

FAQ



Unknown macro: 'html'

- [Infrastructure](#)
 - [Problems accessing data, or data seems to have disappeared](#)
 - [How do I keep programs running if a ssh connection fails](#)
 - [Why did my batch job fail? I'm getting 'command not found'](#)
 - [sit_setup fails in script using ssh](#)
- [Psana](#)
 - [Where is my epics variable?](#)
 - [How do I access data inside a Psana class?](#)
 - [How do I find out the experiment number \(expNum\) or experiment?](#)
 - [Why isn't there anything in the Psana Event?](#)
 - [It seems that for as much as 5% of the time, CsPad DataV2 is not in the Event](#)
 - [How do I set psana verbosity from the config file?](#)
 - [Strange Results after doing Math with data in Python](#)
 - [Specialized Psana Options, liveTimeout, firstControlStream, etc](#)
- [Hdf5](#)
 - [Why is there both CsPad DataV2 and CsPad DataV1 in the translation?](#)
 - [How do I write hdf5 files from C++ or Python](#)
 - [Chunks, compression, and why is my random access algorithm so slow?](#)
- [TimeTool](#)
- [An error occurred during authentication](#)



Unknown macro: 'html'

Infrastructure

Problems accessing data, or data seems to have disappeared

Two things to check:

1. Have you been given access to view the data?
2. Has the data been removed due to the data retention policy?

For the first, new users to an experiment need to ask the experiment POC to add them to the experiment. After this is done, you must **log out and log back in** for the change to take affect.

For the second, when analysis code that used to work stops working, check to see if the xtc is visible. For example, if you are analyzing run 68 of xpptut13, take a look in the xtc directory for that experiment, i.e:

```
ls /reg/d/psdm/xpp/xpptut13/xtc
```

If the directory is visible but run 68 is not there, it maybe that the data was removed due to the [Data Retention Policy Version 2](#). The data is still available on disk and can be restored using the [Web Portal of Experiments](#).

If the xtc directory is not visible, make sure you are running on a node that can see the data (i.e, you are on a psana node, rather than a psdev or pslogin node). If it is still not visible, email pcds-help@slac.stanford.edu.

How do I keep programs running if a ssh connection fails

See if you can use the [LSF batch](#) nodes for your work. If not, see if nomachine technology will work. Otherwise, there are three unix programs to help with this are tmux, nohup and screen. None of these programs will preserve a graphical program, or a X11 connection, so run your programs in terminal mode.

tmux

For example, with tmux, if one does

```
ssh psexport
ssh psana
# suppose we have landed on psanacs040 and that there is a matlab license here
tmux
matlab --nosplash --nodesktop
```

If you lose the connection to psanacs040, you can go back to that node and reattach:

```
ssh psexport
ssh psanacs040
tmux attach
```

You need to remember the node you ran tmux on. If you are running matlab, you can run the [matlab license script](#) with the --show-users parameter to see where you are running it:

```
/reg/common/package/scripts/matlic --show-users
```

nohup

You could run a batch process with nohup (no hangup) as follows

```
nohup myprogram
```

For example, suppose we want to run a Python script that prints to the screen and save its output (the below syntax is for the bash shell):

```
nohup python myscript.py > myoutput 2>&1 &
```

Here we are capturing the output of the program in myoutput, along with anything it writes to stderr (the 2>&1), then putting it in the background. The job will persist after you logout. You can take a look at the output in the file myoutput the next day. As with tmux you will need to remember the node you launched nohup on.

Why did my batch job fail? I'm getting 'command not found'

Before running your script, make sure you can run something, for instance do

```
bsub -q psnehq pwd
```

(substitute the appropriate queue for psnehq). If you created a script and are running

```
bsub -q psnehq myscript
```

Then it maybe that the current directory is not in your path, run

```
bsub -q psnehq ./myscript
```

Check that myscript is executable by yourself, check that you have the correct #! line to start the script.

sit_setup fails in script using ssh

Users have run into issues in the following scenario

- writing a script that uses ssh to get to another node
- uses sit_setup to set up the environment for that node (in the script)

The issue is that sit_setup is an alias (defined by /reg/g/psdm/etc/ana_env.sh). In bash aliases are not expanded by default in non-interactive shells so you have two options:

1. './reg/g/psdm/bin/sit_setup.sh' instead of sit_setup
2. change the shell behavior by saying 'shopt expand_aliases'

Typically option 1) works best.

Psana

Topics specific to Psana

Where is my epics variable?

Make sure it is a epics variable - it may be a control monitor variable. An easy way to see what is in the file is to use psana modules that dump data. For instance:

```
psana -m psana_examples.DumpEpics exp=cxitut13:run=0022
```

will show what epics variables are defined. Likewise

```
psana -m psana_examples.DumpControl exp=xpptut13:run=0179
```

will *almost always* show what control variables are defined. It defaults to use the standard Source "ProcInfo()" for control data. It is possible (though very unlikely) for control data to come from a different source. One can use the [EventKeys](#) module to see all Source's present, and then specify the source for [DumpControl](#) through a config file.

How do I access data inside a Psana class?

Look for an example in the [psana_examples package](#) that dumps this class. There should be both a C++ and Python module that dumps data for the class.

How do I find out the experiment number (expNum) or experiment?

Psana stores both the experiment and expNum in the environment object - Env that Modules are passed, or that one obtains from the DataSource in interactive Psana. See [Interactive Analysis](#) document and the C++ reference for [Psana::Env](#)

Why isn't there anything in the Psana Event?

This may be due to a problem in the DAQ software that was used during the experiment. The DAQ software may have incorrectly been setting the L3 trim flag. This flag is supposed to be set for events that should not be processed (perhaps they did not meet a scientific criteria involving beam energy). When the flag is set, there should be very little in the xtc datagram - save perhaps epics updates. Psana (as of release ana-0.10.2 from October 2013) will by default not deliver these events to the user. The bug is that the flag was set when there was valid data. To force psana to look inside datagram's where L3T was set, use the option l3t-accept-only. To use this option from the command line do:

```
psana -o psana.l3t-accept-only=0 ...
```

Or you can add the option to your psana configuration file (if you are using one):

```
[psana]
l3t-accept-only=0
```

It seems that for as much as 5% of the time, CsPad DataV2 is not in the Event

The only distinction between CsPad DataV1 and DataV2 is the sparsification of particular sections as given in the configuration object. That is DataV2 may be sparser. The actual hardware is kicking out DataV1 but the DAQ event builder is making a DataV2 when it can. Sometimes the DAQ sends the original DataV1 instead of the DataV2. This can be due to limited resources, in particular competition with resources required for compressing cspad in the xtc files. If you do not find a DataV2 in the Event, look for a DataV1

How do I set psana verbosity from the config file?

Most all psana options can be set from both the config file as well as the command line. Unfortunately verbosity cannot be set from a config file. That is

```
psana -v -v ...
```

has no config file equivalent. This is because verbosity is a function of the Message service that psana is a client of, rather than server for. Unfortunately this makes it difficult to turn on debugging messages from within a python script that is run as

```
python myscript.py
```

However one can configure the message logging service through the MSGLOGCONFIG environment variable. In particular

```
MSGLOGCONFIG=trace python myscript.py
```

or

```
MSGLOGCONFIG=debug python myscript.py
```

turns on trace or debug level MsgLog messages. The above examples were tested with the bash shell. For full details on configuring the MsgLogger through the MSGLOGCONFIG environment variables, see https://pswww.slac.stanford.edu/swdoc/releases/ana-current/doxy-all/html/group__MsgLogger.html

Strange Results after doing Math with data in Python

One thing to bear in mind with the data in the Python interface, the detector data is almost always returns as a numpy array of an integral type. For example, after getting a waveform of Acqiris data, if you were to print it out during an interactive Python session, you might see

```
waveform
array([234, 53, 5, ..., 324], dtype=np.int16)
```

Note the dtype, this data is two byte signed integers. In particular if a value of the waveform is over 32000 and you add 10000 to it, those values will wrap around and become negative. It may be a good idea, before doing any math with the data, to convert it to floats:

```
waveform = waveform.astype(np.float)
```

Specialized Psana Options, liveTimeOut, firstControlStream, etc

There are a number of specialized options in psana that we do not generally document in the main user documentation. These are options that engineers, points of contact (POC's), or possibly users may need to deal with special circumstances. These are described in the table below:

package. module. option or psana. option	data type	option description
PSXtcInput. XtcInputMo dule. liveTimeout	integer	live time out in seconds run running in live mode (defaults to 120)
PSXtcInput. XtcInputMo dule. runLiveTim eout	integer	starting in ana-0.13.17, live time out when waiting for a new run (defaults to 0)
psana. first_control _stream	integer	which stream starts the control or IOC streams (xtcav, etc) defaults to 80
psana.l3t- accept-only	bool	when true, checks for trimmed datagrams. If the datagram is trimmed, it does not trigger an event for the user. Trimmed datagrams should be empty. There was a case where trimming was not set up properly, and we needed to turn this off to see the data. Note - this is different then the l3t pass/fail that POC's may setup - though one does not expect trimmed data unless the l3t is a fail.
psana. allow- corrupt- epics	bool	for data from before approximately July 2014 when the DAQ epics archiver recorded different variables at different rates, the index files do not contain enough information to properly get access the correct epics values. When this happens, and the user is running in index mode, psana will exit with a fatal error. Setting this flag to True will allow psana to analyze these runs (in index mode) but some of the epics data could be corrupted.

Hdf5

Topics specific to hdf5

Why is there both CsPad DataV2 and CsPad DataV1 in the translation?

The only distinction between CsPad DataV1 and DataV2 is the sparsification of particular sections as given in the configuration object. That is DataV2 may be sparser. The actual hardware is kicking out DataV1 but the DAQ event builder is making a DataV2 when it can. Sometimes the DAQ sends the original DataV1 instead of the DataV2. This can be due to limited resources, in particular competition with resources required for compressing cspad in the xtc files.

How do I write hdf5 files from C++ or Python

Python:

From Python we recommend h5py. For interactive Python, an example is found at [Using h5py to Save Data](#). For Python you can also use [pytables](#). This is installed in the analysis release. Do

```
import tables
```

In your Python code.

C++

If developing a Psana module to process xtc, consider splitting your module into a C++ module which puts ndarrays in the event store, and a Python module which retrieves them and writes the hdf5 file using h5py.

You can also work with the C interface to hdf5. hdf5 is installed as a package in the analysis release. From your C++ code, do

```
#include "hdf5/hdf5.h"
```

A tip for learning hdf5 is to run example programs from an 'app' subdirectory of your package. For example, if you create an analysis release and a package for yourself, create an app subdirectory to that package and put an example file there:

```
~/myrelease/mypackage/app/hdf5_example.c
```

Now run 'scons' from the ~/myrelease directory, and then run hdf5_example.

Psana Modules - Using the Translator:

The [Psana ddl based Translator](#) can be used to write ndarrays, strings and a few simple types that C++ modules register. These will be organized in the same groups that we use to translate xtc to hdf5. Datasets with event times will be written as well. To use this, create a psana config file that turns off the translation of all xtc types but allows translation of ndarrays and strings. An example cfg file is here: [psana_translate_noxtc.cfg](#) You would just change the modules and files parameters for psana and the output_file parameter to Translator.H5Output. Load modules before the translator that put ndarrays into the event store. The Translator will pick them up and write them to the hdf5 file

Chunks, compression, and why is my random access algorithm so slow?

By default, Hdf5 files are translated in compressed chunks. The compression (standard gzip, with deflate=1 (the range in [0,9])) reduces file size by about 50%. The chunk size varies with the data type. The chunk policy focuses on not having too many chunks, as we believe this degrades performance. Typically we shoot for chunks of 16MB, however for full cspad, this is only 4 objects per chunk - so we default to using 22 objects per chunk - or 100MB chunks. This is fine for a program that linearly reads through a hdf5 file, or a parallel program that divides the file into sections - i.e, start, middle, end, but it is not optimal for a random access analysis. If you read one cspad, you read the other 21 in its chunk, and decompressed the whole 100MB chunk.

In terms of how this plays with the filesystem cache, and hdf5 cache, the filesystem will cache the original data in the hdf5 - so if you revisit cspad from the same chunk, the compressed data may be cached by the filesystem, but hdf5 will have to decompress the whole chunk again. Now hdf5 maintains its own chunk cache which stores recently used chunks (uncompressed). There is one cache global cache, but one can also create caches on a per dataset basis. The Translator creates datasets with a cache to hold 3 chunks - so a 300MB cache for a cspad dataset. This information is available to any program that reads the hdf5 file, but whether or not it is used is unclear. If you are writing your own program to read an hdf5 file, you can set the overall chunk cache, as well as per dataset cache's explicitly.

There are options with translation to control the objects per chunk. One can also turn off compression when translating.

TimeTool

The TimeTool results can be obtained in one of two ways depending on the experimental setup. The first is directly during data acquisition, the second is during offline analysis using the psana module TimeTool.Analyze. Regarding data acquisition, for data recorded prior to Oct 13, 2014, the timetool results were always recorded as EPICS PV's. After Oct 13, 2014, they are recorded in their own data type: TimeTool::DataV*. The version increments as the time tool develops. For data recorded up to around November 2014, this was DataV1. Around December of 2014 it changed to DataV2. Regarding offline analysis with the TimeTool.Analyze psana module, similarly to the experiment data files, this module puts a TimeTool::DataV* object in the event store depending on the version of the software. Please see the documentation on the TimeTool in the [Psana Module Catalog - TimeTool Package](#)

An error occurred during authentication

This is a common error when a user has not obtained an AFS token. You can obtain a token with:

```
kinit
```