

psana

Introduction

Psana has a simple python-script interface, which can be run online (in realtime), offline, and parallelized over many cores/machines both online and offline. Once you have run the quick example below (we believe you should be able to run/understand it in <30 minutes) you can run a variety of "building block" examples [here](#).

There are also some higher-level hutch-specific analysis packages [here](#).

A paper discussing psana can be found here: <https://scripts.iucr.org/cgi-bin/paper?S1600576716004349>

Quick Example

To do analysis with a psana python script execute the following commands. This **requires the ability to ssh to an LCLS unix account and the ability to open up a graphics window on your computer ("X-windows")**. Two free options for graphics: we recommend the free [NX Technology](#) (for Mac or Windows) for improved performance, but [XQuartz](#) (Mac) or XWin32 (Windows) also works. Replace "YOURACCOUNTNAME" in the first line below with your unix account name.

Note that psana is a package in a larger python-ecosystem called [conda](#). Conda only supports the "bash" unix shell. Once you login you can see what shell you have by typing "echo \$SHELL". If it is not bash you can type "bash" to start the correct shell.

NOTE: we recommend to make an alias you can remember to set the analysis environment rather than always sourcing it in your .bashrc as we have seen issues e.g. with NoMachine when doing that.

```
ssh -X s3dfogin.slac.stanford.edu -l YOURACCOUNTNAME
ssh -X psana

source /sdf/group/lcls/ds/ana/sw/conda1/manage/bin/psconda.sh

cp /sdf/group/lcls/ds/ana/tutorials/psana1_examples/firstExample.py firstExample.py
python firstExample.py
```

The above 19-line ipsana.py script loops over 2 events and plots a CSPAD camera image with many applied calibration corrections, and geometry corrections for each of the detector panels. Close each image window to see the next one. You can examine the script with the command "cat firstExample.py" or edit it with unix editors like emacs, vim, or vi.

Features

Some of the features of psana-python:

- many of the "building block" tools can be used at FELs around the world: e.g. python, MPI and [scikit-beam](#) algorithms
- part of the world-standard [conda](#) python ecosystem
- same code works for online/offline analysis analysis (with real-time plot display), allowing users to do their entire analysis chain with one tool
- ability to use both simple and complex languages (python/C++)
- access to calibrated images both online/offline (pedestals, common-mode, geometry, bad-pixel mask) for complex detectors (CSPAD, pnCCD, EPIX)
- simple detector names using experiment-specific "aliases"
- ability to randomly access xtc data ("jump" to event in middle of file)
- support for fast parallel processing of xtc files with hundreds of cores using MPI
- open source