Data Quality Management

The data quality management system provides important histograms that allow subsystems experts to check the quality of the data that is being taken soon after it is taken.

The DQM is run using a swif workflow, which is a collection of auger jsub requests. There are 3 phases of each workflow:

1. Run Recon and DQM on the first 10k events in individual evio files. This phase has one job per evio file, and it also takes the longest amount of time per job
2. Sum the DQM files using hadd and then renormalize the SVT occupancy plots.
3. Add all the DQM root files to the data catalog.

To run DQM

example uses run 7373, and reads the files from cache (replace with the number of the run you are using):

1. Check to make sure that the evio files have been copied from the counting house to /cache/mss/hallb/hps/data/ before proceeding. (Note: the script hps2tape.sh, which copies the files to tape, also copies it to cache as well).
2. Check if the DQM workflow already exists for the run before starting the DQM. If it already exists, do not proceed. In this case, DQM is either running, or is on queue to run on the farm.
   - `swif status run-7373-dqm-workflow`
3. Execute the following commands to create the workflow and run it.

   - ssh hps@ifarm
   - cd /group/hps/production/dqm/scripts
   - python mkworkflow.py 7373 --request --cache

This creates a workflow, generates a group of jsub scripts, adds the jsubs to the workflow, and then runs the workflow.

A note from the DQM coordinator: The mkworkflow.py script generates an error message if anyone attempts to execute it from any account other than the hps account. This is intentional, because whichever account starts the workflow owns the dqm files, which only the hps account should own, not somebody's personal account (including the DQM coodinator).

File Locations

The DQM files will be found in

`/work/hallb/hps/physrun2016/pass0/dqm`

An example of a name of a dqm output file name generated from a single evio file is:

`hps_7373.0_dqm_3.6-SNAPSHOT.root`

An example of a combined dqm output file is:

`hps_7373_dqm_3.6-SNAPSHOT.root`

Plots Found in DQM

Folder	Driver	Notes:
EcalClusters	org.hps.analysis.dataquality.EcalMonitoring	contains separate plots for corrected clusters and for pre-corrected clusters
EcalHits	org.hps.analysis.dataquality.EcalMonitoring	
FinalStateMonitoring	org.hps.analysis.dataquality.FinalStateMonitoring	information about reconstructed particles.
HelicalTrackHits	org.hps.analysis.dataquality.TrackingMonitoring	
Tracks	org.hps.analysis.dataquality.TrackingMonitoring	
TrackingResiduals	org.hps.analysis.dataquality.TrackingResiduals	Residuals in time and position of tracks
SVTMonitoring	org.hps.analysis.dataquality.SVTMonitoring	SVT plots including occupancy, amplitude, etc. for each module
TridentMonitoring	org.hps.analysis.dataquality.TridentMonitoring	Details about e+e- vertices, using many cuts
V0Monitoring	org.hps.analysis.dataquality.V0Monitoring	Vertices without any cuts (or minimal cuts). Includes Mollers.

The steering file used for generating the DQM files is /org/hps/steering/production/DataQualityRecon.lcsim .