

Detector Calibration Constants Deployment

Content

- [Content](#)
- [Overview](#)
- [Code](#)
- [Command](#)
- [Repository](#)
- [What is available](#)
 - [epix100a default constants](#)
 - [methods to populate repository](#)
- [Functionality](#)
- [Extension](#)
- [References](#)

Overview

Population of the calibration store with constants is a significant problem for instrument scientists and users. Regular dark processing script [calibrun](#) produces significant portion of constants (pedestals, pixel_status, pixel_rms) for all regular detectors with single gain range.

Recently a few scripts were developed for dark processing and deployment constants for multi-gain detectors [Jungfrau and Epix10ka](#). Separate application was developed for [geometry constants deployment](#).

New script [deploy_constants](#) is intended for deployment of default-style calibration constants. These constants can be deployed depending on detector type, specific detector id/name, time stamp.

Code

Essential code is implemented under the Detector package in modules [app/deploy_constants](#) and [src/UtilsDeployConstants.py](#)

Command

In [psana environment](#) on any psana node (with access to data) use command

```
deploy_constants -h
```

deploy_constants -h output on 2022-06-09

```
(ana-4.0.42) [dubrovina@psanagpu105:~/LCLS/con-py2]$ deploy_constants -h
usage:
deploy_constants -e <experiment> -d <detector> -r <run-number> -C <calib-type> [-D] [-L <logging-mode>] [...]

REGULAR COMMAND EXAMPLE:
  deploy_constants -e xpptut15 -r 260 -d XcsEndstation.0:Epix100a.1 -C gain -D -c ./calib -D --runrange 10-20

Input parameters to select dataset:                                # DataSource argument:
  -e xpptut15 -r 260                                             # exp=xpptut15:run=260
  -e xpptut15 -r 260 -s :smd                                     # exp=xpptut15:run=260:smd
  -e xpptut15 -r 260 -s :dir=path/xtc::stream=0-79              # exp=xpptut15:run=260:dir=path/xtc::stream=0-79
  -x /cds/data/drpsrcf/mfx/mfxlv4920/xtc/<file-name.xtc>         # /cds/data/drpsrcf/mfx/mfxlv4920/xtc/<file-name.
xtc>

TEST COMMANDS:
Ex.1  for epix100a      : deploy_constants -e xpptut15 -r 260 -d XcsEndstation.0:Epix100a.1 -C gain -D -c ./calib
Ex.2  for Cspad CXI     : deploy_constants -e xpptut15 -r 380 -d CxiDs1.0:Cspad.0 -C gain -D -c ./calib
Ex.3  for Cspad CXI     : deploy_constants -e xpptut15 -r 380 -d CxiDs2.0:Cspad.0 -C gain -D -c ./calib
Ex.4  for Cspad XPP      : deploy_constants -e xpptut15 -r 320 -d XppGon.0:Cspad.0 -C gain -D -c ./calib
Ex.5  for cspad2x2       : deploy_constants -e xpptut15 -r 460 -d MecTargetChamber.0:Cspad2x2.3 -C gain -D -c .
/calib
Ex.6  for pnCCD          : deploy_constants -e xpptut15 -r 450 -d Camp.0:pnCCD.1 -C gain -D -c ./calib

Help:  deploy_constants -h

Deployment of calibration parameters

optional arguments:
-h, --help                show this help message and exit
-e EXP, --exp EXP          experiment name, default = None
-d DET, --det DET          detector name, default = None
-r RUN, --run RUN          run number of the existing dataset to get run time-
stamp, detector full name etc., default = None
-t TSTAMP, --tstamp TSTAMP
                           non-default time stamp (YYYYmmddHHMMSS) to search for
                           constants in repo. By default run time is used,
                           default = None
-s DSSUFFIX, --dssuffix DSSUFFIX
                           dataset name suffix beyond
                           exp=<expname>:run=<runnum><dssuffix>, for example
                           ":smd:dir=<path>/xtc/", default = None
-x DSNAME, --dsname DSNAME
                           overrides exp=<expname>:run=<runnum><dssuffix> if
                           specified, e.g. path to *.xtc file, default = None
-o DIRREPO, --dirrepo DIRREPO
                           non-default repository of calibration results, default
                           = /reg/g/psdm/detector/calib/constants/
-c DIRCALIB, --dircalib DIRCALIB
                           non-default deployment calib directory, different from
                           standard one, default = None
-C CTYPE, --ctype CTYPE   calibration type [gain, common_mode, etc.], default =
gain
-l LOGLEV, --loglev LOGLEV
                           logging mode, one of INFO, CRITICAL, WARN, WARNING,
                           ERROR, DEBUG, NOTSET, default = INFO
--dirmode DIRMODE          mode for all mkdir, default = 511
--filemode FILEMODE        mode for all saved files, default = 438
-D, --deploy               deploy constants to the calib dir, default = False
-R RUNRANGE, --runrange RUNRANGE
                           validity run range for output calibration file,
                           default = 0-end
```

Repository

Structure of repository is designed to account for detector type/id/name and time dependent constants:

```
Repository:
repodir = /reg/g/psdm/detector/calib/constants/

Log files:
<repodir>/logs/ - directory for log files
<repodir>/logs/<YEAR>/<time-stamp>_log_deploy_constants_<uid>.txt
ex: <repodir>/logs/2022/2022-06-09T121926_log_deploy_constants_dubrovin.txt

Detector type directory:
<repodir>/<dettype> - detector type directory for dettypes: cspad, cspad2x2, epix100a, pnccd
<repodir>/<dettype>/aliases.txt - dictionary between detector id/name and short alias

Directory for default files:
<repodir>/<dettype>/default/
                                <dettype>_default_<ctype>.txt
                                ex: epix100a_default_gain.txt

Specific detector directory:
<repodir>/<dettype>/<detector-id-or-name>/

Calibration type subdirectory:
<repodir>/<dettype>/<detector-id-or-name>/<calib-type>/
                                <dettype>_<alias>_<YYYYmmddHHMMSS>_<exp-name>_r<run-
number>_<ctype>.txt
                                ex: epix100a_0001_20100101000000_xpptut15_r0260_gain.txt
```

What is available

Currently a few files with constants are supplied for epix100a gain, common_mode and geometry

- /reg/g/psdm/detector/calib/constants/epix100a/default/epix100a_default_common_mode.txt, epix100a_default_gain.txt, epix100a_default_geometry.txt
- /reg/g/psdm/detector/calib/constants/epix100a/3925999616-0996579585-0553648138-1232098304-1221641739-2650251521-3976200215/gain/epix100a_0001_20100101000000_xpptut15_r0260_gain.txt reference to one of defaults.

If needed content of repository can be extended for other detectors and calibration types.

epix100a default constants

Default gains for epix100a are set equal to ones. Some of users requested constants as 0.060 keV/ADU. File epix100a_default_gain.txt contains gain factors 0.06 for all pixels. Command to install this constants is

```
deploy_constants -e <experiment> -d <detector> -r <run-number> -C gain -D
```

Test command below deploys constants under the local calib directory:

```
deploy_constants -e xpptut15 -r 260 -d XcsEndstation.0:Epix100a.1 -C gain -D -c calib
```

methods to populate repository

Since ana-4.0.45 a couple of methods are introduced to populate repository with constants, as shown in examples.

Example for method file_name_in_repo

```
from Detector.UtilsDeployConstants import file_name_in_repo
exp = 'xpptut15'
run = 260
det = 'XcsEndstation.0:Epix100a.1'
ctype = 'gain'
#tstamp = None
#rundepl = None
tstamp = '20220901120000' # or None
undepl = '123' # or None
print('File name in repository: %s' %\
      file_name_in_repo(exp, run, det, ctype, tstamp=tstamp, rundepl=rundepl)
)
```

Example for method save_epix100a_ctype_in_repo

```
import logging
logger = logging.getLogger(__name__)
logging.basicConfig(format='[%(levelname).1s]: %(message)s', level=logging.DEBUG) #INFO)

from Detector.UtilsDeployConstants import save_epix100a_ctype_in_repo
import numpy as np

arr2d = 0.061*np.ones((704,768))
exp = 'xpptut15'
run = 260
det = 'XcsEndstation.0:Epix100a.1'
ctype = 'gain'
tstamp = '20220901120000' # or None
rundepl = '123' # or None
save_epix100a_ctype_in_repo(arr2d, exp, run, det, ctype, tstamp=tstamp, rundepl=rundepl, fmt='%.4f')
```

Functionality

Command with correct list of parameters (-e xpptut15 -r 260 -d XcsEndstation.0:Epix100a.1 -C gain) tries to access if available or create the repository directory tree. If specific file for the run or specified (-t TSTAMP, --tstamp TSTAMP) time stamp is available it will be deployed under standard or specified (-c DIRCALIB, --dircalib DIRCALIB) calib directory. If calibration file is not found - nothing is deployed and appropriate warning is generated in the log file. Created directory tree for new detector can be used to add new default or time-stamp-dependent calibration files.

Extension

Currently repository is populated manually for a few default files.

Script [calibrun](#) can be updated to populate new repository for dark processing results (pedestals, pixel_status, pixel_rms, etc.)

References

- [calibrun](#) - CLI for dark calibration of regular detectors
- [Detector geometry constants deployment](#)
- [Jungfrau and Epix10ka Calibration](#)
 - [EPIX10KA2M and EPIX10KAQUAD](#), Production of calibration constants for multi-panel epix10ka detectors
 - [Jungfrau References](#), Jungfrau dark processing
- [Calibration Scripts Repository and Logging](#)