

4.2.4 *process*

Process is the documentation of the data collection strategy (*acquisition*) steps, all transformations, analyses and interpretations of data performed by a sequence of process functions (*actor*) as well as any sample preparation step done ahead of the experiment and during the measurement (e.g. environment conditions etc.).

Maintaining this history, also called provenance, allows for reproducible data. The Data Exchange format tracks process by allowing each actor to append process information to a process table.

The process table tracks provenance in the execution order as a series of processing steps by appending sequential actor entries in the process table.

Member	Type	Example
name	string dataset	"name"
description	string dataset	"optional"
actor_1	group	
actor_2	group	
actor_n	group	
table	group	

Table: Process Group Members

name

Descriptive process task.

description

Description of the process task.

actor

This is the actor description group. Each entry of the process *table* will refer to the correspondent actor description.

Member	Type	Example
name	string dataset	"test rec"
description	string dataset	"optional"
version	string dataset	https://github.com/tomopy_scripts/b9ad87e17
input_data	string dataset	"/exchange"
output_data	string dataset	"/exchange_1"
set-up	group	

Table: Actor Group Members

name

Descriptive actor task.

description

Description of the actor task.

version

Version of the actor task.

If available this can be the repository link to the actor version used
https://github.com/tomopy_scripts/b9ad87e17

input_data, output_data

Origin and destination of the data processed by the actor.

setup (actor)

Here is where to log the actor setup parameters (static setup values).

Member	Type	Example
parameter_name_1	float	0.0
parameter_name_2	string dataset	"Parzen"
parameter_name_n	float	2.0
module__name_1	string dataset	https://github.com/astra/b9ad87e17
module_name_2	string dataset	https://github.com/tomopy/c9ad87e77

Table: Actor Setup Group

table

Scientific users will not generally be expected to maintain data in this group. The expectation is that the data collection and analysis pipeline tools will automatically record process steps using this group. In addition, it is possible to re-run an analysis using the information provided here.

actor	start_time	end_time	status	message	reference	description
actor_1	21:15:22	21:15:23	SUCCESS	OK	/process/actor_1	raw data collection
actor_2	21:15:26	21:15:27	RUNNING	OK	/process/actor_2	reconstruct
actor_n	21:17:28	22:15:22	QUEUED	OK	/process/actor_n	transfer data to user

Table: Process table to log actors activity

actor

Name of the process in the pipeline stage that is executed at this step.

start_time

Time the process started.

end_time

TIme the process ended.

status

Current process status. May be one of the following: QUEUED, RUNNING, FAILED, or SUCCESS.

message

A process specific message generated by the process. It may be a

confirmation that the process was successful, or a detailed error message, for example.

reference

Path to the actor description group. The process description group contains all metadata to perform the specific process. This reference is simply the HDF5 path within this file of the technique specific process description group. The process description group should contain all parameters necessary to run the process, including the name and version of any external analysis tool used to process the data. It should also contain input and output references that point to the **exchange_N** groups that contain the input and output datasets of the process.

description

Process description.

4.3 X-ray Tomography

This section describes extensions and additions to the core Data Exchange format for X-ray Tomography. We begin with the extensions to the exchange and instrument groups, and then describe the possible tomography data collection schemes and corresponding data structures.

4.3.1 Top level (root)

This node represents the top level of the HDF5 file and holds some general information about the file.

Member	Type	Example
implements	string dataset	**exchange**:*measurement*:*process*
exchange	group	
measurement	group	
process	group	

implements

A colon separated list that shows which components are present in the file. The only **mandatory** component is **exchange**. A more general Data Exchange file also contains *measurement* and *process* information, if so these will be declared in **implements** as **exchange**:*measurement*:*process*

exchange or exchange_N

The data taken from measurements or processing. Dimension descriptors within the group may also serve to describe "positioner" values involved in a scan.

measurement or measurement_N

Description of the sample and instrument as configured for the measurement. This group is appropriate for relatively static metadata. For measurements where there are many "positioner" values (aka a "scan"), it is more sensible to add dimension(s) to the exchange dataset, and describe the "positioner" values as dimension scales rather than record the data via multiple matching *measurement* and **exchange** groups. This is a judgement left to the user.

process

The Process group describes all the "work" that has been done. This includes data processing steps that have been applied to the data as well as experimental steps (e.g. data collection strategy etc.) and sample preparation ahead of the experiment and during the measurement (e.g. environment conditions etc.).

4.3.2 exchange

In X-ray tomography, the 3D arrays representing the most basic version of the data include projections, dark, and white fields. It is **mandatory** that there is at least one dataset named **data** in each exchange group. Most data analysis and plotting programs will primarily focus in this group.

Member	Type	Example/Attributes
name	string dataset	"absorption_tomography"
description	string dataset	"raw absorption tomo"
data	3D dataset	axes: *theta:y:x*
theta	1D dataset	units: "deg"
data_dark	3D dataset	axes: *theta_dark:y:x*
theta_dark	1D dataset	units: "deg"
data_white	3D dataset	axes: *theta_white:y:x*
theta_white	1D dataset	units: "deg"
data_shift_x	relative x shift of data at each angular position	
data_shift_y	relative y shift of data at each angular position	

Table: Exchange Group Members for Tomography

name

Descriptive *name* for **data** dataset. Current types include: absorption_tomography, phase_tomography, dpc_tomography

description

Description.

data

A tomographic data set consists of a series of projections (**data**), dark field (*data_dark*), and white field (*data_white*) images. The dark and white fields must have the same projection image dimensions and can be collected at any time before, after or during the projection data collection. The angular position of the tomographic rotation axis, theta, can be used to keep track of when the dark and white images are collected. These datasets are saved in 3D arrays using, by default, the natural HDF5 order of a multidimensional array (rotation axis, ccd y, ccd x), i.e. with the fastest changing dimension being the last dimension, and the slowest changing dimension being the first dimension. If using the default dimension order, the axes attribute *theta:y:x* can be omitted. The attribute is **mandatory** if the 3D arrays use a different axes order. This could be the case when, for example, the arrays are optimized for sinogram read (= *y:theta:x*). As no units are specified the data is assumed to be in *counts* with the axes (x, y) in pixels.

data_dark, data_white

The dark field and white fields must have the same dimensions as the projection images and can be collected at any time before, during, or after the projection data collection. To specify where dark and white images were taken, specify the axes attribute with "theta_dark:y:x" and "theta_white:y:x" and provide *theta_dark* and *theta_white* vector datasets that specify the rotation angles where they were collected.

theta, theta dark, theta_white

Theta is a vector dataset storing the projection angular positions. If theta is not defined the projections are assumed to be collected at equally spaced angular interval between 0 and 180 degree. The dark field and white fields can be collected at any time before, during, or after the projection data. *theta_dark*, and *theta_white* store the position of the tomographic rotation axis when the corresponding dark and white images are collected. If *theta_dark* and *theta_white* are missing the corresponding *data_dark* and *data_white* are assumed to be collected all at the beginning or at the end of the projection data collection.

data_shift_x, data_shift_y

Data_shift_x and data_shift_y are the vectors storing at each projection angular positions the image relative shift in x and y. These vectors are used in high resolution CT when at each angular position the sample x and y are moved to keep the sample in the field of view based on a pre-calibration of rotary stage runout. If the unit is not defined are assumed to be in pixels.

Attribute

Description and units can be added as attribute to any data, both array or values, inside a data exchange file. If units is omitted default is SI.

Member	Type	Example
description	string attribute	"transmission"
units	string attribute	*counts*

Table: data attributes

4.3.3 *measurement*

This group holds sample and instrument information. These groups are designed to hold relatively static data about the sample and instrument configuration at the time of the measurement. Rapidly changing *positioner* values (aka scan) are better represented in the exchange group dataset.

Member	Type	Example
instrument	group	
sample	group	

Table: Measurement Group Members

instrument

The instrument used to collect this data.

sample

The sample measured.

instrument

The instrument group stores all relevant beamline components status at the beginning of a measurement. While all these fields are optional, if you do intend to include them they should appear within this parentage of groups.

Member	Type	Example
name	string dataset	"XSD/32-ID/TXM"
description	string dataset	"X-ray Microscope"
attenuator	group	
beam_monitor	group	
beam_stop	group	
bertrand_lens	group	
condenser	group	
crl	group	
detection_system	group	
detector	group	
diffuser	group	
flight_tube	group	
interferometer	group	
mirror	group	
monochromator	group	
pin_hole	group	
samplee	group	
shutter	group	
source	group	
slits	group	
table	group	
zone_plate	group	
setup	group	

Table: Instrument Group for Tomography

name

Name of the instrument.

source

The source used by the instrument.

shutter

The shutter(s) used by the instrument.

attenuator

The attenuators that are part of the instrument.

monochromator

The monochromator used by the instrument.

detector

The detectors that compose the instrument.

attenuator

This class describes the beamline attenuator(s) used during data collection. If more than one attenuators are used they will be named as attenuator_1, attenuator_2 etc.

Member	Type	Example
name	string dataset	"Filter Set 1"
description	string dataset	"Al"
thickness	float dataset	1e-3
transmission	float dataset	unit-less
geometry	group	
setup	group	

Table: Attenuator Group Members

name

Name.

description

Description.

thickness

Thickness of attenuator along beam direction.

attenuator_transmission

The nominal amount of the beam that gets through (transmitted intensity)/(incident intensity).

description

Type or composition of attenuator.

beam_monitor

Class describing the beam monitor being used, if there is more than one append _##

Member	Type	Example
name	string dataset	"Beam Monitor"
description	string dataset	"optional"
geometry	group	
setup	group	

Table: Beam Monitor Group Members

beam_stop

Class describing the beam stop being used, if there is more than one append _##

Member	Type	Example
name	string dataset	"Beam Stop"
description	string dataset	"optional"
geometry	group	
setup	group	

Table: Beam Stop Group Members

bertrand_lens

Class describing the Bertrand lens being used, if there is more than one append _##

Member	Type	Example
name	string dataset	"Bertrand Lens"
description	string dataset	"optional"
geometry	group	
setup	group	

Table: Bertrand Lens Group Members

condenser

Class describing the condenser being used, if there is more than one append _##

Member	Type	Example
name	string dataset	"Condenser"
description	string dataset	"optional"
geometry	group	
setup	group	

Table: Condenser Group Members

crl

Class describing the compound refractive lenses being used, if there is more than one append _##

Member	Type	Example
name	string dataset	"CRL"
description	string dataset	"optional"
geometry	group	
setup	group	

Table: CRL Group Members

detection_system

In full field imaging the detector consists of microscope objective and a scintillator screen.

Member	Type	Example
name	string dataset	"Detection 1"
description	string dataset	"Standard microCT"
objective	group	
scintillator	group	

Table: Detection System Group Members

name

Name.

description

Description.

objective_N

List of the visible light objectives mounted between the detector and the scintillator screen.

scintillator

Scintillator screen

detector

This class holds information about the detector used during the experiment. If more than one detector are used they will be all listed as detector_N. In full field imaging the detector consists of a CCD camera, microscope objective and a scintillator screen. Raw data recorded by a detector as well as its position and geometry should be stored in this class.

Member	Type	Example
name	string dataset	“DIMAX 1”
description	string dataset	“description”
manufacturer	string dataset	“CooKe Corporation”
model	string dataset	“pco dimax”
serial_number	string dataset	“1234XW2”
firmware_version	string dataset	“3.7.9”
software_version	string dataset	“1.3.14”
bit_depth	integer	12
pixel_size_x	float	6.7e-6
pixel_size_y	float	6.7e-6
actual_pixel_size_x	float	1.2e-6
actual_pixel_size_y	float	1.2e-6
dimension_x	integer	2048
dimension_y	integer	2048
binning_x	integer	1
binning_y	integer	1
operating_temperature	float	270
exposure_time	float	1.7e-3
delay_time	float	1.7e-3
stabilization_time	float	1.7e-3
frame_rate	integer	2
output_data	string dataset	“/exchange”
roi	group	
counts_per_joule	float	unitless
basis_vectors	float array	length
corner_position	3 floats	length
geometry	group	
setup	group	

Table: Detector Group Members for Tomography

name

Name.

description

Description.

manufacturer

The detector manufacturer.

model

The detector model.

serial_number

The detector serial number .

bit_depth

The detector bit depth.

pixel_size_x, pixel_size_y

Physical detector pixel size (m).

dimension_x, dimension_y

The detector horiz./vertical dimension.

actual_pixel_size_x, actual_pixel_size_y

Actual pixel size on the sample plane.

binning_x, binning_y

If the data are collected binning the detector binning_x and binning_y store the binning factor.

operating_temperature

The detector operating temperature (K).

exposure_time

The detector exposure time (s).

delay_time

Delay time between projections when using a mechanical shutter to reduce radiation damage of the sample (s).

stabilization_time

Time required by the sample to stabilize (s).

frame_rate

The detector frame rate (fps). This parameter is set for fly scan.

roi

The detector selected Region Of Interest (ROI).

counts_per_joule

Number of counts recorded per each joule of energy received by the detector. The number of incident photons can then be calculated by:

basis_vectors

A matrix with the basis vectors of the detector data.

corner_position

The x, y and z coordinates of the corner of the first data element.

geometry

Position and orientation of the center of mass of the detector. This should only be specified for non pixel detectors. For pixel detectors use basis_vectors and corner_position.

diffuser

Class describing the diffuser being used, if there is more than one append _##

Member	Type	Example
name	string dataset	"Diffuser"
description	string dataset	"optional"
geometry	group	
setup	group	

Table: Diffuser Group Members

flight_tube

Class describing the flight tube being used, if there is more than one append _##

Member	Type	Example
name	string dataset	“Flight Tube”
description	string dataset	“optional”
geometry	group	
setup	group	

Table: Flight Tube Group Members

interferometer

This group stores the interferometer parameters.

Member	Type	Example
name	string dataset	“Inter 1”
description	string dataset	“description”
grid_start	float	1.8
grid_end	float	3.51
number_of_grid_periods	int	1
number_of_grid_steps	int	6
geometry	group	
setup	group	

Table: Interferometer Group Members

name

Name.

description

Description.

start_angle

Interferometer start angle.

grid_start

Interferometer grid start angle.

grid_end

Interferometer grid end angle.

grid_position_for_scan

Interferometer grid position for scan.

number_of_grid_steps

Number of grid steps.

mirror

Class describing the mirror being used, if there is more than one append _##

Member	Type	Example
name	string dataset	"M1"
description	string dataset	"optional"
angle	float	"optional"
geometry	group	
setup	group	

Table: Mirror Group Members

monochromator

Define the monochromator used in the instrument.

Member	Type	Example
name	string dataset	"Mono 1"
description	string dataset	"Multilayer"
energy	float dataset	1.602e-15
energy_error	float dataset	1.602e-17
mono_stripe	string dataset	"Ru/C"
geometry	group	
setup	group	

Table: Monochromator Group Members

name

Name.

description

Description.

energy

Peak of the spectrum that the monochromator selects. Since units is not defined this field is in J and corresponds to 10 keV.

energy_error

Standard deviation of the spectrum that the monochromator selects. Since units is not defined this field is in J.

mono_stripe

Type of multilayer coating or crystal.

pin_hole

Class describing the pin hole being used, if there is more than one append _##

Member	Type	Example
name	string dataset	"Pin Hole"
description	string dataset	"optional"
geometry	group	
setup	group	

Table: Pin Hole Group Members

shutter

Class describing the shutter being used.

Member	Type	Example
name	string dataset	"Front End Shutter 1"
description	string dataset	"optional"
status	string dataset	"OPEN"
geometry	group	
setup	group	

Table: Shutter Group Members

name

Name.

description

Description.

status

"OPEN" or "CLOSED"

sample

Class describing the sample stage stack being used.

Member	Type	Example
name	string dataset	"TXM sample stack"
description	string dataset	"optional"
detector_distance	string dataset	"optional"
geometry	group	
setup	group	

Table: Sample stage stack Group Members

source

Class describing the light source being used.

Member	Type	Example
name	string dataset	"APS"
description	float dataset	"optional"
datetime	string dataset (ISO 8601)	"2011-07-15T15:10Z"
beamline	string dataset	"2-BM"
current	float dataset	0.094
energy	float dataset	4.807e-15
pulse_energy	float dataset	1.602e-15
pulse_width	float dataset	15e-11
mode	string dataset	"TOPUP"
beam_intensity_incident	float dataset	55.93
beam_intensity_transmitted	float dataset	100.0
geometry	group	
setup	group	

Table: Source

name

Name.

description

Description.

datetime

Date and time source was measured.

beamline

Name of the beamline.

current

Electron beam current (A).

energy

Characteristic photon energy of the source (J). For an APS bending magnet this is 30 keV or 4.807e-15 J.

pulse_energy

Sum of the energy of all the photons in the pulse (J). pulse_width Duration of the pulse (s).

mode

Beam mode: TOP-UP.

beam_intensity_incident

Incident beam intensity in (photons per s).

beam_intensity_transmitted

Transmitted beam intensity (photons per s).

slists

Class describing the slits being used.

Member	Type	Example
name	string dataset	"A slits"
description	string dataset	"Horizontal Slits"
geometry	group	
setup	group	

Table: Slits Group Members

name

Name.

description

Description.

table

Class describing the zone plate being used, if there is more than one append _##

Member	Type	Example
name	string dataset	"Optical Table"
description	string dataset	"optional"
geometry	group	
setup	group	

Table: Optical Table Group Members

zone_plate

Class describing the zone plate being used, if there is more than one append _##

Member	Type	Example
name	string dataset	"Zone Plate"
description	string dataset	"optional"
geometry	group	
setup	group	

Table: Zone Plate Group Members

roi

Group describing the region of interest (ROI) of the image actually collected, if smaller than the full CCD.

Member	Type	Example
name	string dataset	"ROI 04"
description	string dataset	"center third"
min_x	integer	256
size_x	integer	256
min_y	integer	1792
size_y	integer	1792

Table: ROI Group Members

name

Name.

description

Description.

min_x, min_y

Top Left pixel x and y position.

size_x, size_y

x and y image size.

objective

Group describing the microscope objective lenses used.

Member	Type	Example
name	string dataset	"Lens 01"
description	string dataset	"ZeissAx"
manufacturer	string dataset	"Zeiss"
model	string dataset	"Axioplan"
magnification	float dataset	5
numerical_aperture	float dataset	0.8
geometry	group	
setup	group	

Table: Objective Group Members

name

Name.

description

Description.

manufacturer

Lens manufacturer.

model

Lens model.

magnification

Lens specified magnification.

numerical_aperture

The numerical aperture (N.A.) is a measure of the light-gathering characteristics of the lens.

scintillator

Group describing the visible light scintillator coupled to the CCD camera objective lens.

Member	Type	Example
name	string dataset	"Yag polished"
description	string dataset	"Yag on Yag"
manufacturer	string dataset	"Crytur"
serial_number	string dataset	"12"
scintillating_thickness	float dataset	5e-6
substrate_thickness	float dataset	1e-4
geometry	group	
setup	group	

Table: Scintillator Group Members

name

Scintillator name.

description

Scintillator description.

manufacturer

Scintillator Manufacturer.

serial_number

Scintillator serial number.

scintillating_thickness

Scintillator thickness.

substrate_thickness

Scintillator substrate thickness.

setup

Logging instrument and beamline component setup parameters (static setup values) is not defined by Data Exchange because is specific and different for each instrument and beamline. To capture this information Data Exchange requires to set a *setup* group under each beamline component and leaves each facility free to store what is relevant for each component (list of motor positions etc.). Ideally each component in the instrument list (source, shutter, attenuator etc.) should have included its setup group. For setup values not associated with a specific beamline component a *setup* group in the instrument group should be created.

Member	Type	Example
motor_x	float	-10.107
motor_y	float	-17.900
motor_z	float	-5.950
motor_xx	float	-1.559
motor_zz	float	1.307

sample

This group holds basic information about the sample, its geometry, properties, the sample owner (user) and sample proposal information. While all these fields are optional, if you do intend to include them they should appear within this parentage of groups.

Member	Type	Example
name	string dataset	"cells sample 1"
description	string dataset	"malaria cells"
file_path	string dataset	"/2016-03/tst/"
preparation_date	string dataset (ISO 8601)	"2012-07-31T21:15:22+0600"
chemical_formula	string dataset (abbr. CIF format)	"(Cd 2+)3, 2(H2 O)"
mass	float dataset	0.25
concentration	float dataset	0.4
environment	string dataset	"air"
temperature	float dataset	25.4
temperature_set	float dataset	26.0
pressure	float dataset	101325
thickness	float dataset	0.001
position	string dataset	"2D" APS robot coord.
geometry	group	
experiment	group	
experimenter	group	

Table: Sample Group Members

name

Descriptive name of the sample.

file_path

Directory path where the data were originally saved.

description

Description of the sample.

preparation_date

Date and time the sample was prepared.

chemical_formula

Sample chemical formula using the CIF format.

mass

Mass of the sample.

concentration

Mass/volume.

environment

Sample environment.

temperature

Sample temperature.

temperature_set

Sample temperature set point.

pressure

Sample pressure.

thickness

Sample thickness.

position

Sample position in the sample changer/robot.

geometry

Sample center of mass position and orientation.

experiment

Facility experiment identifiers.

experimenter

Experimenter identifiers.

experiment

This provides references to facility ids for the proposal, scheduled activity, and safety form.

Member	Type	Example
proposal	string dataset	"1234"
activity	string dataset	"9876"
safety	string dataset	"9876"
title	string dataset	"Al 4D dynamic tomo"

Table: Experiment Group Members

proposal

Proposal reference number. For the APS this is the General User
Proposal number.

activity

Proposal scheduler id. For the APS this is the beamline scheduler activity id.

safety

Safety reference document. For the APS this is the Experiment
Safety Approval Form number.

title

Proposal title.

experimenter

Description of a single experimenter. Multiple experimenters can be represented through numbered entries such as experimenter_1, experimenter_2.

Member	Type	Example
name	string dataset	"John Doe"
role	string dataset	"Project PI"
affiliation	string dataset	"University of California, Berkeley"
address	string dataset	"EPS UC Berkeley CA 94720 4767 USA"
phone	string dataset	"+1 123 456 0000"
email	string dataset	"johndoe@berkeley.edu"
facility_user_id	string dataset	"a123456"

Table: Experimenter Group Members

name: User name.

role: User role.

affiliation: User affiliation.

address: User address.

phoen: User phone number.

email: User e-mail address

facility_user_id: User badge number

geometry

The geometry group is common to many of the subgroups under measurement. The intent is to describe the translation and rotation (orientation) of the sample or instrument component relative to some coordinate system. Since we believe it is not possible to determine all possible uses at this time, we leave the precise definition of geometry up to the technique. We do encourage the use of separate translation and orientation subgroups within geometry. As such, we do not describe geometry further here. This class holds the general position and orientation of a component.

Member	Type	Example
translation	group	
orientation	group	

translation

The position of the object with respect to the origin of your coordinate system.

orientation

The rotation of the object with respect to your coordinate system.

translation

This is the description for the general spatial location of a component for tomography.

Member	Type	Example
distances	3 float array dataset	(0, 0.001, 0)

distances

The x, y and z components of the translation of the origin of the object relative to the origin of the global coordinate system (the place where the X-ray beam meets the sample when the sample is first aligned in the beam). If distances does not have the attribute units set then the units are in meters.

orientation

This is the description for the orientation of a component for tomography.

Member	Type	Example
value	6 float array dataset	

value

Dot products between the local and the global unit vectors. Unitless

The orientation information is stored as direction cosines. The direction cosines will be between the local coordinate directions and the global coordinate directions. The unit vectors in both the local and global coordinates are right-handed and orthonormal.

Calling the local unit vectors (x', y',z') and the reference unit vectors (x, y, z) the six numbers will be

$$[x \cdot x, x' \cdot y, x' \cdot z, y' \cdot x, y' \cdot y, y' \cdot z]$$

where

‘ . ‘

is the scalar dot product (cosine of the angle between the unit vectors).

Notice that this corresponds to the first two rows of the rotation matrix that transforms from the global orientation to the local orientation. The third row can be recovered by using the fact that the basis vectors are orthonormal.

4.3.4 *process*

Process is the documentation of the data collection strategy (*acquisition*) steps, all transformations, analyses and interpretations of data performed by a sequence of process functions (*actor*) as well as any sample preparation step done ahead of the experiment and during the measurement (e.g. environment conditions etc.).

Maintaining this history, also called provenance, allows for reproducible data. The Data Exchange format tracks process by allowing each actor to append process information to a process table.

The process table tracks provenance in the execution order as a series of processing steps by appending sequential actor entries in the process table.

Member	Type	Example
name	string dataset	“name”
description	string dataset	“optional”
acquisition	group	
tomo_rec	group	
transfer	group	
table	group	

Table: Process Group Members

name

Descriptive process task.

description

Description of the process task.

acquisition

Logging acquisition parameters (static setup and per-image values) is not defined by Data Exchange because is specific and different for each instrument and beamline. In the table below we present the implementation adopted by the Swiss Light Source and Advanced Photon Source.

Member	Type	Example
name	string dataset	"mosaic"
description	string dataset	"step scan"
output_data	string dataset	"/exchange"
version	string dataset	https://github.com/data_collection_scripts/b9ad87e17
sample_position_x	1D array	Position of the sample axis x for each image collected
sample_position_y	1D array	Position of the sample axis y for each image collected
sample_position_z	1D array	Position of the sample axis z for each image collected
sample_image_shift_x	1D array	Vector containing the shift of the sample axis x at each projection on the detector plane.
sample_image_shift_y	1D array	Vector containing the shift of the sample axis y at each projection on the detector plane.
sample_image_shift_z	1D array	Vector containing the shift of the sample axis z at each projection on the detector plane.
image_theta	1D array	Vector containing the rotary stage angular position read from the encoder at each image.
scan_index	1D array	Vector containin for each image the identifier assigned by beamline controls to each individual series of images or scan.
scan_date	1D array	Vector containin for each image the wall date/time at start of scan in iso 8601.
image_date	1D array	Vector containing the date/time each image was acquired in iso 8601.
time_stamp	1D array	Vector containin for each image the relative time since scan_date
image_number	1D array	Vector containin for each image the the image serial number as assigned by the camera. Unique for each individual scan. Always starts at 0.0
image_exposure_time	1D array	Vector containin for each image the the measured exposure time
image_is_complete	1D array	Vector containin for each image the boolen status of: is any pixel data missing?
image_type	1D array	Vector containin for each image contained in /exchange/data 0 for white, 1 for projection and 2 for dark.
set-up	group	

Table: Acquisition Group Members

name

Descriptive *name* for *acquisition*. Current name include: tomo, interlaced, mosaic.

description

Description.

setup

List of static scan setup values. In the table below we present the implementation adopted by the Swiss Light Source and Advanced Photon Source.

Member	Type	Example
rotation_start_angle	float	0.0
rotation_end_angle	float	180.0
rotation_speed	float	180.0
angular_step	float	0.125
number_of_projections	integer	1441
number_of_whites	integer	100
number_of_darks	integer	32
number_of_inter_whites	integer	1
inner_scan_flag	integer	1
white_frequency	integer	0
sample_in	float	0.0
sample_out	float	4.0

Table: Static Setup Acquisition Group for Tomography

tomo_rec (APS)

The Reconstruction process description group contains metadata required to run a tomography reconstruction. The specific algorithm is described in a separate group under the reconstruction setup group. Here is where to log the algorithm setup parameters. In the case of tomoPy this can simply be the link to the scrip used to run the reconstruction.

Member	Type	Example
name	string dataset	"test rec"
description	string dataset	"optional"
version	string dataset	https://github.com/tomopy_scripts/b9ad87e17
input_data	string dataset	"/exchange"
output_data	string dataset	"/exchange_1"
set_up	group	

Table: Reconstruction Actor Group Members

name

Descriptive actor task.

description

Description of the actor task.

version

Version of the actor task.

If available this can be the repository link to the actor version used
https://github.com/tomopy_scripts/b9ad87e17

input_data, output_data

Origin and destination of the data processed by the reconstruction task.

setup (APS)

Here is where to log the algorithms used by the reconstruction actor.

Member	Type	Example
astra	string dataset	https://github.com/astra/b9ad87e17
tomopy	string dataset	https://github.com/tomopy/c9ad87e77

Table: Reconstruction Setup Group Members

tomo_rec (SLS)

The reconstruction process description group contains metadata required to run a tomography reconstruction. The specific algorithm is described in a separate group under the reconstruction setup group. Here is where to log the algorithm setup parameters.

Member	Type	Example
name	string dataset	"sls rec"
description	string dataset	"optional"
version	string dataset	https://github.com/sls_scripts/b9ad87e17
input_data	string dataset	"/exchange"
output_data	string dataset	"/exchange_1"
set_up_sls	group	

Table: Reconstruction Actor Group Members

name

Descriptive actor task.

description

Description of the actor task.

version

Version of the actor task.

If available this can be the repository link to the actor version used
https://github.com/tomopy_scripts/b9ad87e17

input_data, output_data

Origin and destination of the data processed by the reconstruction task.

setup (SLS)

Here is where to log the algorithms used by the reconstruction actor.

Member	Type	Example
reconstruction_slice_start	int dataset	1000
reconstruction_slice_end	int dataset	1030
rotation_center	Float dataset	1048.50
algorithm-sls	Group	

Table: Reconstruction Setup SLS Group Members

reconstruction_slice_start

First reconstruction slice.

reconstruction_slice_end

Last reconstruction slice.

rotation_center

Center of rotation in pixels.

algorithm

Algorithm group describing reconstruction algorithm parameters.

algorithm (SLS iterative)

The Algorithm group contains information required to run a tomography reconstruction algorithm.

Member	Type	Example
name	string dataset	"SART"
version	string dataset	"1.0"
implementation	string dataset	"GPU"
number_of_nodes	int dataset	16
type	string dataset	"Iterative"
stop_condition	string dataset	"iteration_max"
iteration_max	int dataset	200
projection_threshold	float dataset	
difference_threshold_percent	float dataset	
difference_threshold_value	float dataset	
regularization_type	string dataset	"total_variation"
regularization_parameter	float dataset	
step_size	float dataset	0.3
sampling_step_size	float dataset	0.2

Table: Algorithm Group Members

name

Reconstruction method name: SART, EM, FBP.

version

Algorithm version.

implementation

CPU or GPU.

number_of_nodes

Number of nodes to use on cluster. This parameter is set when the reconstruction is parallelized and run on a cluster.

type

Tomography reconstruction method: iterative.

stop_condition

iteration_max, projection_threshold, difference_threshold_percent, difference_threshold_value.

iteration_max

Maximum number of iterations.

projection_threshold

The threshold of projection difference to stop the iterations as

$$|y - Ax_{\mathrm{n}}| < p$$

difference_threshold_percent

The threshold of reconstruction difference to stop the iterations as

$$|x_{\mathrm{n}+1}|/|x_{\mathrm{n}}| < p$$

difference_threshold_value

The threshold of reconstruction difference to stop the iterations as:

$$|x_{\mathrm{n}+1}| - |x_{\mathrm{n}}| < p$$

regularization_type

total_variation, none.

regularization_parameter

step_size

Step size between iterations in iterative methods

sampling_step_size

Step size used for forward projection calculation in iterative methods.

algorithm (SLS analytic)

The Algorithm group contains information required to run a tomography reconstruction algorithm.

Member	Type	Example
name	string dataset	“gridrec”
version	string dataset	“1.0”
implementation	string dataset	“CPU”
number_of_nodes	int dataset	16
type	string dataset	“analytic”
filter	string dataset	“Parzen”
padding	float dataset	0.50

Table: Algorithm Group Members

name

Reconstruction method name: GridRec.

version

Algorithm version.

implementation

CPU or GPU.

number_of_nodes

Number of nodes to use on cluster. This parameter is set when the reconstruction is parallelized and run on a cluster.

type

Tomography reconstruction method: analytic.

filter

Filter type.

padding

transfer

The transfer process description group contains metadata required to trasfer data from source (data analysis machine) to destination (data distribution server).

Member	Type	Example
name	string dataset	"Globus"
description	string dataset	"data distribution to users"
version	string dataset	https://github.com/globus/b9ad87e17
input_data	string dataset	"gsiftp://host1/path"
output_data	string dataset	"gsiftp://host2/path"
setup	group	

Table: Transfer Actor Group Members

name

Descriptive actor task.

description

Description of the actor task.

version

Version of the actor task.

If available this can be the repository link to the actor version used
https://github.com/globus/b9ad87e17

input_data, output_data

Origin and destination of the data processed by the trasnfer task.

setup

Group containing the specific data transfer protocol paramenters.

table

Scientific users will not generally be expected to maintain data in this group. The expectation is that analysis pipeline tools will automatically record process steps using this group. In addition, it is possible to re-run an analysis using the information provided here.

actor	start_time	end_time	status	message	reference	description
acquisition	21:15:22	21:15:23	FAILED	beamline off line	/process/acquisition	raw data collection
acquisition	21:15:26	21:15:27	FAILED	beamline off line	/process/acquisition	raw data collection
acquisition	21:17:28	22:15:22	SUCCESS	OK	/process/acquisition	raw data collection
tomo_rec	22:30:23	22:50:22	SUCCESS	OK	/process/tomo_rec	reconstruct
transfer			QUEUED		/process/transfer	transfer data to user

Table: Process table to log actors activity

actor

Name of the process in the pipeline stage that is executed at this step.

start_time

Time the process started.

end_time

TIme the process ended.

status

Current process status. May be one of the following: QUEUED, RUNNING, FAILED, or SUCCESS.

message

A process specific message generated by the process. It may be a confirmation that the process was successful, or a detailed error message, for example.

reference

Path to the actor description group. The process description group

contains all metadata to perform the specific process. This reference is simply the HDF5 path within this file of the technique specific process description group. The process description group should contain all parameters necessary to run the process, including the name and version of any external analysis tool used to process the data. It should also contain input and output references that point to the **exchange_N** groups that contain the input and output datasets of the process.

description

Process description.

4.4 X-ray Fluorescence

This section describes extensions and additions to the core Data Exchange format for X-ray Fluorescence. We begin with the extensions to the exchange and instrument groups, and then describe the possible fluorescence data collection schemes and corresponding data structures.

4.4.1 Top level (root)

This node represents the top level of the HDF5 file and holds some general information about the file.

TO BE COMPLETED

4.5 X-ray Photon Correlation Spectroscopy

This section describes extensions and additions to the core Data Exchange format for X-ray Photon Correlation Spectroscopy. We begin with the extensions to the exchange and instrument groups, and then describe the possible XPCS data collection schemes and corresponding data structures.

4.5.1 Top level (root)

This node represents the top level of the HDF5 file and holds some general information about the file.

TO BE COMPLETED

4.6 Install

This section covers the basics of how to download and install DXfile.

Contents:

4.6.1 Installing from source

Clone the DXfile from GitHub repository:

```
git clone https://github.com/data-exchange/dxfile DXfile
```

then:

```
cd DXfile
python setup.py install
```

4.6.2 Installing from Conda/Binstar

First you must have Conda installed, then open a terminal or a command prompt window and run:

```
conda install -c conda-forge dxfile
```

4.6.3 Updating the installation

Data Management is an active project, so we suggest you update your installation frequently. To update the installation run in your terminal:

```
conda update -c conda-forge dxfile
```

For some more information about using Conda, please refer to the docs.

4.7 API reference

DXfile subclasses the h5py module for interacting with Data Exchange files.

DXFile Modules:

4.7.1 dxfile

Subclasses the h5py module for interacting with Data Exchange files.

Functions:

File(*args, **kwargs)	Interact with Data Exchange files.
Entry(**kwargs)	Interact with Data Exchange files.

class dxfile.dxtomo.**Entry**(***kwargs*)

Bases: object

Interact with Data Exchange files.

_entry_definitions(*self*)

Contains the archetypes for Data Exchange file entries.

_generate_classes(*self*)

This method is used to turn the Entry._entry_definitions into generate_classes which can be instantitated for hold data.

class acquisition(***kwargs*)

Bases: object

docstring = 'Tomography specific tag to store dynamic (per image) parameters.'

end_date = {'docstring': 'Date and time measurement ends.', 'units': 'text', 'value': None}

entry_name = 'acquisition'

image_date = {'docstring': 'Vector containing the date/time each image was acquired in iso 8601.', 'units': 'time', 'value': None}

image_exposure_time = {'docstring': 'Vector containin for each image the the measured exposure time in 1e-7 seconds (0.1us)', 'units': None, 'value': None}

image_is_complete = {'docstring': 'Vector containin for each image the boolen status of: is any pixel data missing?', 'units': None, 'value': None}

image_number = {'docstring': 'Vector containin for each image the the image serial number as assigned by the camera. Unique for each individual scan. Always starts at 0.', 'units': None, 'value': None}

image_theta = {'docstring': 'Vector containing the rotary stage angular position read from the encoder at each image.', 'units': 'degree', 'value': None}

image_type = {'docstring': 'Vector containin for each image contained in /exchange/data 0 for white, 1 for projection and 2 for dark', 'units': None, 'value': None}

root = '/process'

sample_image_shift_x = {'docstring': 'Vector containing the shift of the sample axis x at each projection on the detector plane.', 'units': 'pixels', 'value': None}

sample_image_shift_y = {'docstring': 'Vector containing the shift of the sample axis y at each projection on the detector plane.', 'units': 'pixels', 'value': None}

sample_position_x = {'docstring': 'Vector containing the position of the sample axis x at each projection image collection.', 'units': 'mm', 'value': None}

sample_position_y = {'docstring': 'Vector containing the position of the sample axis y at each projection image collection.', 'units': 'mm', 'value': None}

sample_position_z = {'docstring': 'Vector containing the position of the sample axis z at each projection image collection.', 'units': 'mm', 'value': None}

scan_date = {'docstring': 'Vector containing for each image the wall date/time at start of scan in iso 8601.', 'units': None, 'value': None}

scan_index = {'docstring': 'Vector containin for each image the identifier assigned by beamline controls to each individual series of images or scan.', 'units': None, 'value': None}

shutter = {'docstring': 'Vector containin for each image the beamline shutter status: 0 for closed, 1 for open', 'units': None, 'value': None}

start_date = {'docstring': 'Date and time measurement starts.', 'units': 'text', 'value': None}

time_stamp = {'docstring': 'Vector containin for each image the relative time since scan_date in 1e-7 seconds.', 'units': None, 'value': None}

acquisition_setup

alias of setup

class attenuator(**kwargs)

Bases: object

description = {'docstring': 'Description or composition of attenuator.', 'units': 'text', 'value': None}

docstring = 'X-ray beam attenuator.'

entry_name = 'attenuator'

name = {'docstring': 'Name of the attenuator.', 'units': 'text', 'value': None}

root = '/measurement/instrument'

thickness = {'docstring': 'Thickness of attenuator along beam direction.', 'units': 'm', 'value': None}

transmission = {'docstring': 'The nominal amount of the beam that gets through (transmitted intensity)/(incident intensity)', 'units': 'None', 'value': None}

data

alias of

class detector(***kwargs*)

Bases: object

actual_pixel_size_x = {'docstring': 'Pixel size on the sample plane (m).', 'units': 'm', 'value': None}

actual_pixel_size_y = {'docstring': 'Pixel size on the sample plane (m).', 'units': 'm', 'value': None}

basis_vectors = {'docstring': 'A matrix with the basis vectors of the detector data.', 'units': 'fps', 'value': None}

binning_x = {'docstring': 'If the data are collected binning the detector x binning and y binning store the binning factor.', 'units': 'pixels', 'value': None}

binning_y = {'docstring': 'If the data are collected binning the detector x binning and y binning store the binning factor.', 'units': 'dimensionless', 'value': None}

bit_depth = {'docstring': 'The detector ADC bit depth.', 'units': 'dimensionless', 'value': None}

corner_position = {'docstring': 'The x, y and z coordinates of the corner of the first data element.', 'units': 'fps', 'value': None}

counts_per_joule = {'docstring': 'Number of counts recorded per each joule of energy received by the detector', 'units': 'counts', 'value': None}

delay_time = {'docstring': 'Detector delay time (s). This is used in combination with a mechanical shutter.', 'units': 's', 'value': None}

description = {'docstring': 'Description of the detector', 'units': 'text', 'value': None}

dimension_x = {'docstring': 'The detector horiz. dimension.', 'units': 'pixels', 'value': None}

dimension_y = {'docstring': 'The detector vertical dimension.', 'units': 'text', 'value': None}

docstring = 'X-ray detector.'

entry_name = 'detector'

exposure_time = {'docstring': 'The set detector exposure time (s).', 'units': 's', 'value': None}

firmware_version = {'docstring': 'The detector firmware version.', 'units': 'text', 'value': None}

frame_rate = {'docstring': 'The detector frame rate (fps).', 'units': 'fps', 'value': None}

manufacturer = {'docstring': 'The detector manufacturer.', 'units': 'text', 'value': None}

model = {'docstring': 'The detector model', 'units': 'text', 'value': None}

name = {'docstring': 'Name of the detector.', 'units': 'text', 'value': None}

operating_temperature = {'docstring': 'The detector operating temperature (K).', 'units': 'dimensionless', 'value': None}

output_data = {'docstring': 'String HDF5 path to the exchange group where the detector output data is located.', 'units': 'text', 'value': None}

pixel_size_x = {'docstring': 'Physical detector pixel size (m).', 'units': 'm', 'value': None}

pixel_size_y = {'docstring': 'Physical detector pixel size (m).', 'units': 'm', 'value': None}

root = '/measurement/instrument'

serial_number = {'docstring': 'The detector serial number.', 'units': 'text', 'value': None}

shutter_mode = {'docstring': 'The detector shutter mode: global, rolling etc.', 'units': 'text', 'value': None}

software_version = {'docstring': 'The detector software version.', 'units': 'text', 'value': None}

stabilization_time = {'docstring': 'Detector delay time (s). This is used during stop and go data collection to allow the sample to stabilize.', 'units': 's', 'value': None}

exchange

alias of

class experiment(***kwargs*)

Bases: object

activity = {'docstring': 'Proposal scheduler id. For the APS this is the beamline scheduler activity id.', 'units': 'text', 'value': None}

docstring = 'This provides references to facility ids for the proposal, scheduled activity, and safety form.'

entry_name = 'experiment'

proposal = {'docstring': 'Proposal reference number. For the APS this is the General User Proposal number.', 'units': 'text', 'value': None}

root = '/measurement/sample'

safety = {'docstring': 'Safety reference document. For the APS this is the Experiment Safety Approval Form number.', 'units': 'text', 'value': None}

title = {'docstring': 'Experiment title. For the APS this is the proposal title assigned by the user.', 'units': 'text', 'value': None}

class experimenter(***kwargs*)

Bases: object

address = {'docstring': 'User address.', 'units': 'text', 'value': None}

affiliation = {'docstring': 'User affiliation.', 'units': 'text', 'value': None}

docstring = 'Description of a single experimenter.'

email = {'docstring': 'User email address.', 'units': 'text', 'value': None}

entry_name = 'experimenter'

facility_user_id = {'docstring': 'User badge number.', 'units': 'text', 'value': None}

name = {'docstring': 'User name.', 'units': 'text', 'value': None}

phone = {'docstring': 'User phone number.', 'units': 'text', 'value': None}

role = {'docstring': 'User role.', 'units': 'text', 'value': None}

root = '/measurement/sample'

class instrument(***kwargs*)

Bases: object

comment = {'docstring': 'comment', 'units': 'text', 'value': None}

docstring = 'All relevant beamline components status at the beginning of a measurement'

entry_name = 'instrument'

name = {'docstring': 'Name of the instrument.', 'units': 'text', 'value': None}

root = '/measurement'

class interferometer(***kwargs*)

Bases: object

description = {'docstring': 'Description of the interferometer.', 'units': 'text', 'value': None}

docstring = 'interferometer name'

entry_name = 'interferometer'

name = {'docstring': 'Descriptive name of the interferometer.', 'units': 'text', 'value': None}

root = '/measurement/instrument/'

interferometer_setup

alias of setup

class mirror(**kwargs)

Bases: object

angle = {'docstring': 'Mirror incident angle', 'units': 'rad', 'value': None}

description = {'docstring': 'Description of the mirror', 'units': 'text', 'value': None}

docstring = 'X-ray beam mirror.'

entry_name = 'mirror'

name = {'docstring': 'Name of the mirror.', 'units': 'text', 'value': None}

root = '/measurement/instrument'

class monochromator(**kwargs)

Bases: object

description = {'docstring': 'Description of the monochromator', 'units': 'text', 'value': None}

docstring = 'X-ray beam monochromator.'

energy = {'docstring': 'Peak of the spectrum that the monochromator selects. When units is not defined this field is in J', 'units': 'J', 'value': None}

energy_error = {'docstring': 'Standard deviation of the spectrum that the monochromator selects. When units is not defined this field is in J.', 'units': 'J', 'value': None}

entry_name = 'monochromator'

mono_stripe = {'docstring': 'Type of multilayer coating or crystal.', 'units': 'text', 'value': None}

name = {'docstring': 'Name of the monochromator.', 'units': 'text', 'value': None}

root = '/measurement/instrument'

class objective(**kwargs)

Bases: object

description = {'docstring': 'Lens description', 'units': 'text', 'value': None}

docstring = 'microscope objective lenses used.'

entry_name = 'objective'

magnification = {'docstring': 'Lens specified magnification', 'units': 'dimensionless', 'value': None}

manufacturer = {'docstring': 'Lens manufacturer', 'units': 'text', 'value': None}

model = {'docstring': 'Lens model.', 'units': 'text', 'value': None}

name = {'docstring': 'Lens name', 'units': 'text', 'value': None}

numerical_aperture = {'docstring': 'The numerical aperture (N.A.) is a measure of the light-gathering characteristics of the lens.', 'units': 'dimensionless', 'value': None}

root = '/measurement/instrument/detection_system'

process

alias of

class roi(***kwargs*)

Bases: object

description = {'docstring': 'ROI description', 'units': 'text', 'value': None}

docstring = 'region of interest (ROI) of the image actually collected, if smaller than the full CCD.'

entry_name = 'roi'

min_x = {'docstring': 'Top left x pixel position', 'units': 'pixels', 'value': None}

min_y = {'docstring': 'Top left y pixel position', 'units': 'pixels', 'value': None}

name = {'docstring': 'ROI name', 'units': 'text', 'value': None}

root = '/measurement/instrument/detector'

size_x = {'docstring': 'Horizontal image size', 'units': 'pixels', 'value': None}

size_y = {'docstring': 'Vertical image size', 'units': 'pixels', 'value': None}

class sample(***kwargs*)

Bases: object

chemical_formula = {'docstring': 'Sample chemical formula using the CIF format.', 'units': 'text', 'value': None}

comment = {'docstring': 'comment', 'units': 'text', 'value': None}

concentration = {'docstring': 'Mass/volume.', 'units': 'kgm^-3', 'value': None}

description = {'docstring': 'Description of the sample.', 'units': 'text', 'value': None}

docstring = 'The sample measured.'

entry_name = 'sample'

environment = {'docstring': 'Sample environment.', 'units': 'text', 'value': None}

fatigue_cycle = {'docstring': 'Sample fatigue cycles.', 'units': None, 'value': None}

mass = {'docstring': 'Mass of the sample.', 'units': 'kg', 'value': None}

name = {'docstring': 'Descriptive name of the sample.', 'units': 'text', 'value': None}

preparation_date = {'docstring': 'Date and time the sample was prepared.', 'units': 'text', 'value': None}

pressure = {'docstring': 'Sample pressure.', 'units': 'kPa', 'value': None}

root = '/measurement'

temperature = {'docstring': 'Sample temperature.', 'units': 'kelvin', 'value': None}

temperature_set = {'docstring': 'Sample temperature set point.', 'units': 'kelvin', 'value': None}

thickness = {'docstring': 'Sample thickness.', 'units': 'm', 'value': None}

tray = {'docstring': 'Sample position in the sample changer/robot.', 'units': 'text', 'value': None}

sample_stack

alias of *sample*

sample_stack_setup

alias of setup

class scintillator(***kwargs*)

Bases: object

description = {'docstring': 'Scintillator description', 'units': 'text', 'value': None}

docstring = 'scintillator used.'

entry_name = 'scintillator'

manufacturer = {'docstring': 'Scintillator Manufacturer.', 'units': 'text', 'value': None}

name = {'docstring': 'Scintillator name', 'units': 'text', 'value': None}

root = '/measurement/instrument/detection_system'

scintillating_thickness = {'docstring': 'Scintillator thickness.', 'units': 'm', 'value': None}

serial_number = {'docstring': 'Scintillator serial number.', 'units': 'text', 'value': None}

substrate_thickness = {'docstring': 'Scintillator substrate thickness.', 'units': 'm', 'value': None}

class source(***kwargs*)

Bases: object

beam_intensity_incident = {'docstring': 'Incident beam intensity in (photons per s).', 'units': 'phs^-1', 'value': None}

beam_intensity_transmitted = {'docstring': 'Transmitted beam intensity (photons per s).', 'units': 'phs^-1', 'value': None}

beamline = {'docstring': 'Name of the beamline.', 'units': 'text', 'value': None}

current = {'docstring': 'Electron beam current (A).', 'units': 'A', 'value': None}

datetime = {'docstring': 'Date and time source was measured.', 'units': 'text', 'value': None}

docstring = 'The light source being used'

energy = {'docstring': 'Characteristic photon energy of the source (J). For an APS bending magnet this is 30 keV or 4.807e-15 J.', 'units': 'J', 'value': None}

entry_name = 'source'

mode = {'docstring': 'top-up', 'units': 'text', 'value': None}

name = {'docstring': 'Name of the facility.', 'units': 'text', 'value': None}

pulse_energy = {'docstring': 'Sum of the energy of all the photons in the pulse (J).', 'units': 'J', 'value': None}

pulse_width = {'docstring': 'Duration of the pulse (s).', 'units': 's', 'value': None}

root = '/measurement/instrument'

class dxfile.dxtomo.**File**(**args: Any*, ***kwargs: Any*)

Bases: File

Interact with Data Exchange files.

create_top_level_group(self, group_name):

Helper function for creating a top level group which will update the implements group automagically.

add_entry(self, dexen_ob, overwrite=False):

This method is used to parse DataExchangeEntry objects and add them to the DataExchangeFile.

add_entry(*dexen_ob*, *overwrite=False*)

This method is used to parse DataExchangeEntry objects and add them to the DataExchangeFile.

`create_top_level_group`(*group_name*)

Create a group in the file root and updates the `implements` group accordingly. This method should ALWAYS be used to create groups in the file root.

4.8 Examples

4.8.1 Tomographic data files

For a repository of experimental and simulated data sets using the the Data Exchange file format (DXfile) [B5], please check TomoBank [B3].

For reading tomography files formatted in different ways, please go check the DXchange package. There are various examples and demonstration scripts about how to load your datasets.

4.8.2 Area Detector

At synchrotron facilities using the EPICS [B1] software for area detectors [B12] with the NDFileHDF5 plugin [B11], is possible to directly save DXfile by properly configure the detector and the HDF schema attribute files. Below are examples on how this has been implemented at various facilities.

Advanced Photon Source

At synchrotron facilities using the EPICS [B1] software for area detectors [B12] with the NDFileHDF5 plugin [B11], is possible to save Data Exchange files by properly configure the detector and the HDF schema attribute files to obtain txm.h5

Here are the templates in use at the Advanced Photon Source:

- **2-BM-A/B**
  - Micro Tomography Instrument: `hdf_schema.xml` plus `A_station_detector_attributes.xml` or `B_station_detector_attributes.xml`
- **6-BM**
  - Micro Tomography Instrument: `TomoScanLayout.xml` and `TomoScanDetectorAttributes.xml`
- **7-BM**
  - Fast Micro Tomography Instrument: `mct_hdf_schema.xml` and `mct_detector_attribute.xml`
- **13-BM**
  - Micro-tomography system at 13-BM-D using PG cameras: `tomoLayout.xml` and `tomoDetectorAttributes.xml`
- **32-ID**
  - Transmission X-Ray Microscope: `hdf_schema.xml` and `txm_detector_attribute.xml`.
  - Micro Tomography Instrument: `mct_hdf_schema.xml` and `mct_detector_attribute.xml`.

XML

To check that the areadetector attributes and layout XML contain a set of matching names run:

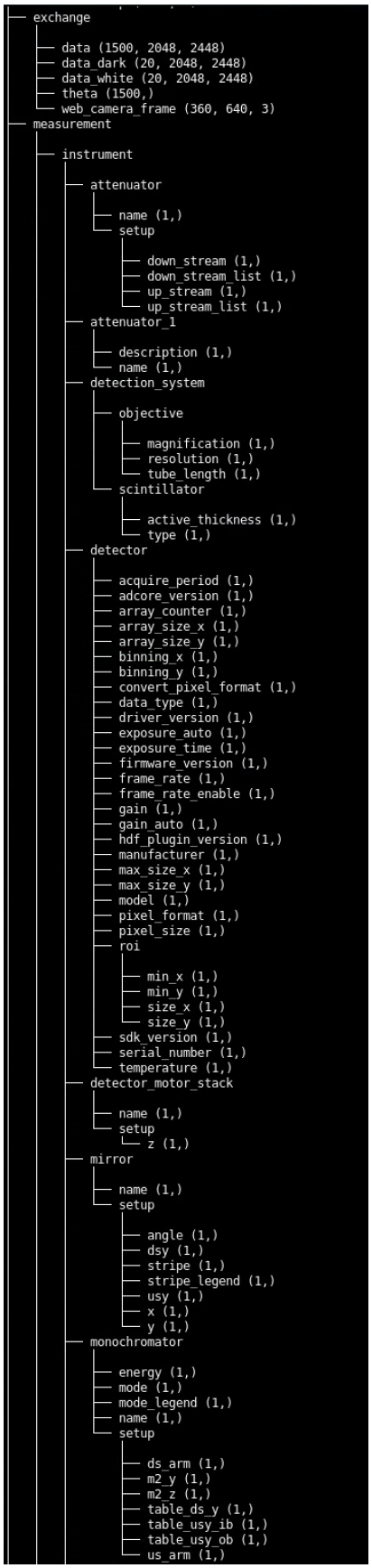
```
$ bash
usertxm@txmtwo$ grep -oP 'name=\"\K[^\"]+' TomoScanDetectorAttributes.xml | while read -
↪r line ; do echo -n "$line " ; grep -q "$line" TomoScanLayout.xml && echo true || echo␣
↪false ; done | grep false
usertxm@txmtwo$ grep -oP 'ndattribute=\"\K[^\"]+' TomoScanLayout.xml | while read -r␣
↪line; do echo -n "$line "; grep -q "$line" TomoScanDetectorAttributes.xml && echo true␣
↪|| echo false ; done |grep false
```

To visualize the meta data and the layout of the hdf file use meta cli

View the hdf tree

To view the data tree contained in a generic hdf file:

```
$ meta tree --file-name data/base_file_name_001.h5
```


exchange
data (1500, 2048, 2448)
data_dark (20, 2048, 2448)
data_white (20, 2048, 2448)
theta (1500,)
web_camera_frame (360, 640, 3)
measurement
instrument
attenuator
name (1,)
setup
down_stream (1,)
down_stream_list (1,)
up_stream (1,)
up_stream_list (1,)
attenuator_1
description (1,)
name (1,)
detection_system
objective
magnification (1,)
resolution (1,)
tube_length (1,)
scintillator
active_thickness (1,)
type (1,)
detector
acquire_period (1,)
adcore_version (1,)
array_counter (1,)
array_size_x (1,)
array_size_y (1,)
binning_x (1,)
binning_y (1,)
convert_pixel_format (1,)
data_type (1,)
driver_version (1,)
exposure_auto (1,)
exposure_time (1,)
firmware_version (1,)
frame_rate (1,)
frame_rate_enable (1,)
gain (1,)
gain_auto (1,)
hdf_plugin_version (1,)
manufacturer (1,)
max_size_x (1,)
max_size_y (1,)
model (1,)
pixel_format (1,)
pixel_size (1,)
roi
min_x (1,)
min_y (1,)
size_x (1,)
size_y (1,)
sdk_version (1,)
serial_number (1,)
temperature (1,)
detector_motor_stack
name (1,)
setup
z (1,)
mirror
name (1,)
setup
angle (1,)
dsy (1,)
stripe (1,)
stripe_legend (1,)
usy (1,)
x (1,)
y (1,)
monochromator
energy (1,)
mode (1,)
mode_legend (1,)
name (1,)
setup
ds_arm (1,)
m2_y (1,)
m2_z (1,)
table_ds_y (1,)
table_usy_ib (1,)
table_usy_ob (1,)
us_arm (1,)

View the meta data

To view the meta data contained in a generic hdf file:

```
$ meta show --file-name data/base_file_name_001.h5
```

View a subset meta data

To view a subset of the meta data contained in a generic hdf file:

```
$ meta show --file-name data/base_file_name_001.h5 --key energy
```

Replace an hdf entry value

To replace the value of an entry:

```
$ meta set --file-name data/base_file_name_001.h5 --key /process/acquisition/
↪rotation/rotation_start --value 10
```

Meta data rst table

To generate a meta data rst table compatible with sphinx/readthedocs:

```
$ meta docs --file-name data/base_file_name_001.h5
2022-02-09 12:30:16,983 - Please copy/paste the content of ./log_2020-05.rst in your rst␣
↪docs file
```

The content of the generated rst file will publish in a sphinx/readthedocs document as:

2022-05

decarlo

	value	unit
000/measurement/instrument/monochromator/energy	30.0	keV
000/measurement/instrument/sample_motor_stack/setup/x	0.0	mm
000/measurement/instrument/sample_motor_stack/setup/y	0.4000116247000278	mm
000/measurement/sample/experimenter/email	decarlof@gmail.com	

Note: when using the **docs** option –file-name can be also a folder, e.g. –file-name data/ in this case all hdf files in the folder will be processed.

to list of all available options:

```
$ meta  -h
```

4.8.3 Python

This section contains python code examples on how to generate and access the meta-data of a DXfile.

Utility

This section contains links to python code examples to generate a `simple.py` and a `full.py` data-exchange file using the DXfile class.

`dump_dxfile.py` allows to print the list of Groups/Datasets names and values contained in a DataExchange hdf file. Using > is possible to save this script output to a text file. The script has also an option to convert a DataExchange file into a stack of tiff files.

Usage:

```
python dump_dxfile.py -h
usage: dump_dxfile.py [-h] [--tiff] fname


positional arguments:
        fname       directory containing multiple dxfiles or a single DataExchange
            file: /data/ or /data/sample.h5

optional arguments:
        -h, --help      show this help message and exit
        --tiff          convert a single DataExchange file to a stack of tiff files
```

Example:

```
python dump_dxfile.py test01/ | grep "start_date"

    test01/001_test.h5 /process/acquisition/start_date = ['May 29, 2019 19:20:21']
    test01/002_test.h5 /process/acquisition/start_date = ['May 29, 2019 19:23:26']
    test01/003_test.h5 /process/acquisition/start_date = ['May 29, 2019 19:26:51']
    test01/004_test.h5 /process/acquisition/start_date = ['May 29, 2019 19:30:17']
    test01/005_test.h5 /process/acquisition/start_date = ['May 29, 2019 19:33:42']
    test01/006_test.h5 /process/acquisition/start_date = ['May 29, 2019 19:37:07']
    ...


python dump_dxfile.py test01/ | grep "data array"
    data array test01/001_test.h5 /exchange/data (1500, 2048, 2448)
    data array test01/002_test.h5 /exchange/data (1500, 2048, 2448)
    data array test01/003_test.h5 /exchange/data (1500, 2048, 2448)
    data array test01/004_test.h5 /exchange/data (1500, 2048, 2448)
    data array test01/005_test.h5 /exchange/data (1500, 2048, 2448)
    data array test01/006_test.h5 /exchange/data (1500, 2048, 2448)
    ...

    python dump_dxfile.py /tomobank/tomo_00001.h5 > experiment_log.txt
    python dump_dxfile.py /tomobank/tomo_00001.h5 --tiff
```

4.9 Credits

4.9.1 Citations

We kindly request that you cite the following article [A1] if you use DXfile.

4.9.2 References

4.10 Appendix

4.10.1 Default units for Data Exchange entries

The default units for Data Exchange entries follow the CXI entries definition, i.e. are SI based units unless the "units" attribute is specified. Data Exchange prefers to use the default SI based units whenever possible.

Quantity	Units	Abbreviation
length	meter	m
mass	kilogram	kg
time	second	s
electric current	ampere	A
temperature	kelvin	K
amount of substance	mole	mol
luminous intensity	candela	cd
frequency	hertz	Hz
force	newton	N
pressure	pascal	Pa
energy	joule	J
power	watt	W
electric potential	volt	V
capacitance	farad	F
electric resistance	ohm	Omega
absorbed dose	gray	Gy
area	square meter	m^2
volume	cubic meter	m^3

Table: SI (and common derived) base units for different quantities

Exceptions

Angles are always defined in degrees *not* in radians and use the abbreviation "degree".

Times and Dates

Times and Dates are always specified according to the ISO 8601. This means for example "1996-07-31T21:15:22+0600". Note the "T" separating the data from the time and the "+0600" timezone specification.

4.10.2 Geometry

Coordinate System

The Data Exchange uses the same CXI coordinate system. This is a right handed system with the z axis parallel to the X-ray beam, with the positive z direction pointing away from the light source, in the downstream direction. The y axis is vertical with the positive direction pointing up, while the x axis is horizontal completing the right handed system (see Fig. [fig:CoordSystem]). The origin of the coordinate system is defined by the point where the X-ray beam meets the sample.

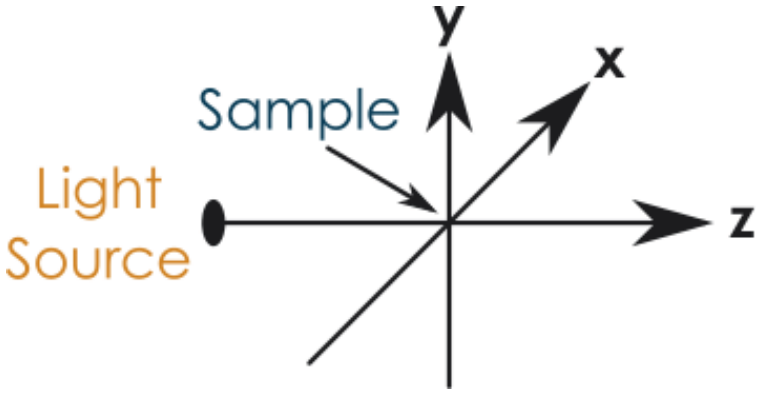


Fig. 9: The coordinate system used by CXI. The intersection of the X-ray beam with the sample define the origin of the system. The z axis is parallel to the beam and points downstream.

The local coordinate system of objects

For many detectors their location and orientation is crucial to interpret results. Translations and rotations are used to define the absolute position of each object. But to be able to apply these transformations we need to know what is the origin of the local coordinate system of each object. Unless otherwise specified the origin should be assumed to be the geometrical center of the object in question. The default orientation of the object should have the longest axis of the object aligned with the x axis, the second longest with the y axis and the shortest with the z axis.

4.11 Indices and tables

- genindex
- modindex
- search

BIBLIOGRAPHY

[A1] De Carlo F, Gursoy D, Marone F, Rivers M, Parkinson YD, Khan F, Schwarz N, Vine DJ, Vogt S, Gleber SC, Narayanan S, Newville M, Lanzirotti T, Sun Y, Hong YP, and Jacobsen C. Scientific data exchange: a schema for hdf5-based storage of raw and analyzed data. *Journal of Synchrotron Radiation*, 21(6):1224–1230, 2014.

[B1] The EPICS control system. http://www.aps.anl.gov/epics/. Accessed: 2016-03-12.

[B2] The UDUNITS at UNIDATA. http://www.unidata.ucar.edu/software/udunits/. Accessed: 2016-03-12.

[B3] Francesco De Carlo, Doga Gursoy, Daniel Jackson Ching, Kees Joost Batenburg, Wolfgang Ludwig, Lucia Mancini, Federica Marone, Rajmund Mokso, Daniel M. Pelt, Jan Sijbers, and Mark Rivers. Tomobank: a tomographic data repository for computational x-ray science. *Measurement Science and Technology*, 2017. URL: https://doi.org/10.1088/1361-6501/aa9c19.

[B4] Gürsoy D, De Carlo F, Xiao X, and Jacobsen C. Tomopy: a framework for the analysis of synchrotron tomographic data. *Journal of Synchrotron Radiation*, 21(5):1188–1193, 2014.

[B5] De Carlo F, Gursoy D, Marone F, Rivers M, Parkinson YD, Khan F, Schwarz N, Vine DJ, Vogt S, Gleber SC, Narayanan S, Newville M, Lanzirotti T, Sun Y, Hong YP, and Jacobsen C. Scientific data exchange: a schema for hdf5-based storage of raw and analyzed data. *Journal of Synchrotron Radiation*, 21(6):1224–1230, 2014.

[B6] The HDF Group. The HDF Dump. http://www.hdfgroup.org/HDF5/doc/RM/Tools.html#Tools-Dump. Accessed: 2016-03-12.

[B7] The HDF Group. The HDF File Format. http://www.hdfgroup.org/HDF5. Accessed: 2016-03-12.

[B8] The HDF Group. The HDF viewer. http://www.hdfgroup.org/hdf-java-html/hdfview. Accessed: 2016-03-12.

[B9] Filipe R. N. C. Maia. The CXI File Format. https://github.com/FilipeMaia/CXI/raw/master/cxi_file_format.pdf. Accessed: 2016-03-12.

[B10] Filipe R. N. C. Maia. The Coherent X-ray Imaging Data Bank. http://cxidb.org/cxi.html. Accessed: 2016-03-12.

[B11] Ulrik Pedersen, Arthur Glowacki, Alan Greer, and Mark Rivers. Area Detector HDF plugin. http://cars.uchicago.edu/software/epics/NDFileHDF5.html. Accessed: 2016-03-12.

[B12] Mark Rivers. Area Detector. http://cars9.uchicago.edu/software/epics/areaDetector.html. Accessed: 2016-03-12.

PYTHON MODULE INDEX

d

INDEX

Symbols

A

B

C

D

E

F

I

M

N

O

P

R

S

T