The Scientific Data Exchange Reference Guide

Includes implementation for x-ray tomography

http://www.aps.anl.gov/DataExchange

Version 0.9.0

May 3, 2012

Table 1: Version history

Version	Date	Notes
0.9.0	May 3, 2012	saunders: Document ready for final review prior to version 1.0.0

table:SI

Contents

1 Introduction

This document is a complete reference to the Data Exchange file format, including documented implementations for various beamline techniques. Briefly, Data Exchange is a set of guidelines for storing scientific data and metadata in a Hierarchical Data Format 5 (HDF5) file (http://www.hdfgroup.org/HDF5).

This reference guide describes the basic design principles, examples of their application, a core reference for guidelines common to most uses, and coding examples. The guide ends with a section devoted to each known implementation for a particular beamline technique.

2 The Design of Data Exchange

For various reasons, many x-ray techniques developed at synchrotron facilities around the world are unable to directly compare results due to their inability to exchange data and software tools. The aim of Data Exchange is to define a simple file format offering few basic rules and allowing each community to extend and add technique specific components. The goal is to provide extensibility in defining data, meta data and provenance information in a simple way that can be easily adopted by various x-ray techniques.

The Data Exchange format is implemented using Hierarchical Data Format 5 (HDF5), which offers platform-independent binary data storage with optional compression, hierarchical data ordering, and self-describing tags so that one can examine a HDF5 file's contents with no knowledge of how the file writing program was coded.

The aim and the scope of Data Exchange is very similar to the Coherent X-ray Imaging Data Bank file format (CXI), so whenever possible we will use the same conventions, name tags, and reference system. This document is using a similar diagram definition set by Filipe R. N. C. Maia in "CXI file format" (`http://cxidb.org/cxi.html`), with a few minor modifications for Data Exchange definitions.

The core principle of Data Exchange is that it must be simple enough that it is not necessary to use a support library beyond core HDF5. The simplicity of Data Exchange makes it easy for anyone to either look at an example file using h5dump or HDFView, or to look at example code in language X, and then create their own read and write routines in language Y.

The simplest Data Exchange file provides information sufficient to share a multidimensional data array. In this simplest form, Data Exchange implements only one "exchange" group. The "exchange" definition is designed to allow for simple exchange of images, spectra, and other forms of beamline detector data with a minimum of fields. This definition is essentially a technique-agnostic format for exchanging data with others. Data Exchange is also designed to be extended to include technique-specific data and metadata. This is achieved by providing optional, but clearly defined, metadata components to the base definition.

2.1 HDF5

The HDF5 format is the basis of the Data Exchange format. Data Exchange, like CXI, is not a completely new file format, but simply a set of rules designed to create HDF5 files with a common structure to allow a uniform and consistent interpretation of such files.

HDF5 was chosen as the basis because it is a widely used high performance scientific data format which many programs can already, at least partially, read and write. It also brings with it the almost automatic fulfillment of the Data Exchange requirements, i.e. simplicity, flexibility and extensibility. HDF5 version 1.8 or higher is required as previous versions don't support all features required by Data Exchange.

2.2 Data types

Data Exchange uses the same CXI convention for data types as defined at `http://cxidb.org/cxi.html` using HDF5 native datatypes. The data should be saved in the same format as it was created/acquired. For example CCD images acquired as 16 bit integers should be saved using the `H5T_NATIVE_SHORT` HDF5 type. In this way all cross platform big-little endian issues reading and writing files are eliminated.

`DataExchange`

2.3 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.

tab:genrules

The general structure of a Data Exchange file is shown in table 2. 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 "provenance" groups are also defined. Beyond this, additional groups may be added to meet individual needs, with guidelines suggesting the best structure.

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.

tab:genrules

Table 2: Data Exchange Top Level Members

Member	Type	Example
implements	string dataset	"exchange:measurement:provenance"
exchange	group	
`measurement`	group	
`provenance`	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 `provenance`, in which case the implements string would be: "*exchange*: `measurement`: `provenance`"

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.

`provenance`

Optional group containing information about the status of each processing step.

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 A.1 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 Data Exchange Core Reference section.

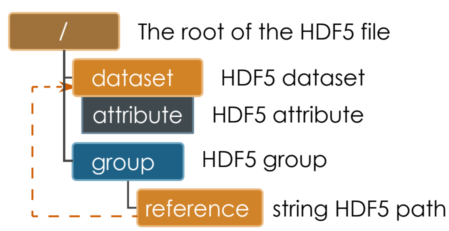
3 Data Exchange by Example

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 4. Technique specific extensions to the core structure can be found starting with Section 5

3.1 Diagram color code

All the diagrams in this section follow the color conventions shown in Figure 1. 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

DiagramColorCode



ig:DiagramColorCode

Figure 1: Explanation of the color code used in the diagrams

3.2 A minimal Data Exchange file for imaging

Figures 2 shows a diagram of a minimal Data Exchange file to store a single projection image. As no units or axes are specified, the data is assumed to be in "counts" with the axes (x, y) in pixels.

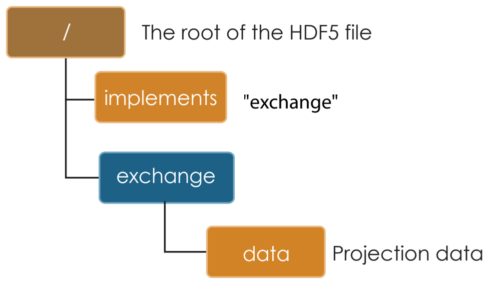


fig:Minimal1

Figure 2: Diagram of a minimal Data Exchange file for a single image.

3.3 Storing and describing a multidimensional dataset

A multidimensional dataset may be stored with no units or axes attributes. In this minimal case, the units are assumed to be "count". Any other interpretation necessarily involves an implicit agreement between the file writer and reader. If there is no axis attribute, each axis is simply a linear integer dimension scale typically interpreted as pixels.

To describe the dataset further and provide for automatic interpretation, a units attribute should be added. Data Exchange recommends using SI units as described in Appendix A.1. The description attribute may be used to characterize the dataset further. An axes attribute may be added containing colon separated names of additional datasets called dimension scales. For example, an axes attribute of "theta:y:x" combined with an additional one-dimensional dataset called theta containing the rotation angles of each projection, provides a complete description. Note that it is not necessary to include x and y datasets since those are assumed to be a linear integer scale (pixels). For other cases, it may be necessary to include a one-dim dataset to describe each axis.

section:minimalTomo

3.4 Storing projections, dark fields, and white fields

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 Figure 3 and 4 using, by default, the natural HDF5 order of the 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 `axes` 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 (`axes` = ”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.

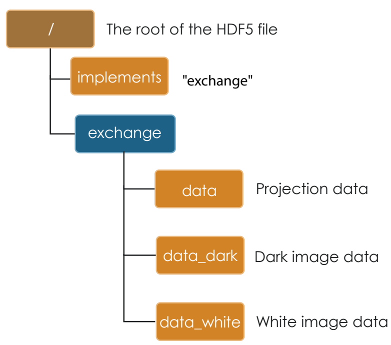


Figure 3: Diagram of a minimal Data Exchange file for a single tomographic data set including raw projections, dark, and white fields.

fig:MinimalTomo0

3.5 A typical Data Exchange file for tomography

A series of tomographic data sets are typically collected changing the instrument status (energy, detector or optics position) or changing the sample status (position, environment etc.). Figure 5, 6 and 7 show the content of files changing the sample temperature, the x-ray source energy and detector-sample distance.

fig:MinimalTomo4

3.5.1 Sample Temperature Scan

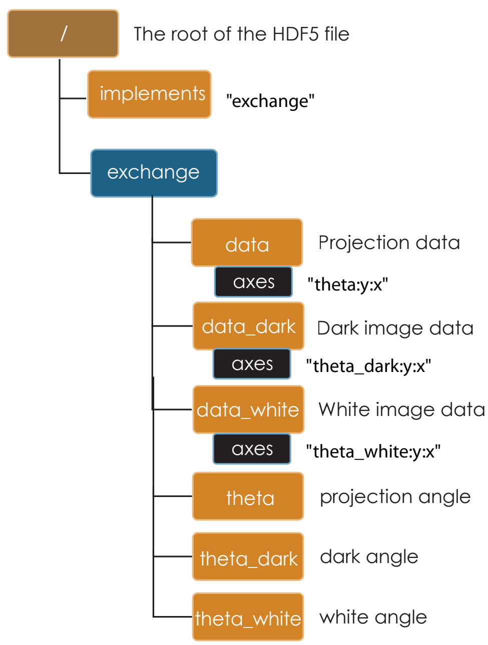


Figure 4: Diagram of a minimal Data Exchange file for a single tomographic data set including raw projections, dark and white fields. In this case the attribute `axes` indicates the presence of theta vectors containing the positions of the rotation axis for each projection, dark and white images.

fig:MinimalTomo1

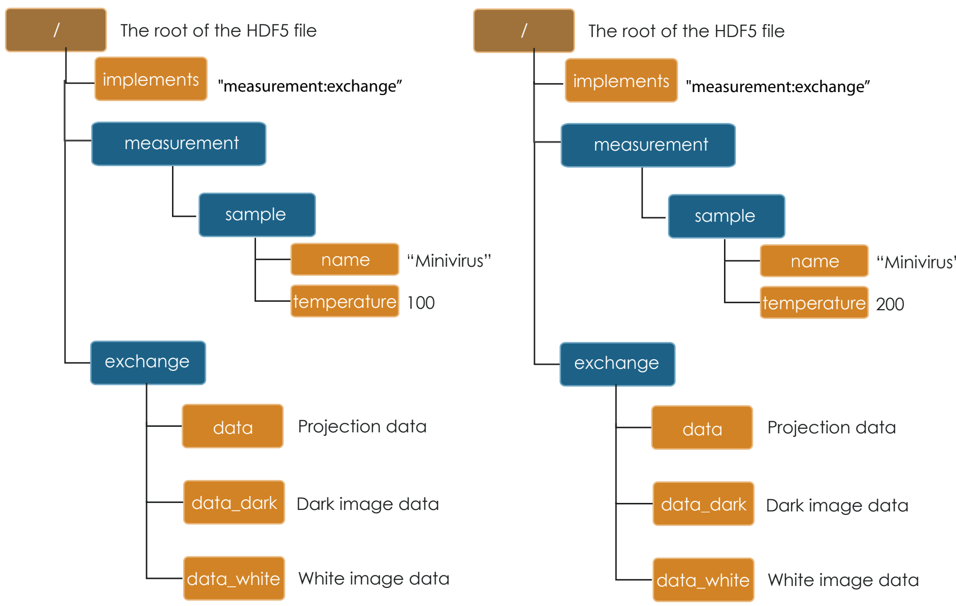


fig:MinimalTomo2

Figure 5: Diagram of two tomographic data sets taken at two different sample temperatures (100 and 200 K). To store the temperature in °C is necessary to add the attribute `units = "celsius"` to the `temperature` tag.

3.5.2 X-ray Energy Scan

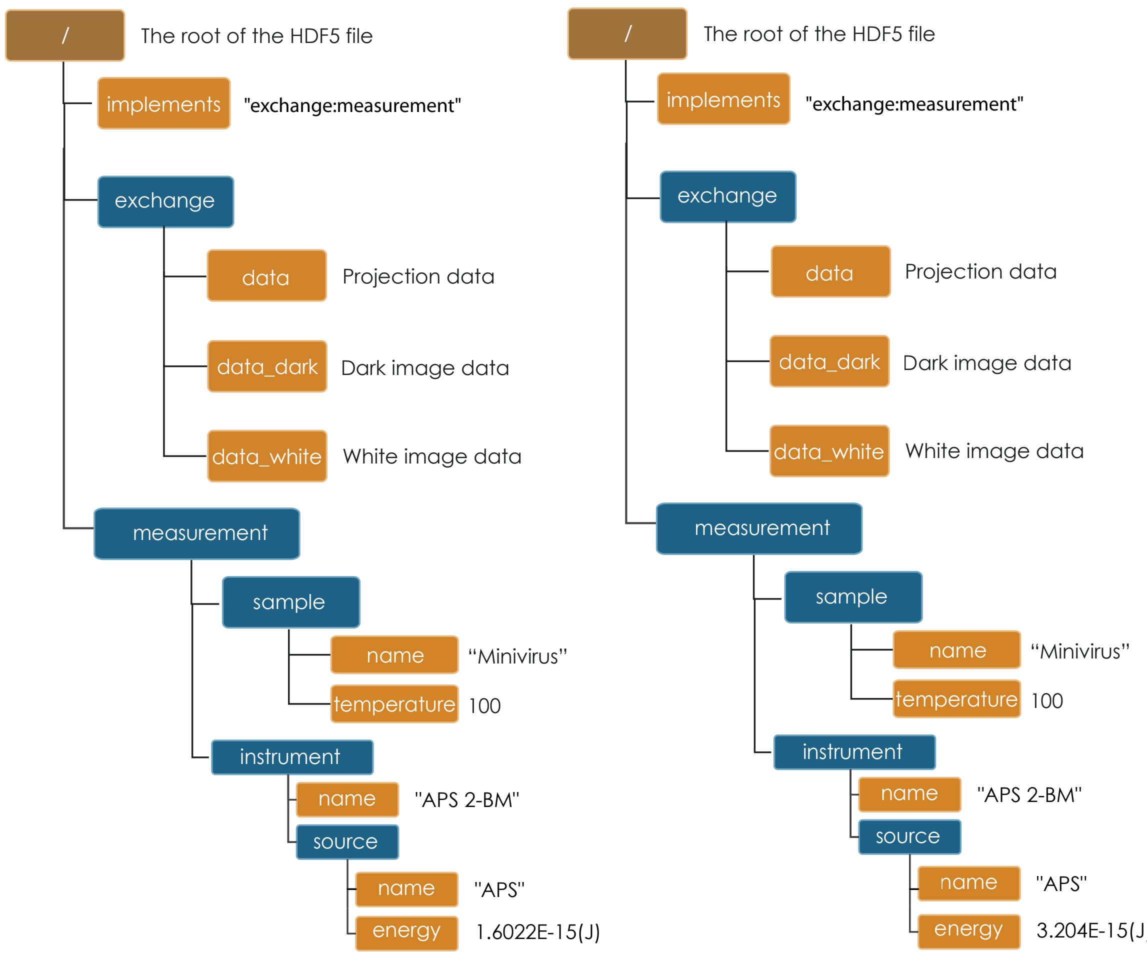


fig:MinimalTomo3

Figure 6: Diagram of two tomographic data sets taken at two different energy (10 and 20 keV). To store the temperature in keV is necessary to add the attribute `units = "keV"` to the `energy` tag.

3.5.3 Detector-sample Distance Scan

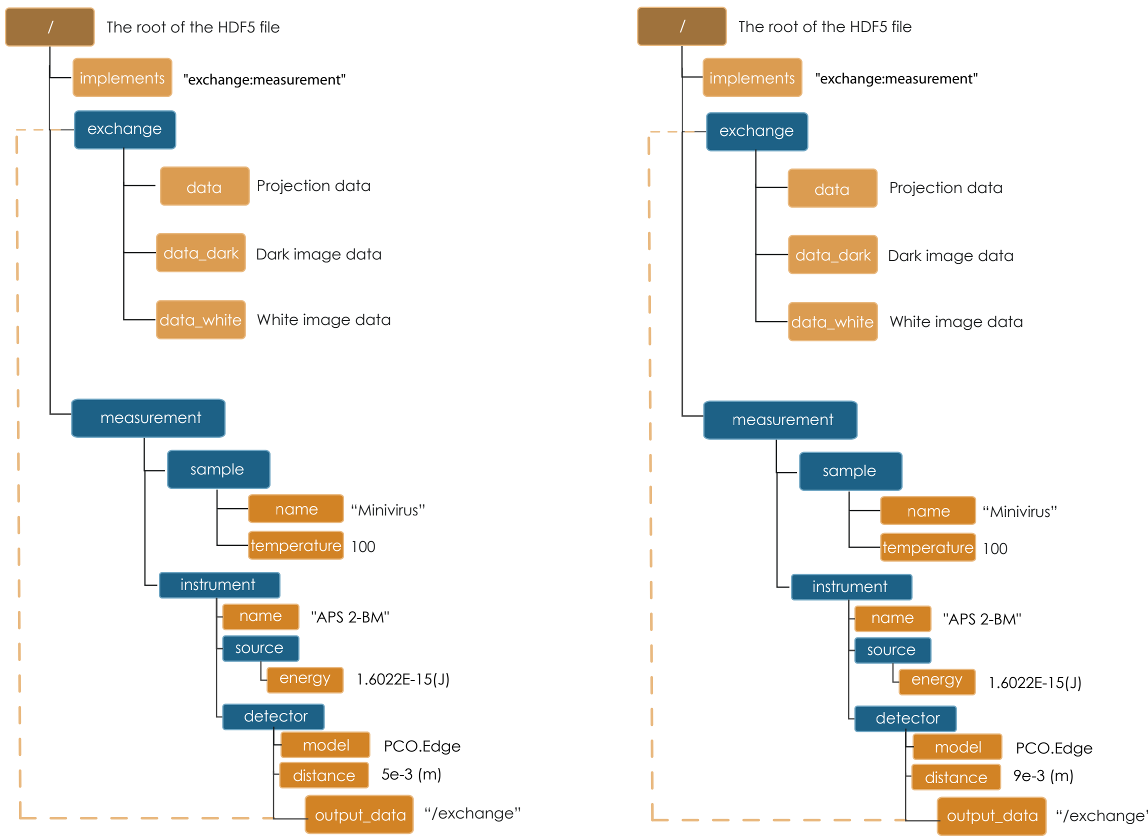


Figure 7: Diagram of two tomographic data sets collected with two different detector-sample distances (5 and 9 mm). Note the use of `output_data` dataset to associate the detector with the exchange group generated from the acquisition.

fig:MinimalTomo4

3.5.4 Series of Tomographic Measurements

A series of tomographic measurements, when relevant, can be stored in the same file appending $\_N$ to the measurement tag. 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 8 shows a file from a nano tomography experiment when the sample rasters through the field of view.

fig:NanoTomo1

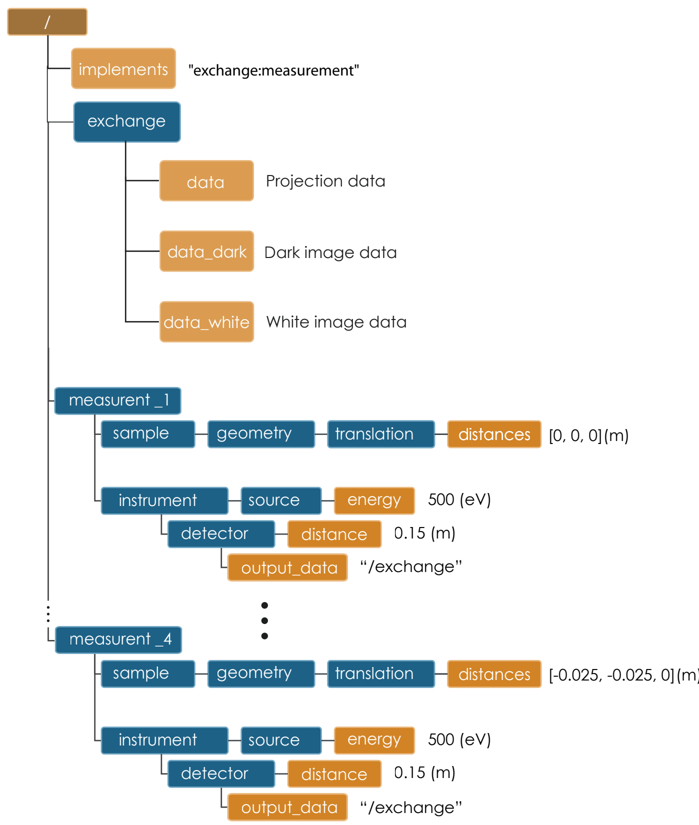


fig:NanoTomo1

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

sec:corereference

4 Data Exchange Core Reference

4.1 Top level (root)

table:top

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

Table 3: Top Level Members

Member	Type	Example
implements	string dataset	"exchange:measurement:provenance"
exchange_N	group	
`measurement_N`	group	
`provenance`	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 `provenance` information, if so these will be declared in implements as "exchange:measurement:provenance"

exchange_N

The data taken from measurements or processing.

`measurement_N`

Each measurement made on the sample.

`provenance`

The Provenance group describes all process steps that have been applied to the data.

4.2 Exchange

sec: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.

Table 4: Exchange Group Members

Member	Type	Example
`title`	string dataset	"tomography projections"
data	array dataset	n-dimensional dataset

`title`
: Descriptive title for data dataset.

data
: The primary scientific dataset. Additional related datasets may have any arbitrary name. Each dataset may have description, units, and axes attributes.

Table 5: data attributes

data:attr

Attribute	Type	Example
`description`	string attribute	"transmission"
`units`	string attribute	"counts"
`axes`	string attribute	"theta:y:x"

4.2.1 Storing and describing a multidimensional dataset

A multidimensional dataset may be stored with no units or axes attributes. In this minimal case, the units are assumed to be "count". Any other interpretation necessarily involves an implicit agreement between the file writer and reader. If there is no axis attribute, each axis is simply a linear integer dimension scale typically interpreted as pixels.

To describe the dataset further and provide for automatic interpretation, a units attribute should be added. Data Exchange recommends using SI units as described in Appendix A.1. The description attribute may be used to characterize the dataset further. An axes attribute may be added containing colon separated names of additional datasets called dimension scales. For example, an axes attribute of "theta:y:x" combined with an additional one-dimensional dataset called theta containing the rotation angles of each projection, provides a complete description for tomography. Note that it is not necessary to include x and y datasets since those are assumed to be a linear integer scale (pixels). For other cases, it may be necessary to include a one-dim dataset to describe each axis.

table:measurement

4.3 Measurement

This group holds sample and instrument information.

Table 6: Measurement Group Members

Member	Type	Example
sample	group	
instrument	group	

sample
: The sample measured.

instrument
: The instrument used to collect this data.

table:sample

4.3.1 Sample

This group holds basic information about the sample, its geometry, properties, the sample owner (user) and sample proposal information.

Table 7: Sample Group Members

Member	Type	Example
name	string dataset	"cells sample 1"
description	string dataset	"malaria cells"
preparation_date	string dataset (ISO 8601)	"2011-07-15T15:10Z"
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	

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.

Table 8: Geometry Group Members

Member	Type	Example
translation	group	
orientation	group	

table:geometry

4.3.1.1 Geometry

This class holds the general position and orientation of a component.

translation
: The position of the object with respect to the origin.

orientation
: The rotation of the object with respect to the coordinate system.

Only one orientation and one translation is permitted in each geometry class.

The position of the origin of the object should be explicitly defined for each object. If it is not defined it should be assumed to be the center of the object.

table:translation

4.3.1.1.1 Translation

This is the description for the general spatial location of a component - it is used by the Geometry class

Table 9: Translation Group Members

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.

table:orientation

4.3.1.1.2 Orientation

This is the description for a general orientation of a component - it is used by the Geometry class.

Table 10: Orientation Group Members

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 ”$\cdot$” 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.

4.3.1.2 Experiment

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

Table 11: Experiment Group Members

table:experiment

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

`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.

4.3.1.3 Experimenter

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

Table 12: Experimenter Group Members

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

`name` User name.

`role` User role.

`affiliation` User affiliation.

`address` User address.

`phone` User phone number.

`email` User e-mail address

`facility_user_id` User badge number

4.3.2 Instrument

table:instrument

The instrument group stores all relevant beamline components status at the beginning of a measurement.

`name`
Name of the instrument.

`source`
The source used by the instrument.

Table 13: Instrument Group Members

Member	Type	Example
name	string dataset	"XSD/2-BM"
source	group	
shutter_N	group	
attenuator_N	group	
monochromator	group	
interferometer	group	
detector_N	group	

shutter_N
: The shutter(s) used by the instrument.

attenuator_N
: The attenuators that are part of the instrument.

monochromator
: The monochromator used by the instrument.

detector_N
: The detectors that compose the instrument.

table:source

4.3.2.1 Source Class describing the light source being used.

Table 14: Source Group Members

Member	Type	Example
name	string dataset	"APS"
datetime	string dataset (ISO 8601)	"2011-07-15T15:10Z"
beamline	string dataset	"2-BM"
distance	float dataset	-48.5
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

name
: Name of the facility.

`datetime`
Date and time source was measured.

`beamline`
Name of the beamline.

`distance`
The source distance (m) from the sample.

`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).

`source`
Beam mode: TOPUP.

`beam_intensity_incident`
Incident beam intensity in (photons per s).

`beam_intensity_transmitted`
Transmitted beam intensity (photons per s).

table:shutter

4.3.2.2 Shutter

Class describing the shutter being used.

Table 15: Shutter Group Members

Member	Type	Example
name	string dataset	"Front End Shutter 1"
distance	float dataset	-48.5
status	string dataset	"OPEN"

`name`
Shutter name.

`distance`
: Shutter distance (m) from the sample.

`status`
: ”OPEN” or ”CLOSED” or ”NORMAL”

table:attenuator

4.3.2.3 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.

Table 16: Attenuator Group Members

Member	Type	Example
distance	float dataset	-35.7
thickness	float dataset	1e-3
attenuator_transmission	float dataset	unit-less
type	string dataset	”Al”

`distance`
: The Attenuator distance (m) from the sample. Negative distances represent beamline components that are before the sample while positive distances represent components that are after the sample. In this case the filter is located 35.7 m upstream of the sample.

`thickness`
: Thickness of attenuator along beam direction.

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

`type`
: Type or composition of attenuator.

table:monochromator

4.3.2.4 Monochromator Define the monochromator used in the instrument.

`type`
: Multilayer type.

Table 17: Monochromator Group Members

Member	Type	Example
type	string dataset	"Multilayer"
energy	float dataset	1.602e-15
energy_error	float dataset	1.602e-17
mono_stripe	string dataset	"Ru/C"

`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.

table:detector

4.3.2.5 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.

Table 18: Detector Group Members

Member	Type	Example
manufacturer	string dataset	"CooKe Corporation"
model	string dataset	"pco dimax"
serial_number	string dataset	"1234XW2"
distance	float dataset	5e-3
output_data	string dataset	"/exchange"

`manufacturer`
: The detector manufacturer.

`model`
: The detector model.

`serial_number`
: The detector serial number .

`distance`
: Distance of the detector from the sample.

`output_data`
: String HDF5 path to the exchange group where the detector output data is located.

section:provenance

4.4 Provenance

Data provenance is the documentation of all transformations, analyses and interpretations of data. Maintaining this history allows for reproducible data. The Data Exchange format permits tracking provenance using a combination of this Provenance group plus input/output datasets in the technique specific groups. 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.

The provenance group tracks the execution order of a series of processes using numbered process groups. Each process group uses references to other groups that describe the analysis in detail. Each process group represents the status of the process and some kind of informative summary message.

Table 19: Provenance Group Members

table:provenance

Member	Type	Example
process_N	group	

process_N

A process applied to the data.

section:process

4.4.1 Process

The process class holds basic information about a process. It is a generic container for recording the status of a process, and maintaining references to detailed, scientific domain-specific information.

Table 20: Process class members

table:process

Member	Type	Example
status	string dataset	"SUCCESS"
actor	string dataset	"reconstruction"
reference	string dataset	"/reconstruction"
message	string dataset	"Full reconstruction complete."

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

`actor`
Name of pipeline stage that executes this process group.

reference
Path to a process 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.

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.

Table 21: Provenance Group Example

process_1		
	status	"SUCCESS"
	reference	"/gridftp"
	message	"detector controller to cluster data transfer"
process_2		
	status	"SUCCESS"
	reference	"/sinogram"
	message	"modified axes from "theta:y:x" to "y:theta:x"
process_3		
	status	"SUCCESS"
	reference	"/ring_removal"
	message	"ring removal algorithm complete"
process_4		
	status	"SUCCESS"
	reference	"/reconstruction"
	message	"Full reconstruction complete."
process_5		
	status	"RUNNING"
	reference	"/export"
	message	"converting reconstructed data to tiff"

table:process

5 Data Exchange for X-ray Tomography

exchange: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 tomography process groups.

5.1 Exchange Group for X-ray Tomography

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.

Table 22: Exchange Group Members for Tomography

Member	Type	Example/Attributes
title	string dataset	"raw absorption tomo"
data	3D dataset	axes: "theta:y:x"
x	dimension scale 2	
y	dimension scale 1	
theta	dimension scale 0	units: "deg"
data_dark	3D dataset	axes: "theta_dark:y:x"
theta_dark	dimension scale 0	units: "deg"
data_white	3D dataset	axes: "theta_white:y:x"
theta_white	dimension scale 0	units: "deg"

title

This is the data title.

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 `axes` 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 (`axes` = "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.

`x, y`
: X and y are vectors storing the dimension scale for the second and third data array dimension. If x, y are not defined, the second and third dimensions of the data array are assumed to be in pixels.

`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.

5.2 Instrument Group for Tomography

`tomo:instrument`

The instrument group for X-ray tomography introduces an extended detector group definition, along with new interferometer, setup (for static setup), and acquisition (for scan setup) groups. The extended instrument group is as shown in Table 5.2.

`ble:tomo:instrument` `table:tomo:instrument`

Table 23: Instrument Group for Tomography

Member	Type	Example
name	string dataset	"XSD/2-BM"
source	group	same as core
shutter_N	group	same as core
attenuator_N	group	same as core
monochromator	group	same as core
interferometer	group	new
detector_N	group	extended from core
setup	group	new
acquisition	group	new

tomo:interferometer

5.2.1 Interferometer

This group stores the interferometer parameters.

Table 24: Interferometer Group Members

Member	Type	Example
start_angle	float dataset	0.000
grid_start	float dataset	0.000
grid_end	float dataset	2.4e-6
grid_position_for_scan	float dataset	1.3e-6
number_of_grid_steps	int dataset	8

`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.

table:tomo:detector

5.2.2 Detector Group for Tomography

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.

Table 25: Detector Group Members for Tomography

Member	Type	Example
manufacturer	string dataset	"CooKe Corporation"
model	string dataset	"pco dimax"
serial_number	string dataset	"1234XW2"
bit_depth	integer	12
x_pixel_size	float	6.7e-6
y_pixel_size	float	6.7e-6
x_dimension	integer	2048
y_dimension	integer	2048
x_binning	integer	1
y_binning	integer	1
operating_temperature	float	270
exposure_time	float	1.7e-3
frame_rate	integer	2
distance	float	5.7e-3
output_data	string dataset	"/exchange"
roi	group	
objective_N	group	
scintillator	group	
counts_per_joule	float	unitless
basis_vectors	float array	length
corner_position	3 floats	length
geometry	group	

manufacturer
: The detector manufacturer.

model
: The detector model.

serial_number
: The detector serial number .

`bit_depth`
: The detector bit depth.

`x_pixel_size, y_pixel_size`
: Physical detector pixel size (m).

`x_dimension, y_dimension`
: The detector horiz./vertical dimension.

`x_binning, y_binning`
: If the data are collected binning the detector x_binning and y_binning store the binning factor.

`operating_temperature`
: The detector operating temperature (K).

`exposure_time`
: The detector exposure time (s).

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

`distance`
: The detector distance from the sample.

`roi`
: The detector selected Region Of Interest (ROI).

`objective_`N
: List of the visible light objectives mounted between the detector and the scintillator screen.

`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:

$$\text{number of photons} = \frac{\text{source energy} \times \text{data counts}}{\text{counts per joule}}$$

`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.

table:roi

5.2.2.1 ROI

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

Table 26: ROI Group Members

Member	Type	Example
`name`	string dataset	"center third"
`x1`	integer	256
`y1`	integer	256
`x2`	integer	1792
`y2`	integer	1792

`x1`
: Left pixel position.

`y1`
: Top pixel position.

`x2`
: Right pixel position.

`y2`
: Bottom pixel position.

table:objective

5.2.2.2 Objective

Group describing the microscope objective lenses used.

Table 27: Objective Group Members

Member	Type	Example
`manufacturer`	string dataset	"Zeiss"
`model`	string dataset	"Axioplan"
`magnification`	float dataset	5
`na`	float dataset	0.8

manufacturer
Lens manufacturer.

model
Lens model.

magnification
Lens specified magnification.

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

table:scintillator

5.2.2.3 Scintillator

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

Table 28: Scintillator Group Members

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

manufacturer
Scintillator Manufacturer.

serial_number
Scintillator serial number.

name
Scintillator name.

scintillating_thickness
Scintillator thickness.

substrate_thickness
Scintillator substrate thickness.

table:tomo:setup

5.2.3 Setup

Group storing the positions of the stack of stages located under the sample before the tomographic data collection start. The stack defined in Table table:stack1 29 consists of an x-y-z stack and a pitch-roll (rotation_x and rotation_y) located under the rotary stage, plus and x-z (xx and zz) located above the rotary stage.

Table 29: Setup Group Members

Member	Type	Example
x_coordinate	float dataset	6.6E-3
y_coordinate	float dataset	4.3E-3
z_coordinate	float dataset	5.5E-3
xx_coordinate	float dataset	-8.1E-3
zz_coordinate	float dataset	1.6E-3
rotation_x	float dataset	0.00
rotation_z	float dataset	0.00

table:stack1

`x_coordinate, y_coordinate, z_coordinate`
: Position of the x-y-z stages located under the rotary stage at the beginning of the scan.

`xx_coordinate, zz_coordinate`
: Position of the x and z stages located above the rotary stage.

`rotation_x, rotation_z`
: Position of pitch and roll stages located under the rotary stage

le:tomo:acquisition

5.2.4 Acquisition

A tomographic data set consists of a series of projections, dark, and white field images. The dark field and white fields can be collected at any time before, during, or after the projection data collection. The acquisition class stores scan parameter values associated with the tomographic data collection.

`type`
: Tomographic data collection type: stop and go, fly scan etc.

`start_date`
: Tomographic data collection start.

Table 30: Acquisition Group Members

Member	Type	Example
type	string dataset	"stop and go"
start_date	string dataset - ISO 8601	"2011 07 15T15 10Z"
end_date	string dataset - ISO 8601	"2011 07 15T25 10Z"
number_of_projections	integer dataset	1441
dark_setup	group	
white_setup	group	
rotation_setup	group	

end_date
: Tomographic data collection end.

number_of_projections
: Number of tomographic projections.

dark_setup
: Dark field data collection setup.

white_setup
: White field data collection setup.

rotation_setup
: Rotation stage setup.

table:dark

5.2.4.1 Dark Setup

This group stores the parameters used to collect the dark field images

Table 31: Dark Setup Group Members

Member	Type	Example
frequency	int dataset	0
period	int dataset	0
number_pre	int dataset	1
number_post	int dataset	1

frequency
: The frequency of dark image collection during rotation. Specified as the number of regular projections to take prior to taking a number of dark images given by period. For example, a value of 10 means to take 10 projections, and then one or more dark images.

`period`
The number of dark images to collect during rotation at intervals specified by frequency.

`number_pre`
Number of dark images collected pre-rotation.

`number_post`
Number of dark images collected post-rotation.

table:white

5.2.4.2 White Setup

This group stores the parameters used to collect the white field images.

Table 32: White Setup Group Members

Member	Type	Example
frequency	int dataset	0
period	int dataset	0
number_pre	int dataset	1
number_post	int dataset	1
in_out_axis	string dataset	"X"
in	float dataset	0
out	float dataset	3e-3

`frequency`
The frequency of dark image collection during rotation. Specified as the number of regular projections to take prior to taking a number of dark images given by period. For example, a value of 10 means to take 10 projections, and then one or more dark images.

`period`
The number of dark images to collect during rotation at intervals specified by frequency.

`number_pre`
Number of dark images collected pre-rotation.

`number_post`
Number of dark images collected post-rotation.

`in_out_axis`
Indicates which axis is used to move the sample out of the field of view.

`in`
: Position of the in_out_axis when the sample is in the data collection position.

`out`
: Position of the in_out_axis when the sample is outside of the field of view.

table:rotation

5.2.4.3 Rotation Setup

This group stores the rotary stage parameters.

Table 33: Rotation Setup Group Members

Member	Type	Example
start_angle	float dataset	0.000
end_angle	float dataset	180.000
angular_step	float dataset	0.125
angular_speed	float dataset	0.2

`start_angle`
: Rotary stage position at the beginning of the scan.

`end_angle`
: Rotary stage position at the end of the scan.

`angular_step`
: Rotary stage step size (used if data are collected in a stop-go mode.

`angular_speed`
: Rotary stage speed.

5.3 Tomography Process Descriptions

This section documents a set of process descriptions for tomography data movement and processing. These process description groups are used in a data processing pipeline - each group provides the metadata for one stage in the pipeline.

table:sinogram

5.3.1 Sinogram Process Description

The sinogram process description group contains metadata required to generate sinograms from projection data. The input data is a projection ordered data cube, and the output is a sinogram ordered data cube. The output is stored in a new exchange group.

Table 34: Sinogram Group Members

Member	Type	Example
`name`	string dataset	
`version`	string dataset	1.0
`input_data`	string dataset	"/exchange_1"
`input_data_axes`	string dataset	"theta:y:x"
`output_data`	string dataset	"/exchange_2"
`output_data_axes`	string dataset	"y:theta:x"

`name`
: Algorithm name.

`version`
: Algorithm version.

`input_data`
: Reference to the exchange group containing the projection ordered input data.

`input_data_axes`
: Expected input 3D array axes order.

`output_data`
: Reference to the exchange group that will contain the sinogram ordered output data.

`output_data_axes`
: Output 3D array axes order.

5.3.2 Ring Removal Process Description

The ring removal process description group contains information required to run a ring_removal processing step.

`name`
: Algorithm name.

`version`
: Algorithm version.

Table 35: Ring Removal Group Members

table:ringremoval

Member	Type	Example
name	string dataset	
version	string dataset	1.0
input_data	string dataset	"/exchange_2"
output_data	string dataset	"/exchange_3"
coefficient	float dataset	1.0

input_data
: Reference to the exchange group containing input data.

output_data
: Path to the exchange group containing output data.

coefficient

able:reconstruction

5.3.3 Reconstruction Process Description

The Reconstruction process description group contains metadata required to run a tomography reconstruction. The specific algorithm is described in a separate group.

Table 36: Reconstruction Group Members.

Member	Type	Example
input_data	string dataset	"/exchange_3"
output_data	string dataset	"/exchange_4"
reconstruction_time	float dataset	37.5
reconstruction_slice_start	int dataset	1000
reconstruction_slice_end	int dataset	1030
rotation_center	float dataset	1048.50
algorithm	group	

input_data
: Reference to the exchange group containing input data.

output_data
: Reference to the exchange group containing output data.

reconstruction_time
: Total time (s) to reconstruct the full data set.

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.

section:algorithm

5.3.3.1 Algorithm

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

Table 37: Algorithm Group Members

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"
iterative_stop_condition	string dataset	"iteration_max"
iterative_iteration_max	int dataset	200
iterative_projection_threshold	float dataset	
iterative_difference_threshold_percent	float dataset	
iterative_difference_threshold_value	float dataset	
iterative_regularization_type	string dataset	"total_variation"
iterative_regularization_parameter	float dataset	
iterative_step_size	float dataset	0.3
iterative_sampling_step_size	float dataset	0.2
analytic_filter	string dataset	"Parzen"
analytic_padding	float dataset	0.50
analytic_processed_periods	float dataset	1
analytic_processed_number_of_steps	int dataset	7

table:algorithm

name
: Reconstruction method name: SART, EM, FBP, GridRec.

version
: Algorithm version.

implementation
: CPU or GPU.

`number_of_nodes`
: Number of nodes used. This parameter is set when the reconstruction is parallelized and run on a cluster.

`type`
: Tomography reconstruction method: analytic or iterative.

`iterative_stop_condition`
: iteration_max, projection_threshold, difference_threshold_percent, difference_threshold_value.

`iterative_iteration_max`
: Maximum number of iterations.

`iterative_projection_threshold`
: The threshold of projection difference to stop the iterations as p in $|y - Ax_n| < p$.

`iterative_difference_threshold_percent`
: The threshold of reconstruction difference to stop the iterations as p in $|x_{n+1}|/|x_n| < p$.

`iterative_difference_threshold_value`
: The threshold of reconstruction difference to stop the iterations as p in $|x_{n+1}| - |x_n| < p$.

`iterative_regularization_type`
: total_variation, none.

`iterative_regularization_parameter`
: lambda/alpha value in $(y - A_x)^2 + \alpha * L_1(x)$.

`iterative_step_size`
: Step size between iterations in iterative methods as δ_t in $x_{n+1} = x_n + \delta_t * f(x_n)$.

`iterative_sampling_step_size`
: Step size used for forward projection calculation in iterative methods.

`analytic_filter`
: Filter type.

`analytic_padding`

analytic_processed_periods
number of processed periods of the collected phase stepping curve (differential phase contrast - grating).

analytic_processed_number_of_steps
total number of processed phase steps (differential phase contrast - grating).

table:gridftp

5.3.4 Gridftp Process Description

The gridftp process description group contains metadata required to transfer data between two gridftp endpoints. This assumes a third party transfer.

Table 38: Gridftp Group Members

Member	Type	Example
source_URL	string dataset	"gsiftp://host1/path"
dest_URL	string dataset	"gsiftp://host2/path"

source_URL
A gsiftp URL for the source of the transfer.

dest_URL
A gsiftp URL for the destination of the transfer.

table:export

5.3.5 Export Process Description

The export process description group contains metadata required to extract and convert data from a Data Exchange (HDF5) file into another format.

name
Algorithm name.

version
Algorithm version.

input_data
Reference to the exchange group containing the input data.

Table 39: Export Group Members

Member	Type	Example
name	string dataset	
version	string dataset	"1.0"
input_data	string dataset	"/exchange_4"
output_URL	string dataset	"file://host/path"
output_data_format	string dataset	"TIFF"
output_data_scaling_max	float dataset	0.005
output_data_scaling_min	float dataset	-0.00088

output_URL

A file path and name, either plain or in URL format: file://host/path/file.tif

output_data_format

output_data_scaling_max

output_data_scaling_min

A Appendix

A.1 Default units for Data Exchange entries

appendix:units

The default units for Data Exchange entries follow the CXI entries definition, i.e. are SI based units (see table table:SI 40) unless the "units" attribute is specified. Data Exchange prefers to use the default SI based units whenever possible.

Table 40: SI (and common derived) base units for different quantities

table:SI

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	Ω
absorbed dose	gray	Gy
area	square meter	m^2
volume	cubic meter	m^3

A.1.1 Exceptions

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

A.1.2 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.

A.2 Geometry

A.2.1 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 9). The origin of the coordinate system is defined by the point where the X-ray beam meets the sample.

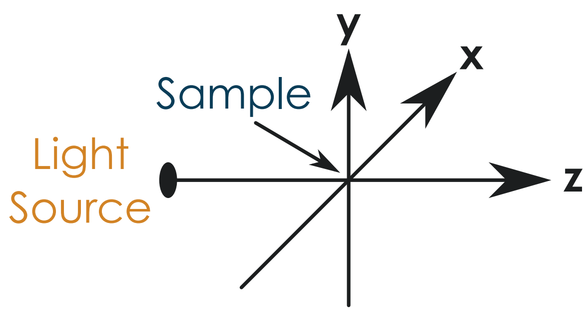


Figure 9: The coordinate system used by CXI. The intersection of the X-ray beam with the sample define the origin of the system. The z axis is parallel to the beam and points downstream.

fig:CoordSystem

OriginOfObjects

A.2.2 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.