

2021 DAQ Instructions

Outline

- **Start SVT DAQ VNC**
- **Software installation locations**
- **Start rogue servers**
- **Taking a CODA run**
- **Online Monitoring**
- **DAQ known issues**
 - CODA stuck in Download phase
- **Hall B run Database**
- **Coda / Rogue Errors**
- **SVT Start-up Procedure**
- **SVT Baseline/Threshold Procedure**
- **SVT Chiller Recovery**
- **Interlocks status**
- **Motor recovery procedure**

Start SVT DAQ VNC

From counting house as hpsrun (should already be done):

In terminal: `daqvnc.py connect config hps-svt-2`

In another terminal: `daqvnc.py connect config hps-svt-3`

Remotely (in Hall B gateway):

```
ssh -Y clonfarm2
vncviewer :2
```

```
ssh -Y clonfarm3
vncviewer :2
```

Remotely (outside jLab network):

```
#tunnel to login.jlab.org. This opens a new xterm with an open tunnel to login
xterm -e "ssh -N -L <port>:hallgw4.jlab.org:22 <username>@login.jlab.org" &

#Tunnel to a machine behind the firewall spawning a top process to keep the tunnel open. First put
gatewayPin+OTP then pwd to the machine behind the firewall

xterm -e "ssh -t -p <port> -L 5902:localhost:5902 <username>@localhost \ \" ssh -t -L 5902:localhost:5902
clonfarm2.jlab.org \ \"top\" \" \" &
```

Then:

- On Linux: `vncviewer localhost:2`
- On Mac: open screen sharing and connect to localhost:5902

Software installation locations (11 Ago 2021)

⚠ Any crate related command should be issued by SVT experts only

SDK software installation to talk to the atca crate

```
source /usr/clas12/release/1.4.0/slac_svt_new/V3.4.0/i86-linux-64/tools/envs-sdk.sh
```

examples:

```
cob_dump --all atcal => dumps the RCE status (booted is 0xea)
cob_rce_reset atcal ==> resets all the RCEs
cob_rce_reset atcal/1/0/2 ==> resets a particular dpm (in this case dpm1)
cob_cold_data_reset
```

et atcal ==> "power cycles" the RCEs (sometimes they do not come back up nicely so rce_reset might be needed after)

....

INFO : clonfarm2 go.....

SVT software is installed in

```
/data/hps/slac_svt/
=> diskless is exported to the cobs via nfs (server hosted on clonfarm1, NFS has to be v2 !! important )
=> daq is exported to the cobs via nfs. Has been compiled on DTm1
=> server contains the current software installation
```

Start rogue servers

On clonfarm2 start the rogue server

```
source /usr/clas12/release/1.4.0/slac_svt_new/anaconda3/etc/profile.d/conda.sh
conda activate rogue_5.9.3
cd /data/hps/slac_svt/server/heavy-photon-daq/software/scripts/
python SvtCodaRun.py --local --env JLAB --epicsEn
```

The "epicsEn" flag is necessary to enable controls via Epics.

On clonfarm3 start the dummy server

```
python SvtCodaDummy.py --local --env JLAB
```

Taking a CODA run



If opening the Rogue GUIs for the first time, make sure all of the FEBs are turned off.

To take a CODA run, both the rogue server and a dummy server need to be started. To start the rogue server, first ssh into clonfarm2 and issue the following commands

```
source /usr/clas12/release/1.4.0/slac_svt_new/anaconda3/etc/profile.d/conda.sh
conda activate rogue_5.9.3
cd /data/hps/slac_svt/server/heavy-photon-daq/software/scripts/
python SvtCodaRun.py --local --env JLAB --epicsEn
```

The dummy server runs on clonfarm3 and can be brought up as follows after ssh'ing into that machine

```
source /usr/clas12/release/1.4.0/slac_svt_new/anaconda3/etc/profile.d/conda.sh
conda activate rogue_5.9.3
cd /data/hps/slac_svt/server/heavy-photon-daq/software/scripts/
python SvtCodaDummy.py --local --env JLAB --epicsEn
```

At this point, the FEBs and hybrids can be brought up via the medm GUIs.

Once the hardware has been powered up, you can initialize a run in CODA using the configuration **PROD77_SVT** and the config file **hps_v2_svtOnly_noThresh**.

❌ CODA run control should be running in a VNC on clondaq7. If not, contact the DAQ expert.

Online Monitoring

The online monitoring can be started by

```
startSVTCheckout
```

Which is in the hpsrun user path.

DAQ known issues

We experience few issues with the DAQ infrastructure during run cycles. In particular, it's advised to reset the DPMs in these cases:

- After a baseline run
- At every Coda cycle (when go in configure or Download)

Usually it's done before anything else.

```
/data/hps/slac_svt/server/heavy-photon-daq/software/scripts/resetDataDpms.sh
```

If the DPMs do not come up after several resets, one might try a cold reset (but might create issues in the recovery)

```
heavy-photon-daq/software/scripts/coldResetDataDpm.sh
```

At this point, the FEBs and hybrids can be brought up via the medm GUIs.

Coda stuck in Download phase

Coda might get stuck in "Download" phase in the case the Rogue Server or the Dummy Rogue servers are not running.

=> If clonfarm2 is stuck in "waiting for Download transition" means that SvtCodaRun.py (main Rogue server) is not running. Start Rogue GUI on clonfarm2

=> If clonfarm3 is stuck in "waiting for Download transition" means that SvtCodaDummy.py (dummy Rogue server) is not running or lost communication with CODA.

- Check if SvtCodaDummy is running on clonfarm3

- If SvtCodaDummy is running, kill the process and restart it. If CODA doesn't progress to Prestart
- If SvtCodaDummy is not running, start it. If CODA doesn't progress to Prestart
- **If Coda doesn't Progress to Prestart => Cancel, Reset and retry from Configure**

Trigger diagnostics

```
tcpClient hps11 tdGStatus  
tcpClient hps11 tsStatus
```

Stop Triggers

```
tcpClient hps11 'tsDisableTriggerSource(0)'  
tcpClient hps11 'tsEnableTriggerSource()'
```

Notes:

- the "tcpClient" commands above will also **pause** and **restart** data taking! This is relevant for the Stepan's instructions on the run TWiki regarding the beam pause for under 40 min.
- According to Cameron: the commands can be run on any **clon** machine when logged in as hpsrun.

Hall B run Database

This is the link to the Hall B run Database to find all the necessary run specifications

<https://clasweb.jlab.org/rcdb>

Coda / Rogue Errors

1) RssiContributor::acceptFrame[X]: error mismatch in batcher sequence 000000dc : 00000000

This is in the event builder and indicates that we are dropping packets.

This is raised in [RssiContributor.cc](#) in the event_builder by acceptFrame thread

Effect:

rol BUSY

Causes: Dropped frames.

Fix: Reset COBs

2) ROL crashes during downloadprestart phase showing readback errors on the FEBs

This usually indicates that the FEBs lost clock

Effect: not possible to run

Causes: FEBs lost clock

Fix: Recycle FEBs and run control

SVT Start-up Procedure : FROM SVT OFF

1) If it is running, kill Rogue by deleting the GUI SvtCodaRun.py that is running in hpsrun-clonfarm2 (TigerVNC window). If doesn't come down, ctrl+c on the terminal where rogue is running

2) In a terminal on hpsrun-clonfarm2: source resetDataDpms.sh

This is in [clonfarm2:/data/hps/slac_svt/erver/heavy-photon-daq/software/scripts](#)
if you want to issue this in a new terminal do

Usually it's done before anything else.

```
bash
sconda
crogue
sdk
cd $SSVT
cd heavy-photon-daq/software/scripts/
source resetDataDpms.sh
```

3) In the terminal in which Rogue had been running, execute "cob_dump --all atca1". In the resulting print-out, look for all codes to be 0xea. If they do not all go to 0xea:

- re-doing the dump command may help, since the status can change in a few seconds (VF)
- otherwise: execute cob_cold_data_reset atca1 and then cob_rce_reset atca1.

4) Check if Flange boards are ON or OFF. If FEBs are powered on, power them off in the GUI window svtFebMain.adl as follows:
At the top of the window next to "ALL FEB" do in this order

- a) Turn ANAN off
- b) Turn ANAP off
- c) Turn DIGI off

5) Start Rogue (SvtCodaRun.py)

6) In the TigerVNC window where RunControl is (CODA), execute the Configure and Download. Doing this early, at this state, can prevent problems that sometimes occur if this is done after the SVT startup.

7) In svtFebMain.adl, at the top, next to ALL FEB, turn DIGI on. Then wait until all the digi currents exceed 1.0 amp (about 10 seconds). Then in quick succession turn on in this order: ANAP followed by ANAN.

8) In the Rogue GUI SvtCodaRun.py Variables tab, set "Poll Enable" to True.
Watch below there for all of the links to turn "True".
To be sure, click "FebArray->AXIVersion->Uptime" and watch to see that the Uptime is incrementing every few seconds

9) Go to the HpsSVTDAQRoot tab in Rogue and click "Load Settings". Select the file rce-test-eel.yml. Wait for that to complete (takes a few seconds).

10) Go back to the variables tab in Rogue and set "GlobalHybridPowerSwitch" to "On".
Wait for all the ANAP currents to settle.
If a FEB is not responding at this point, in Rogue set Poll Enable to "False" and then power cycle just the bad FEB in the same sequence as described above, again waiting for DIGI to go above 1 amp before turning on ANAP and ANAN.

11) Go again to the HpsSVTDAQRoot tab in Rogue and repeat the "Load Settings".

11a) If HV Bias is OFF, turn it ON before data taking.

12) If all the FEBs look good (green) and SVT bias is on, then CODA should be ready for "Prestart" followed by "Go"

Restart a run from CODA Configure : with SVT ON

Very often the "End Run" procedure doesn't end cleanly and one needs to restart the run from CODA Configure. The worst step of the procedure is the Download stage as the clock sent to the FEBs get reset and the FEBs might lose clock and communication to the Rogue server.

This is the smoother procedure I have found.

- 1) Close Rogue
- 2) Reset the DPMS (see paragraph above)
- 3) Start Rogue **after** the DPMS are up
- 4) Keep **PollEn** to **False**
- 5) CODA Configure, followed by CODA Download
- 6) **Check the Flange Epics adl screen (I think this meant to be SVT -> "FEB Main" GUI instead. --VF)**

- If some FEBs show red values of voltages / currents it's likely they lost clock. Power-cycle those FEBs only.
 - Turn off the Febs DIGI, ANAP and ANAN
 - When OFF, turn up DIGI and wait until it comes to ~1A (Additionally some output will be shown in the terminal where Rogue process is running, confirming that communication is established)
 - When DIGI is UP and connection to Rogue is established, turn on ANAP and ANAN
- If all FEBs are green proceed with (7)

7) Set **PollEn** to **True** and wait for all FEB links to become True

8) If all True and no errors in Rogue, LoadSettings (see previous section)

9) Turn on the Hybrids with the global button (1st tab, GlobalHybridPwrSwich)

10) LoadSettings (see previous section)

11) If all went well ==> Prestart and GO.

SVT Baseline/Threshold Procedure

There are two procedures to compute the baselines and thresholds for the HPS.

- Online: take a dedicated baseline run with no beam and analyse the noise using hpstr
- Offline: compute the baselines from a production run via offline analysis based on hps_java and hpstr

Compute Online Thresholds

1) If open, close Rogue Gui

2) Reset the data DPMs before doing a baseline run, and then restart the Rogue GUI

```
sdk
/data/hps/slac_svt/server/heavy-photon-daq/software/scripts/resetDataDpms.sh
```

3) CODA Configure: **PROD77_SVT**

4) CODA Download: **trigger/HPS/Run2021/Before_Sep16/hps_v2_svtOnly_noThr.trg**

5) CODA Prestart

6) CODA Go and take ~2000 events

At the start of the run clonfarm2 and clonfarm3 terminals on CODA runControl (ROCs) might output a large amount of messages which differ with respect to the normal run. This is expected and won't affect the run.

6a) Reminder: need to restart the DPMs after taking the baseline run!

7) The data will end up on clondaq7, so move the data (via scp) from clondaq7 to clonfarm1

```
scp clondaq7:/data/stage_in/hpssvt_<run_number>/hpssvt_<run_number>.evio.00000 clonfarm1:/data/hps/slac_svt
/server/thresholds/run/
```

8) Next process this data to produce a threshold file. Thresholds require the fw channel mapping so use (as hpsrun):

```
bash
sconda
crogue
source /data/hps/src/setupHpstrEnv.sh
cd /data/hps/slac_svt/server/thresholds/run
hpstr /data/hps/src/hpstr/processors/config/evioSvtBl2D_cfg.py -i hpssvt_<run_number>.evio.00000 -o
hpssvt_<run_number>_bl2d_fw.root -c fw
python makeSvtThresholds.py -i hpssvt_<run_number>_bl2d_fw.root -o
svt_<run_number>_thresholds2pt5sig_1pt5sigF5H1
cp svt_<run_number>_thresholds2pt5sig_1pt5sigF5H1.dat ../
```

9) The Thresholds are loaded in CODA by the following configuration (accessible from any clonfarmX machine)

```
/usr/clas12/release/1.4.0/parms/trigger/HPS/Run2021/svt/svt_config.cnf
```

Change the line to point to the new threshold file corresponding to the line starting with **RCE_THR_CONFIG_FILE**, and make a new log entry



After a baseline run it's **necessary** to reset the data dpms in order to properly read data back from the SVT

Compute Online Baselines

The baselines are produced by

```
bash
sconda
croque
source /data/hps/src/setupHpstrEnv.sh
cd /data/hps/slac_svt/server/thresholds/run
hpstr /data/hps/src/hpstr/processors/config/evioSvtBl2D_cfg.py -i hpssvt_<run_number>.evio.00000 -o
hpssvt_<run_number>_bl2d_sw.root -c sw
python makeSvtCond.py -i hpssvt_<run_number>_bl2d_sw.root -o svt_<run_number>
```

The output svt_<run_number>_cond.dat is in the format needed to upload to the database.

This file should be moved to **ifarm:/work/hallb/hps/phys2021_svt_calibrations**

They can be loaded into the db by logging into ifarm and doing:

```
cd /work/hallb/hps/phys2021_svt_calibrations
./load_calibrations.py -f svt_<run_number>_cond.dat -r <run_number>
```

Change the Online monitoring baselines

After changing the baselines in the online reconstruction and database, change the baselines in the online monitoring application

```
/home/hpsrun/hps_software/reconMonitoringSettings/SvtCheckout2021.settings
/home/hpsrun/hps_software/reconMonitoringSettings/KFTrkAndReconOnMon2021.settings

#the line is
UserRunNumber=14335

==>And change that to the appropriate run
```

Finally restart the Online monitoring applications for the svt expert and remote shifter vncs

====> REMOTE SHIFTER: vnc running on **clonsl1**.

```
#This will kill both svt and ecal monitoring
killall -9 java

#In one terminal:
startSVTCheckout

#In another terminal (check actual script name):
startEcalMonitoring
```

Same operation for the EXPERT VNC on clonfarm2

Interlocks



Please be very careful when touching the interlocks, both hardware and software. A wrong setting of the interlocks might cause the hardware to go in ERROR state and require a Power Cycle causing bigger problems. Be very attentive about the limits and the conditions before activating the interlocks.

AS OF October 2021 ALL SOFTWARE INTERLOCK ARE DISABLED. DO NOT ACTIVATE THEM

IN PARTICULAR VACUUM SOFTWARE INTERLOCK *MUST*** BE KEPT IN BYPASS BECAUSE THE GAUGE IS NOT PROVIDING RELIABLE READINGS**

In the case of a failure of the Cooling system (or vacuum or other hardware failures) interlocks might need to be re-enabled. On the night of 2nd October 2021, we experienced a failure of the SVT Chiller and we had to reset the interlocks. Below there is a screenshot of the Hardware (PLC) and Software interlocks for HPS.



- The PLC Interlocks screen can be opened by the hps_epics main adl, then devices, then SVT PLC. **The PLC interlocks are on the LEFT side of the picture**
- The Software interlocks can be opened by the hps_epics main adl, then devices, then SVT Soft Interlocks. **The Soft Interlocks are on the RIGHT side of the image**



AS OF October 2021 ALL SOFTWARE INTERLOCK ARE DISABLED. DO NOT ACTIVATE THEM

IN PARTICULAR VACUUM SOFTWARE INTERLOCK *MUST*** BE KEPT IN BYPASS BECAUSE THE GAUGE IS NOT PROVIDING RELIABLE READINGS**

Resetting PLC Interlocks after SVT Chiller failure

After an SVT Chiller failure the following **hardware (PLC)** interlocks will probably trip. In order to restart the system the interlocks that tripped or an in a **fault** state. Above each interlock State (marked with the flag Disabled/Enabled), there is the current reading, the good value (which is checked against the current reading to kick the interlock) and the interlock state.

- EXAMPLE ABOVE: Flow in **green** 1 is checked against Flow Good Value (1). If the Chiller stops for some reason and the flow goes to **red 0**, then the Interlock goes in **fault** state.

In order to restart the system, the interlock need to be disabled if the current measurement do not match the Good Value

In the case of a chiller failure we saw the RTDs interlocks in Disabled state. They were re-enabled when the RTD readings were back in the interlock safe range.

Restoring the SVT temperature after a SVT Chiller failure

An example of the temperature ramp-down procedure can be found on :

<https://logbooks.jlab.org/entry/3916838>



If > 20' were lost before restarting the chiller, do not go directly to -18C. It is recommended to bring the detector to the setpoint temperature using the step by step procedure.

Remember to put a - (minus) sign at the chiller setpoint temperature and press enter when the set point temperature is changed in EPICS.

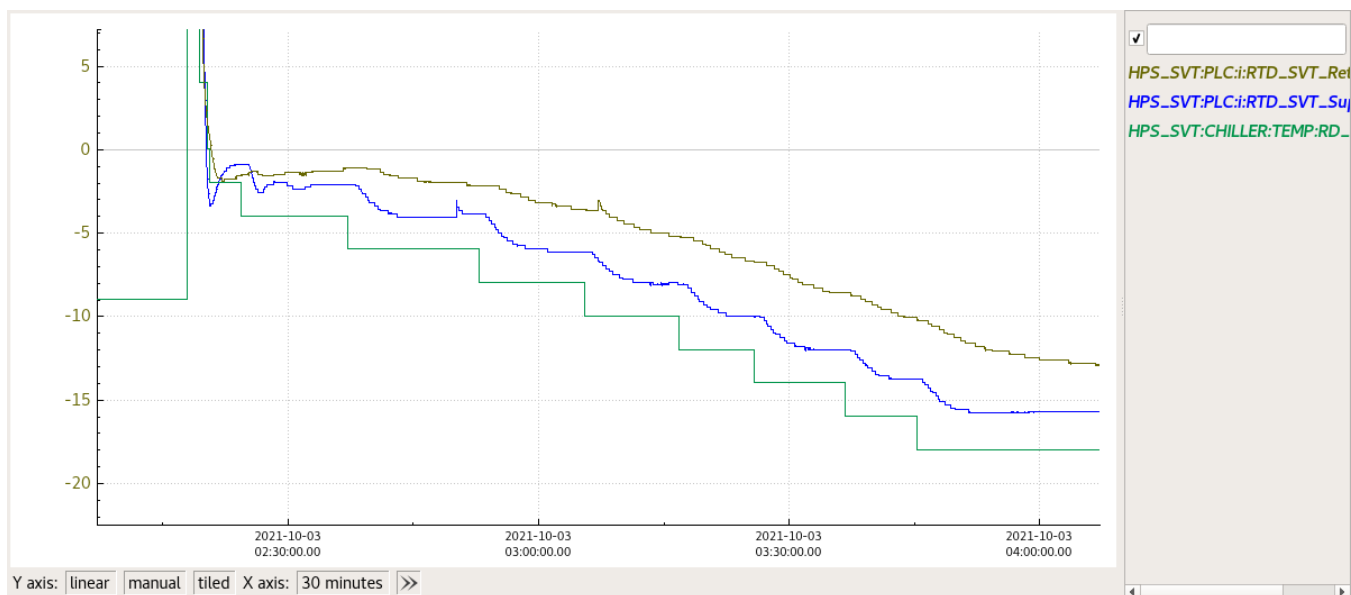
Fast recovery of the chiller

- 1) Open HPS_EPICS => Devices => SVT PLC
- 2) Ac power DIS (turns off the AC box)
- 3) Ac power ENA (turns on the AC box)
- 4) Disable the SVT Chiller interlocks (for example if flow = 0 and flow interlock is enable you won't be able to start the chiller. Hence disable the interlock)
- 5) Chiller Ctrl Stop
- 6) Chiller Ctrl Start
- 7) Set the setpoint temperature (if needed. See Step-by-step if the chiller was off from bit of time) and press ENTER
- 8) Re-enable the interlocks (when all is Green) in the SVT-PLC

Step-by-step procedure

- Prepare a myaPlot of the following quantities (from hps_epics click on the ! next to StripCharts and select myaPlot)
 - HPS_SVT:PLC::RTD_SVT_Return-Value ==> This is the return temperature from the SVT measured by an RTD (During a run is usually 4C higher than the Supply. 2C higher if not running)
 - HPS_SVT:PLC::RTD_SVT_Supply-Value ==> This is the supply temperature read from the RTD. Usually 2C higher than the Chiller temperature
 - HPS_SVT:CHILLER:TEMP:RD_ ==> The chiller temperature setpoint
- The SVT temperature at the restart of the chilling procedure will be unknown. One can try to find a good starting point by setting a temperature of the Chiller such that the Supply-Value is about 1 - 2 C below the Return Value and one can see the Return-Value decreasing. If the Return-Value RTD is increasing, it means that the Chiller setpoint temperature needs to be lowered. (see figure below at around 2:10 AM we were trying to find the proper setpoint temperature (green) and trying to put the Supply (blue) below the Return (gold))
- Gradually bring the setpoint temperature down trying to maintain about ~2C spread (ideally) between the Return and the Supply. In the figure below a spread of about 4C was used. One can notice that the supply temperature flattens faster at fixed setpoint but keeps bringing the return down. Try to go down in temperature to more or less maintain the temperature gradient more or less constant to optimise time
- At the end of the procedure, wait a bit to have the SVT at around -13.8C – -14C.
- **At that point the Various interlocks can be restored**

In this figure is shown the ramp down of the SVT Temperature as described in the procedure above.



Resetting MPOD Interlocks after SVT Chiller failure

After an SVT Chiller failure the Power Supplies for HV will interlock. To check the status of the SVT LV/HV Power supplies go to

<http://hpsmpod/> (Accessible behind the hall-b firewall)

If you see "Interlock" in the last column, means that the Power Supplies are interlocked and need to be reset. It can be done on the expert hps_epics adl of SVT Bias.

On top of the page there is "Reset MPOD interlocks" in red. Click on it and then check if the interlocks are cleared on the the hpsmod webpage

Motor recovery procedure

On Oct 07 2021 we experienced an issue with moving the target from 20um to 8um.

The issue was that the target would not move to 8um when selected from the Epics gui. From the epics gui we saw no indication of problem

To check the status of the motors (SVT TOP, SVT BOT and TARGET) on the motor controller

- 1) Connect to hpsxps.jlab.org from browser.
- 2) Login with username and password.
- 3) Front Panel Move
- 4) Check the screen for errors.
- 5) If one motor says "Uninitialized" click on Initialize and then Home. The motor will be sent home and re-calibrated
- 6) If was just a transient error, this should clear up and recover the motor functionality.

Offline Baseline+Thresholds Procedure

APV25 channel pedestals shift with occupancy, and can significantly change after beam is introduced, in comparison to an online baseline run (no beam) in channels close to the beam, especially in the first 2 layers. Additionally, the pedestals can change over time with radiation exposure, so the more time that elapses between online baseline runs, the more likely the high occupancy channel pedestals may no longer be consistent. For these reasons, we have an offline baseline fitting tool to extract the pedestals from production run data.

TO FIT OFFLINE BASELINES AND GENERATE A BASELINE AND THRESHOLD DATABASE FILE FOR A RUN:

1) SETUP OFFLINE FIT RUN DIRECTORY ON CLONFARM1

1. From behind the hall gateway `'ssh -XY -C clonfarm1'`
2. Access bash tools required for running environment scripts `'scl enable devtoolset-8 bash'`
3. Setup environment to run fit jobs `'source /data/hps/src/setupHpstrEnv.sh'`
4. Check setup Enter `'which hpstr'` into terminal. Should return `"/data/hps/src/hpstr/install/bin/hpstr"` if successfully setup.

5. Navigate to offline fit job run directory '**cd /data/hps/slac_svt/server/offlinePedestals**'
6. Inside this, make evio storage directory to hold data from clondaq7 '**mkdir hps_0<run_number>**'

2) COPY 10 EVIO FILES FROM IFARM CACHE TO CLONFARM1

1. In the clonfarm1 terminal
2. Copy 10 cache files to Clonfarm1 '**scp <user>@ftp.jlab.org:/cache/hallb/hps/physrun2021/data/hps_0<run_number>/hps_0<run_number>.evio.000{40..49} /data/hps/slac_svt/server/offlinePedestals/hps_0<run_number>/'**

3) COPY ONLINE BASELINE FROM IFARM TO CLONFARM1

1. Open new terminal and **ssh** into ifarm
2. Navigate to online baselines '**cd /work/hallb/hps/phys2021_svt_calibrations**'
3. Find the "svt_<online_run_number>_cond.dat" file with <online_run_number> closest to, but less than, the current run being fit.
 - a. For example, if you're trying to offline fit "Run 14596", locate online baseline file "svt_014581_cond.dat"
4. From Clonfarm1 terminal, copy online baseline to clonfarm1 '**scp <user>@ftp.jlab.org:/work/hallb/hps/phys2021_svt_calibrations/svt_0<number>_cond.dat /data/hps/slac_svt/server/offlinePedestals/online_baselines/**'
 - a. File may already be available in online_baselines, so check first.

4) SETUP AND RUN JOBS ON CLONFARM1

1. In Clonfarm1 terminal
2. Run evio rawsvthits_hh jobs to create sample 0 histograms to fit baselines
 - a. Locate run thresholds file that was used for the current run
 - i. Find file with <number> closest to but less than <run_number> being fit '**/data/hps/slac_svt/server/thresholds/svt_0<number>_thresholds<settings>.dat**'
 - ii. Copy this file path to be used in next step.
 - b. Modify 'vars.json' '**vim /data/hps/slac_svt/server/offlinePedestals/vars.json**'
 - i. **update "run_number"** to match the current run
 - ii. **update "thresh"** to match file from step 2.a.i.
 - iii. save changes and close file
 - c. Create jobs '**source /data/hps/slac_svt/server/offlinePedestals/mkjobs.sh**'
 - d. Run jobs '**source run_pool.sh**'
 - e. When jobs finish, combine histograms into one file '**hadd output/rawsvthits_<run_number>_hh.root output/rawsvthits_0<run_number>*.root**'
 - i. NOTE the '0' before <run_number> has been removed from the hadded file name. You must do this!
 - f. Wait for jobs to finish before proceeding to step 3.
3. Run **offline_baseline_fit** jobs
 - a. Create jobs '**source /data/hps/slac_svt/server/offlinePedestals/mkjobs_blfits.sh**'
 - b. Run jobs '**source run_pool_blfits.sh**'
 - c. When jobs finish, combine layer fits into one file '**hadd output/hps_<run_number>_offline_baseline_fits.root output/hps_<run_number>_offline_baseline_layer*.root**'

5) GENERATE OFFLINE BASELINE AND THRESHOLD DATABASE FILES

1. Run **python analysis** script
 - a. From clonfarm1 terminal directory /data/hps/slac_svt/server/offlinePedestals/
 - b. Run python analysis '**python3 offlineBaselineFitAnalysis.py -i output/hps_<run_number>_offline_baseline_fits.root -o output/hps_<run_number>_offline_baseline_fits_analysis.root -b online_baselines/svt_<online_run_number>_cond.dat -dbo output/hps_<run_number>_offline_baselines.dat -thresh output/hps_<run_number>_offline_thresholds.dat**'
 - i. new offline baselines are located output/hps_<run_number>_offline_baselines.dat
 - ii. new offline thresholds are located output/hps_<run_number>_offline_thresholds.dat
2. Cleanup the output directory
 - a. '**rm -rf ./scratch/***' to clear job scratch dir
 - b. '**mkdir output/<run_number>**' and move "output/rawsvthits_0<run_number>_hh.root", "output/hps_<run_number>_offline_baseline_fits_analysis.root" and ALL of the "<file>.dat" files into that output/<run_number> directory for safe-keeping.
 - c. Remove all of the loose files associated with this run inside of output/