Skimming Events

It is often necessary or desirable to select individual events for further analysis. One can select and skim events in both raw (EVIO) and reconstructed (LCIO) data formats by following these steps.

Skimming events in raw EVIO format

Skimming raw EVIO files uses the dedicated main program org.hps.record.evio.EvioEventSkimmer. We currently support skimming events from a single run.

1. Prepare a plain text file containing the event numbers for the events of interest in a single run, one event per line.
2. Prepare a plain text file containing the list of input EVIO files containing the events of interest. The Data Catalog can be used to determine which run partition any particular event is in. These files will have to have been staged and be disk-resident.
3. Run the job using the latest hps-java binary jar file

```
> java -cp hps-distribution-3.11-SNAPSHOT-bin.jar org.hps.record.evio.EvioEventSkimmer -e evioFileList.
txt -s evioEventsToSkim.txt -o skimmedEvents.evio
```

Skimming events in reconstructed LCIO format

Skimming reconstructed LCIO files uses the org.hps.recon.filtering.LcioEventSkimmer Driver and is called in the standard way, summarized here.

1. Prepare a plain text file containing the run and event number for the events of interest. This should be a space-delimited list, with one run-event pair per line.
2. Prepare a plain text file listing the input LCIO files from which to strip the events. The Data Catalog can be used to determine which run partition any particular event is in. These files will have to have been staged and be disk-resident.
3. Prepare a steering file similar to the following:

skimit.lcsim

```xml
<lcsim xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
       xs:noNamespaceSchemaLocation="http://www.lcsim.org/schemas/lcsim/1.0/lcsim.xsd">
    <inputFiles>
      <fileList>${inputFileList}</fileList>
    </inputFiles>
    <control>
        <printInputFiles>true</printInputFiles>
        <printDriversDetailed>true</printDriversDetailed>
    </control>
    <execute>
        <driver name="EventMarkerDriver"/>
        <driver name="LCIOSkimmer"/>
        <driver name="LCIOWriter"/>
    </execute>
    <drivers>
        <driver name="EventMarkerDriver"
                type="org.lcsim.job.EventMarkerDriver">
            <eventInterval>1000</eventInterval>
        </driver>
        <driver name="LCIOSkimmer"
                type="org.hps.recon.filtering.LcioEventSkimmer">
            <runAndEventsToStripFileName>${runAndEventsToSkim}</runAndEventsToStripFileName>
            <debug>true</debug>
        </driver>
        <driver name="LCIOWriter"
                type="org.lcsim.util.loop.LCIODriver">
            <outputFilePath>${outputFile}</outputFilePath>
        </driver>
    </drivers>
</lcsim>
```

4. Run the job using the latest hps-java binary jar file

```
> java -server -jar hps-distribution-3.11-SNAPSHOT-bin.jar -DinputFileList=lcioFileList.txt -
DrunAndEventsToSkim=runAndEventList.txt -DoutputFile=skimmedEvents skimit.lcsim
```

5. A summary will be printed at the end of the job and the selected events will be written to the specified outputFile.

Determining which file partition an event is in

Here's one way to find out the file partition in which a particular event can be found.

Go to the data catalog at : http://hpsweb.jlab.org/datacat/display/datasets/HPS/data/raw
in the *Enter Filter...* field type your search query. For instance, to find

5795 28009772

enter

runMin eq 5795 && runMax eq 5795 && FIRST_PHYSICS_EVENT lt 28009772 && LAST_PHYSICS_EVENT gt 28009772

This should return

[hps_005795.evio.111](#)

Clicking on the link will take you to that partition's entry, where you can find the evio file name. (It would be cool if we could then enable a staging request to be issued from this page as well.) You probably don't want the evio file anyway, but at least now you know which partition to stage to get to the recon or skim file for whatever pass you are interested in.
This is fine for a handful of events, but for bunches of them you most likely want to script this. There is a python interface to the catalog which is documented at:
https://confluence.slac.stanford.edu/display/hpsg/Data+Catalog+Python+Client