

# Online Reconstruction Tools

- [Introduction](#)
- [Installation](#)
- [Starting the Server](#)
- [Using the Client](#)
  - [Client Console](#)
  - [Client Command Line](#)
  - [Using a Command File](#)
- [Viewing Plots](#)
  - [Showing Plots in a Java Web Application](#)
    - [Setup the Web Application](#)
    - [View the Plots in the Browser](#)
  - [Viewing plots in JAS3](#)
- [ET Software](#)
  - [Installing the ET Software](#)
  - [Running the ET Ring](#)
  - [Running an Event Producer](#)
- [Creating Plots](#)
  - [Plotting Drivers and Steering Files](#)
- [Starting a New Run](#)

## Introduction

A new set of components has been developed to replace the now deprecated [Monitoring Application](#). Instead of having all functionality implemented within a single application, the online reconstruction and monitoring has been split up into several different programs in order to provide a scalable and flexible system, which requires much less configuration for shifters who just need to look at the plots. A server creates and manages stations that run the online reconstruction in headless mode. A command line client and interactive console allows the user to send commands to the server and receive responses. Finally, a number of GUI clients such as JAS3 and a Java web application can connect to a remote AIDA tree which shows an aggregated set of plots from all of the stations.

## Installation

The online reconstruction server and client are part of the [HPS Java](#) project.

Start by creating a scratch area on your machine where you can work with the tools.

```
mkdir /scratch && cd /scratch
```

You do not have to use this exact directory, but it is recommended to have a dedicated directory for online reconstruction configuration and data files. The instructions in this documentation will generally assume that you are working within this directory.

The updated online reconstruction tools are currently on a branch of [HPS Java](#), which you can checkout and build using this command:

```
git clone https://github.com/JeffersonLab/hps-java
cd hps-java
git checkout online-recon-dev
mvn clean install -DskipTests
```

Now, you can install the online reconstruction scripts into your work area:

```
# Go into online recon module within the hps-java project
cd online-recon


# Install the online recon run scripts to your work dir
mvn install -DskipTests -DskipCheckstyle -DassembleDirectory=/scratch

# Optionally add the script directory to the path.
export PATH=/scratch/bin:$PATH
```

The scripts should now be installed to the `/scratch/bin` directory, and you should be ready to test the client and server.

## Starting the Server

## ET Software

 The server cannot startup without connecting to an ET ring. If you are testing the tools outside of the counting house where this is not provided, then follow the instructions under "ET Software" for installing and running an ET ring and data producer.

The connection parameters for your local ET ring need to be specified in an initial server config file like this:

### my.cfg

```
et.buffer=/tmp/ETBuffer
et.host=localhost
et.port=11111
```

These settings provide the hostname where the ET ring is running, the port used for accepting connections, and the name of the buffer file. These need to exactly match the settings on your actual ET ring or the server will fail to startup. Currently, these cannot be changed once the server is started.

Once you have confirmed that the ET ring is running and your configuration file is valid, you may launch the server as a background process:

```
hps-recon-server --host localhost --port 22222 -w $PWD/stations -c my.cfg &
```

The `--host` switch specifies a valid network hostname or IP address for the server. For testing and development, it is preferable to use the generic `localhost` but this is not provided as a default. Otherwise, this should be a valid system (DNS) name, or leave it blank to use the actual system name (usually equivalent to the output from the `uname` command on Linux).

The `--port` switch can be used to designate a specific port for the TCP/IP socket to listen on (by default it is actually 22222 already).

The `-w` switch gives the directory where station data such as log files will be written out. By default, this is the directory where the server was started.

Finally, the `-c` option gives the location of the server's initial configuration file, which most importantly contains the parameters for connecting to the ET system (see above).

The server will write its log output to `logs/server.log` instead of the terminal.

## Using the Client


### Client Console

Now you may open the interactive console for sending command requests to the server:

```
hps-recon-client
```

This will open a prompt that allows you to send commands to the server and see the results in your terminal.

#### Console

 Whenever you see `online>` in a code block, you should run the command in the online reconstruction console rather than your system shell.

Type `help` into the console to show the documentation for the client command line interface:

```
online> help
```

This is the output of the above command:

## GENERAL

```
help - print general information
help [cmd] - print information for specific command
exit - quit the console
```

## SETTINGS

```
port [port] - set the server port
host [host] - set the server hostname
file [filename] - write server output to a file
append [true|false] - true to append to output file or false to overwrite
terminal - redirect server output back to the terminal
```

## COMMANDS

```
config - Set new server configuration properties
create - Create a new station
list - List station information in JSON format
log - Tail log file of station (hit any key to stop tailing)
remove - Remove a station that is inactive
save - Save the current set of plots to a ROOT or AIDA file
set - Set a configuration property
shutdown - Shutdown the server
start - Start a station that is inactive
status - Show server and station status
stop - Stop a station
```

You can also get help for a specific command using:

```
online> help [command]
```

For instance:

```
online> help create
```

This will show you the syntax for using that specific command.

If the server was started with a custom port, or it is running on a particular remote host, these can be specified like:

```
online> port 22222
online> host localhost
```

By default, output from the server is printed to the console, but it can instead be saved to a log file:

```
online> file client.log
online> append true
```

Configuration of the online reconstruction stations can be performed using the *set* command:

```
# lcsim detector name for conditions
online>set lcsim.detector HPS-PhysicsRun2016-Pass2

# run number for conditions
online>set lcsim.run 7798

# lcsim steering resource
online>set lcsim.steering /org/hps/steering/recon/PhysicsRun2016OnlineRecon.lcsim
```

The above will configure the stations to run on 2016 data. The detector, run, and steering parameters are all required for running an online reconstruction station.

Once the configuration is set properly, a single station can be created using a command like:

```
online>create 1
```

The argument is the number of stations you want the server to create.

```
online>create 8
```

You will need to test how many stations you can run simultaneously, given the specific configuration being used and the number of plots being created /updated at once. The system should roughly scale to the number of cores on your machine, depending on its overall load.

Stations that have been created but not started are in the *inactive* state. Stations which are inactive have no running system process and no entries in the combined plot tree.

Inactive stations can be started using a command such as:

```
online>start
```

The start command can also take a list of station IDs to start:

```
online>start 1 2 3
```

Many of the online recon commands are similar in that they take no arguments (usually meaning all stations or all stations in a certain state) or a space-delimited list of station IDs, as in the above example.

Once a station is started, it will process events from the ET ring or sit in wait mode waiting for more data to appear. A station that has been started is said to be in the *active* state. In order to process data, you would now start the DAQ or the EVIO file producer.

The current status of the stations and the ET system can be checked using:

```
online>status
```

The log data from individual stations can be streamed back to the client using a command such as:

```
online>log 1
```

To exit from the log streaming mode, simply hit any key on your keyboard.

This command will stop all the active stations:

```
online>stop
```

You may also stop stations by their ID:

```
online>stop 1 2 3
```

All inactive stations can be removed using:

```
online>remove
```

Removing a station deletes its work directory and completely removes it from the online reconstruction server.

Similar to the *stop* command, you can also remove stations using their IDs:

```
online>remove 1 2 3
```

Save some plots to an AIDA file:

```
online>save myplots.aida
```

JAS3 can be used to view AIDA plots.

Save some plots to a ROOT file:

```
online>save myplots.root
```


You can use *rootbrowse* command to look at the plots saved in the ROOT format.

Finally, when you'll all done, shutdown the server using a command like:

```
online>shutdown 5
```

This will wait 5 seconds before stopping and destroying all stations and cleanly shutting down the server and its connection to the ET ring.

Shutting Down the Server

 You should always stop the server using the *shutdown* command, rather than killing its system process. Using a command like *kill -9* on the server process may result in the system processes for the stations not being deactivated properly, forcing you to kill them manually.

## Client Command Line

Now you can connect to the running server using the online recon client:

```
hps-recon-client --host localhost --port 22222 -c station.prop
```

An initial configuration can be provided to the client using the *-c* switch.

You can create an initial file called *station.prop* which will usually have settings specific to the data format/year like detector name, run number, reconstruction steering file:

```
lcsim.detector=HPS-PhysicsRun2016-Pass2
lcsim.run=7798
lcsim.steering=/org/hps/steering/recon/PhysicsRun2016OnlineRecon.lcsim
```

These settings are usually going to be specific to the year of the HPS data being read from the ET ring. The above are settings based on the 2016 physics reconstruction.

The client can also be used without a config file (the server may have been started with one to provide valid defaults).

## Using a Command File

```
hps-recon-client --host localhost my.cmd
```

## Viewing Plots

### Showing Plots in a Java Web Application

#### Setup the Web Application

First, install [Apache Tomcat](#) and start it (not covered here). I have used the 9.0.41 release, but any recent version should work fine.

Next, download and build the Online Recon Web App:

```
git clone https://github.com/JeremyMcCormick/hps-online-recon-webapp
cd hps-online-recon-webapp
mvn clean install war:war -DskipTests
```

Now go to the Tomcat manager webpage, which is typically at:  
<http://localhost:8080/manager/html>

Click **Choose File** in the manager and select the war file at *hps-online-recon-webapp/target/HPSRecon.jar* which you just built.

Click **Deploy** to start the webapp. If there is a previous installation of the war file, then first click on **Undeploy** under Applications to remove it, or this step will fail.

Should there be trouble with the application, error messages can be found in the Tomcat log files at *\$TOMCAT\_DIR/logs*

The connection to the remote AIDA server is defined in *hps-online-recon-webapp/src/main/webapp/WEB-INF/web.xml*

The default value is:

```
<context-param>
  <param-name>hps.online.recon.connection</param-name>
  <param-value>//localhost:3001/HPSRecon</param-value>
</context-param>
```

Should the settings on the AIDA server differ from this, the setting must be updated and the webapp redeployed.

To force a reload of cached AIDA data (e.g. at the start of a new run), click **Reload** on the webapp in the admin panel:

## View the Plots in the Browser

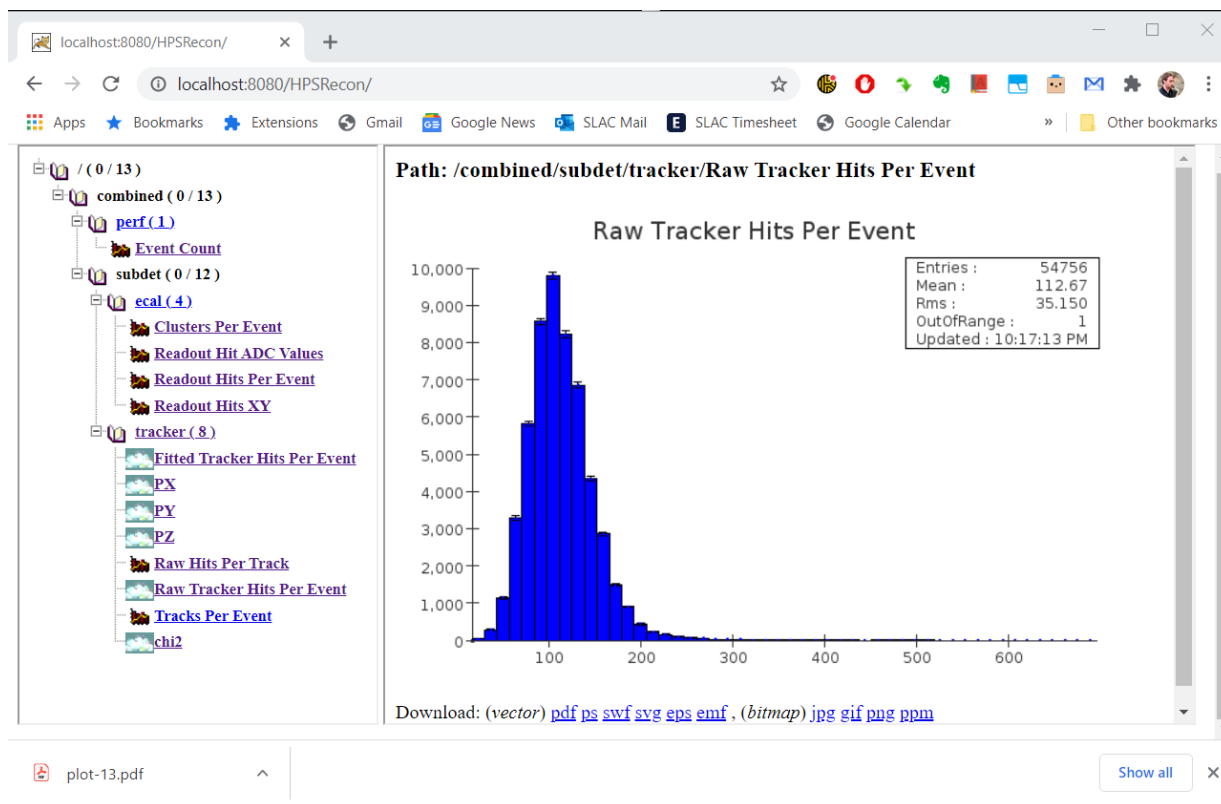
Go to the webpage to view the plots, which would typically be:

<http://localhost:8080/HPSRecon/>

If there is an error on this page, it is likely that the online reconstruction server is not running.

The plots should show in the frame on the left side of the window. They are organized by folders (directories) in the AIDA tree.

Click on one of the plots to show it:



You may also click one of the directories to show a paginated set of thumbnails.

## Viewing plots in JAS3

The remote AIDA tree can be viewed in real-time using JAS3 as a client.

Installation of JAS3 is covered [here](#). You need to install the "Remote AIDA Plugin" from the Plugin Manager as well as the standard AIDA plugin (no other plugins should be required).

Once installed and restarted (ignore any errors from the plugin manager), go to **Tools > Remote AIDA > Connect** and enter the correct settings for your remote AIDA tree:



The "Remote AIDA Connection" dialog box contains the following fields and options:

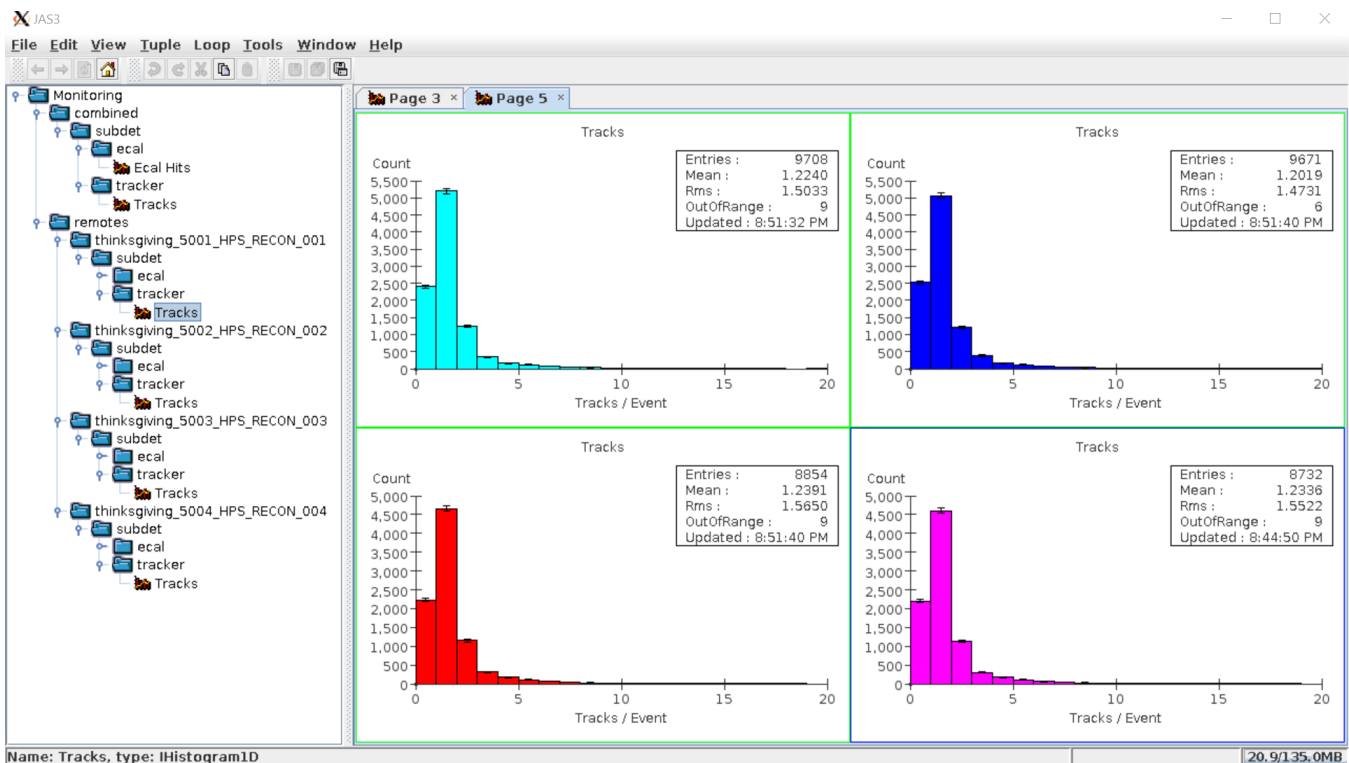
- Name in JAS Tree: HPSRecon
- Server Host Name: localhost
- RMI Bind Name: HPSRecon
- Port: 3001
- ☐ Duplex
- Update (sec): 2.0
- ☐ Recursive ☒ Blocking
- OK and Cancel buttons

Hit OK to load the remote tree.

These are the defaults but this may differ depending on the server's configuration.

Plots should refresh automatically (every 2s by default).

Click on the combined or remote plots in the tree and style them interactively...



TODO: Show how to run a Java main() or Jython script to create AIDA plot pages/regions/styles/etc. from the remote tree data.

ET Software

## Installing the ET Software

If you are running outside Hall B for testing purposes, the CODA ET software is required for running a local ET ring. (This is not needed for actual data runs in Hall B.)

```
# Download and untar sources
wget https://coda.jlab.org/drupal/system/files/et-16.1.tgz
tar -zxvf et-16.1.tgz

# Fix up build files (for some reason this seems to be needed even with Python 2)
cd et-16.1.GIT
2to3-2.7 -w ./coda.py ./SConstruct
autopep8 -i SConstruct
autopep8 -i coda.py

# Build it
export CODA=`pwd`
scons install

# Setup the environment (needs to be done every time you run)
export LD_LIBRARY_PATH=${CODA}/Linux-x86_64/lib
export PATH=${CODA}/Linux-x86_64/bin:${CODA}/Linux-x86_64/bin/examples:$PATH
```

The scons command may not be present in your environment (installation is not covered here but it should be available in your Linux package manager).

## Running the ET Ring

Start the ET ring using a command such as:

```
et_start -rb 8000000 -sb 8000000 -nd -p 11111 -f /tmp/ETBuffer -s 20000 -n 1000 -v -d
```

This starts the ET ring on port 11111 and uses a "standard" location for the swap file (buffer).

The -s argument specifies the max event size.

The -n argument is how many events can be present in the ET ring at once.

The other settings are mainly to try and improve the network performance of the tool by setting generous read/write buffers and using the TCP "no delay" flag.

Experiment with these to find the settings that work best for your machine.

## Running an Event Producer

To do anything useful with the ET ring, we need to stream HPS event data onto it, which can be done using a command like:

```
hps-recon-producer -p 11111 -h localhost -e 1 -f /tmp/ETBuffer -l ./evio_files.txt -s 20000 -d 10
```

The text file contains a list of EVIO files that should be compatible with your configuration settings (e.g. 2016 data for the example settings).

The host, port, and buffer file settings need to match those used to start the ET ring.

## Creating Plots

### Plotting Drivers and Steering Files

x

## Starting a New Run

TODO: Playbook for starting a new run from scratch