

Shifter Documentation

The SDHCAL Collaboration ¹

Abstract

This document describe the minimal knowledge of DQM4HEP framework that any shifter should be aware of. A description of the SDHCAL implementation is given : the analysis modules, the processes, the graphical user interfaces, etc ...

¹Corresponding authors:
Ete Remi; `rete@ipnl.in2p3.fr`, Pingault Antoine `antoine.pingault@ugent.be`

Contents

1 The DQM4HEP framework

1

1 The DQM4HEP framework

The DQM4HEP framework provides the tools to monitoring online data taking for a detector or a group of detectors.

The Fig. 1 show the processes needed to monitor the data taking from the DAQ system up to the visualization interfaces for the shifter.

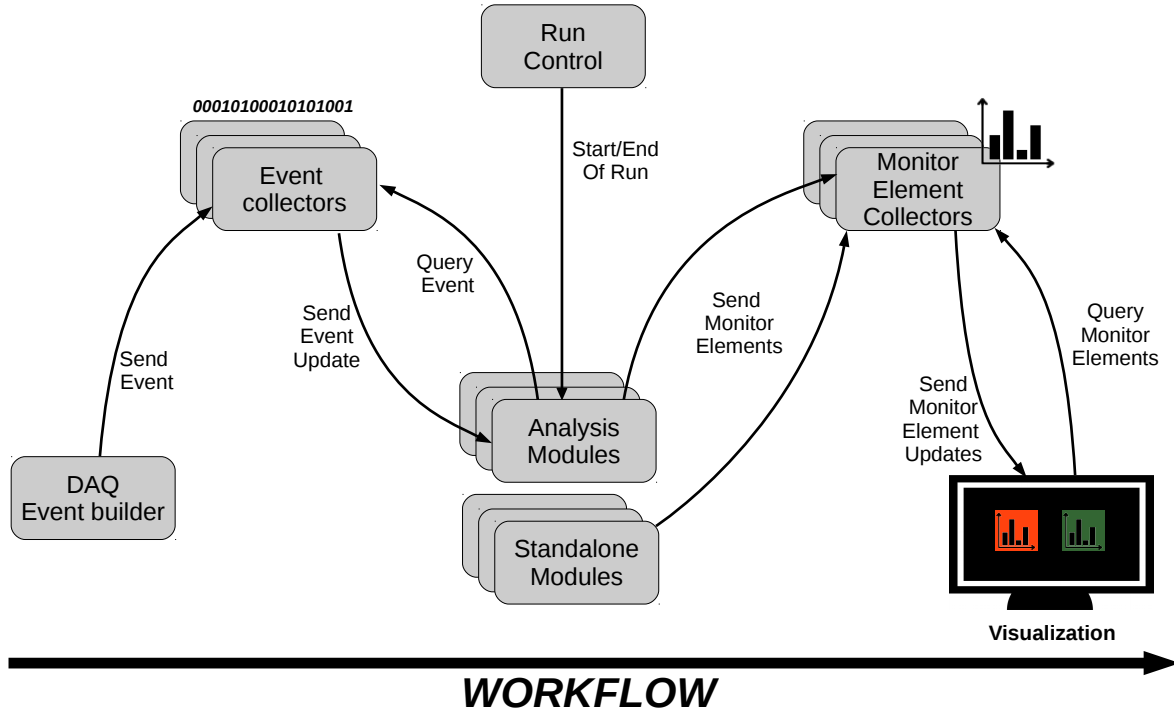


Figure 1: The DQM data workflow, from the DAQ system to visualization clients

First of all, to feed the monitoring system with data, the `levbdim`² software provides functionalities to read data from the shared memory. For the SDHCAL implementation, the DAQ reads data from the shared memory before writing the data to disk. This DAQ process duplicate the data into the shared memory and they are read again by the monitoring system. This process will be labelled as the **shm proxy** afterward in this document.

²See the github page

The monitoring data are then collected by one (or multiple) event collector(s). These server applications are needed to collect the data and make them available to the analysis modules afterward. These processes will be labelled as **event collector**.

Then next step consist in query data and run online analysis. To do that, multiple online analysis are run independently into different processes. These processes will be labelled as **analysis module** afterward. The analyses process events coming from the event collector as soon as the previous event processing is finished and a new event is available. These analysis modules produce histograms, graphs, scalar values, etc ... These elements are labelled as **monitor element**. They encapsulate a ROOT TObject (TH1, TGraph, etc ...) with additional features specific a DQM system such as its quality, quality tests results, etc... To see all the attributes of a monitor element, see the header file. These monitor elements are collected by **monitor element collectors** in order to be distributed to shifters on graphical user interfaces. Events are processed during a **cycle**. Different cycle types are available : i) a **timer cycle**, process data during n seconds, ii) an **event counter cycle**, process n events. At the end of a cycle, the monitor elements are published into the collector. Note that a **run control** must provide the start of run (**SOR**) and end of run (**EOR**) signals to start and stop the analyses.

The last step in the DQM workflow consists in one or multiple client interfaces to monitor the data taking. On these interfaces, clients can query monitor element to watch and can receive frequent update from the system. Multiple canvas areas are provided to draw multiple monitor elements at the same time. This allows to see correlations when, for instance, problem occurs.