

# Instructions for running DAQ system on VTP

## Warning

You'll need experience on how to run an experiment with the CODA DAQ system. For help in this area, David Abbott, Vardan Gurjyan, or Carl Timmer among others can give some assistance.

## Getting Ready

You'll need an account on the DAQ server dubhe mounted under /daqfs. Login into indra-s1 on the DAQ subnet (29) with multiple terminals.

You can use the coda distribution or you can copy it for a little protection. The correct coda is contained in:

```
/daqfs /coda/3.10_arm
```

Do:

```
setenv CODA /daqfs/coda/3.10_arm
```

Be sure you remove any existing \$CODA/common/jar/jfxrt.jar file! Use David Abbott's COOL database, EXPID=experiment\_0, config = test\_VTP\_stream or test\_VTP\_ts\_stream

Since Dave might be running a DAQ with that EXPID at the same time, a better idea is to copy his COOL database by doing:

```
cp -r ~abbottd/coda/cool_v2 myCool  
setenv COOL_HOME /daqfs/home/me/coda/myCool
```

Then set your EXPID to something else:

```
setenv EXPID myExp
```

Now run jcdit on Dave's database and jcdit on the copied database. Load test\_VTP\_stream config from Dave's and create a config on yours which copies Dave's. This is done by hand.

Now you can run a DAQ system with EXPID = myExp and not collide with Dave's platform.

In my case, I copied both configs mentioned above into my database.

Although I didn't do this, it's easiest to create a script to set your environment. Each host may be a little different.

## Running a DAQ

You should already be logged into indra-s1. The platform and rcgui are both **Java** processes.

In addition to setting CODA, EXPID and COOL\_HOME, setup the environment by doing the following to run the platform:

```
setenv CLASS_PATH '/daqfs/coda/3.10_arm/common/jar/*'  
setenv PATH /daqfs/coda/3.10_arm/common/bin:$PATH
```

Then run the CODA run control platform:

```
platform
```

To run the rcgui I had more trouble so I had to set the PATH and CLASS\_PATH to run it out of my own distribution:

```
rcgui
```

These need to be up and running before the ROCs, TS, and PEB are started. The aggregator needs to be running before them as well.

## Setting up VME ROC

Do:

```
ssh davme1
```

In addition to setting CODA, EXPID and COOL\_HOME, setup the environment by doing:

```
setenv LD_LIBRARY_PATH /daqfs/coda/3.10_arm/Linux-x86_64/lib:  
/daqfs/coda/3.10_arm/linuxvme/Linux-x86_64/lib
```

Once the CODA platform is up, run the ROC by doing:

```
cd /daqfs/coda/3.10_arm/Linux-x86_64/bin/coda_roc -i -n ROC1
```

Be sure to use the name of the ROC you used in the config. You should see it connect by looking at the platform's output.

## Setting up VTP ROC

Do:

```
ssh davtp1
```

In addition to setting CODA, EXPID and COOL\_HOME, setup the environment by doing:

```
setenv LD_LIBRARY_PATH /daqfs/coda/3.10_arm/Linux-armv7l/lib:  
/daqfs/coda/3.10_arm/linuxvme/Linux-armv7l/lib
```

Once the CODA platform is up, run the ROC by doing:

```
cd /daqfs/coda/3.10_arm/Linux-armv7l/bin/coda_roc -i -n VTP1
```

Be sure to use the name of the ROC you used in the config. You should see it connect by looking at the platform's output.

## Setting up TS

If your config calls for a hardware trigger (i.e. config = test\_VTP\_ts\_stream), do:

```
ssh dafarm44
```

In addition to setting CODA, EXPID and COOL\_HOME, setup the environment by doing:

```
setenv LD_LIBRARY_PATH /daqfs/coda/3.10_arm/Linux-i686/lib:  
/daqfs/coda/3.10_arm/linuxvme/Linux-i686/lib
```

Once the CODA platform is up, run the TS by doing:

```
cd /daqfs/coda/3.10_arm/Linux-armv7l/bin/coda_ts -i -n TS1
```

Be sure to use the name of the TS you used in the config. You should see it connect by looking at the platform's output.

## Setting up EB

You should already be logged into indra-s1. The EB is a **Java** process. In addition to setting CODA, EXPID and COOL\_HOME, setup the environment by doing:

```
Setenv CLASS_PATH '/daqfs/coda/3.10_arm/common/jar/*'
```

Once the CODA platform is up, run the ROC by doing:

```
cd /daqfs/coda/3.10_arm/Linux-x86_64/bin/coda_emu_peb -i -n PEB1
```

Be sure to use the name of the PEB you used in the config. You should see it connect by looking at the platform's output.

## Setting up Aggregator

You should already be logged into indra-s1. This is a **Java** process. In addition to setting CODA, EXPID and COOL\_HOME, setup the environment by doing:

```
cd /daqfs/home/me/coda  
git clone https://github.com/JeffersonLab/ersap-vtp.git
```

```
cd ersap-vtp  
./gradlew clean  
./gradlew fatJar  
cd bin
```

```
setenv CLASSPATH '/daqfs/me/coda/ersap_vtp/build/libs/*:$CODA/common/jar/*'
```

```
./ersap_vtp 6000 6001
```

Parameters to the executable are communication ports for two streams. This needs to be running before you start up the ROCs so they can connect to it. Note that this writes local files, so do **NOT** start it up in a directory with no write permission.