Calibration Store

Content

Introduction

LCLS experiments use different type of detectors which produce raw data. Detectors are not perfect and in most cases these raw data can not directly be used in analysis. They need to be calibrated. Generic type of corrections for pixel geometry and intensity, such as dark rate (pedestals), common mode, mask, etc. need to be done in almost all experiments. These corrections use files with calibration parameters residing in calibration store.

Calibration store use regular file system to accumulate calibration files with a few simple rules about organization of the calibration store.

All calibration parameters are saved in text files. Full path to calibration file accounts for software version, detector location and name, calibration parameter type, parameters' validity run range.

Advantage or this approach: clean, simple structure of the calibration store structure and files' content with sufficient flexibility for analysis of data. Although, it is not recommended, regular OS tools can be used to browse, modify, re-locate files.

This note explains organization and content of the calibration store, tools for maintenance, application of corrections in psana, access in code to calibration parameters, and relevant references to other confluence documents.

Location

Directory containing calibration files is located at the same level as directories with data `xtc` or `hdf5` files:

```
/reg/d/psdm/<INSTRUMENT>/<experiment>/xtc
/reg/d/psdm/<INSTRUMENT>/<experiment>/hdf5
/reg/d/psdm/<INSTRUMENT>/<experiment>/calib
```

Full path to the calibration file consists of a few sub-directories under the `calib` directory, representing functional logical layers:

```
/reg/d/psdm/<INSTRUMENT>/<experiment>/calib/<calib-version>/<data-source>/<type>/<file>
```

where

- `<calib-version>` - is a directory reserved for different versions of calibration software. For each detector type it is defined through the dictionary.
- `<data-source>` - actual source of data. Each detector has an unique name in DAQ, which also includes its position. For example: `CxiDs1.0:Cspad.0`, `CxiDg1.0:Tm6740.0`, `CxiEndstation.0:Opal2000.1`, etc.
- `<type>`- type of calibration files:
    - `pedestals`,
    - `pixel_rms`,
    - `pixel_status`,
    - `pixel_mask`,
    - `pixel_gain`,
    - `pixel_bkgd`,
    - `common_mode`,
    - `filter`,
    - `geometry`, etc.
    - (`center`, `tilt`, `marg_gap_shift`, etc. are deprecated types).
- `<file>` - calibration file name consisting of the validity run range and extension .data. For example: `3-15.data`, `1-end.data`, `7-end.data`, etc.

Example:

```
/reg/d/psdm/CXI/cxi12315/calib/CsPad::CalibV1/CxiDs1.0:Cspad.0/pedestals/1-end.data
/reg/d/psdm/CXI/cxi12315/calib/CsPad2x2::CalibV1/CxiSc2.0:Cspad2x2.0/geometry/5-end.data
/reg/d/psdm/AMO/amo12313/calib/PNCCD::CalibV1/Camp.0:pnCCD.1/pixel_status/8-22.data
```

Tools

Calibration management tool

Calibration Management Tool - is a central tool for maintenance of calibration store. It allows to create,deploy, modify, and re-locate calibration files.

Apply calibration in psana

Intensity correction

Image intensity correction can be done with psana module ImgAlgos::NDArrCalib.

Geometry

Image reconstructed with geometry file can be obtained with module ImgAlgos::NDArrImageProducer.

Information about pixel coordinates, indexes in the image, areas, mask, etc. can be produced with ImgAlgos::PixCoordsProducer.

Direct access to calibration parameters in code

- class PSCalib::CalibParsStore - C++ interface to correction intensity parameters.
- class PSCalib::GeometryAccess - C++ interface to geometry parameters.
- class PSCalib.GeometryAccess - Python interface to geometry parameters.

References

- Detector Calibration Store
- DEPRECATED - IT WAS NEVER IMPLEMENTED: Calibration Store for LCLS2
- Calibration Management Tool
- Detector Geometry
- CSPAD Alignment
- CSPAD2x2 Alignment
- psana - Module Catalog