

DAQ User Guide (&troubleshooting)

- [Common features](#)
 - [DAQ epicsArch - Archived Data](#)
 - [Adding a single variable to the file](#)
 - [Add a whole file with a bunch of variables](#)
- [LCLS2 DAQ](#)
- [LCLS1 DAQ](#)
 - [DAQ Troubleshooting](#)
 - [Operating the DAQ](#)
 - [AMI Online Monitoring](#)
 - [Python Scripting](#)

Common features

DAQ epicsArch - Archived Data

You can add/remove any data you want in the epicsArchive data group readout.

The path is `/cvs/group/pcds/dist/pds/tmo/misc/`

The tmo file we use is `epicsArch_tmo.txt`. Let's look at two features of this

Adding a single variable to the file

Add a comment with a `#`

```
# Power meter at IM2K4
```

Add a pointer name of your choosing with `*`

```
*IM2K4_XrayPower
```

For the **LCLS2 DAQ**, you should also specify which protocol is used to access the PV:

Add the pv name specifying whether it is read using channel access (`ca`), or pv access (`pva`) at the end

```
IM2K4:PPM:SPM:VOLT_RBV ca
```

For the **LCLS1 DAQ**, you only list the PV

```
IM2K4:PPM:SPM:VOLT_RBV
```

Add a whole file with a bunch of variables

You can made file with a subset of PVs, give it a meaningful name like `epicsArch_vmi.txt` for all the vmi variables to record. You can then reference that file with `'<'` in the main file as below.

```
# MPOD/VMI information
```

```
< epicsArch_vmi.txt
```

LCLS2 DAQ

[LCLS-II Data Acquisition and Analysis](#)

LCLS1 DAQ

[Online Analysis Tutorial \(AMI\)](#) - presented at 2014 SSR/LCLS Users Meeting: LCLS Data Analysis Workshop

[DAQ Overview and Online Analysis](#) - presented at 2010 SSRL/LCLS Users Meeting : LCLS Data Analysis Workshop

[Online Analysis Design](#) - Matt's presentation for Jan 2012 DAQ Meeting

DAQ Troubleshooting

Trouble Shooting guide for scientists or first level responders.

Pedestals for detector correction

Operating the DAQ

- Editing a configuration
 - Example: Opal camera configuration
 - Changing a configuration while running
- Selecting detectors for readout
- Starting/stopping a run
- Watching progress of a run
- Running scans
 - Launching a Scan from DAQ Control GUI (deprecated)
 - Launching Scans Remotely (scripted)
- Running the sequencer
- Configuring the EVR
- Detector timing settings
- Recovering from errors: restarting the DAQ

AMI Online Monitoring

- Online Monitoring GUI
 - Using the Online Monitoring GUI
 - Writing a plug-in to the Online Monitoring GUI
 - Writing a user application (reads from shared memory) (deprecated old C++ psana)
 - Online Monitoring and Simulation Using Files (deprecated old C++ psana)
 - Writing a user application, offline analysis style (reads from a file) (deprecated old C++ psana)
- XTC playback (a.k.a Offline AMI)

Python Scripting

- Editing configurations - the 'pycdb' module
- Controlling the DAQ - the 'pydaq' module
- Monitoring the data - the 'pyami' module