Data Quality Monitoring for High Energy Physics (DQM4HEP) Version 03-02-00

R. Été, A. Pingault, L. Mirabito

Université Claude Bernard Lyon 1 - Institut de Physique Nucléaire de Lyon / Ghent University

26 avril 2016

Overview and packaging

DQM4HEP : an online monitoring system for data quality

Key points

- Event distributed system : server/client paradigm
- Set of interfaces for data analysis, adapted to DQM purpose
- Histogram distributed system
- Visualization interface (Qt GUI)
- Large scale remote process management
- Generic IO support for any edm (opt. LCIO)
- Full size HEP experiment to single detector prototype design
- ELog interface

Set of interfaces inspired from CMS DQM system (monitor elements, collectors).

Application flow inspired from ALICE DQM system, AMORE (cycles).

Overview and packaging

DQM4HEP packages

One location : `https://github.com/DQM4HEP`
Webpage : `dqm4hep.github.io`

The main package : DQM4HEP

Installation package for sub-packages (CMake).
Sub-packages :

- **dim** : Distributed Information Management (Delphi). Manage client/server communications
- **dimjc** : DIM Job Control (L. Mirabito). Remote process management using dim.
- **jsoncpp** : Json I/O for dimjc
- **streamlog** : logging library (used in ILCSOFT)
- **DQMCore** : Core part of the DQM system. Client/server interfaces, analysis, IO, run control interface, plugin management ...
- **DQMViz** : Qt visualization interfaces. Job control gui client, monitoring gui client, run control server gui (standalone).
- **LCIO** : Linear Collider IO. Build support for LCIO streamer

Forseen packages :

- **xdrstream** : Generic Xdr serializer
- **xdrlcio** : Lcio serialization using xdrstream (buffer -> socket)
- **DQM4ILC** : ILC specific implementation (detector prototypes modules, marlin helper, ...)

Overview and packaging

Installation

Installation mode

Designed to be built **standalone** or using **ILCSOFT**.
Basic install requires ROOT.
Full install with DQMViz requires Qt and ROOT **compiled with *–enable-qt* option**.

Standalone mode :

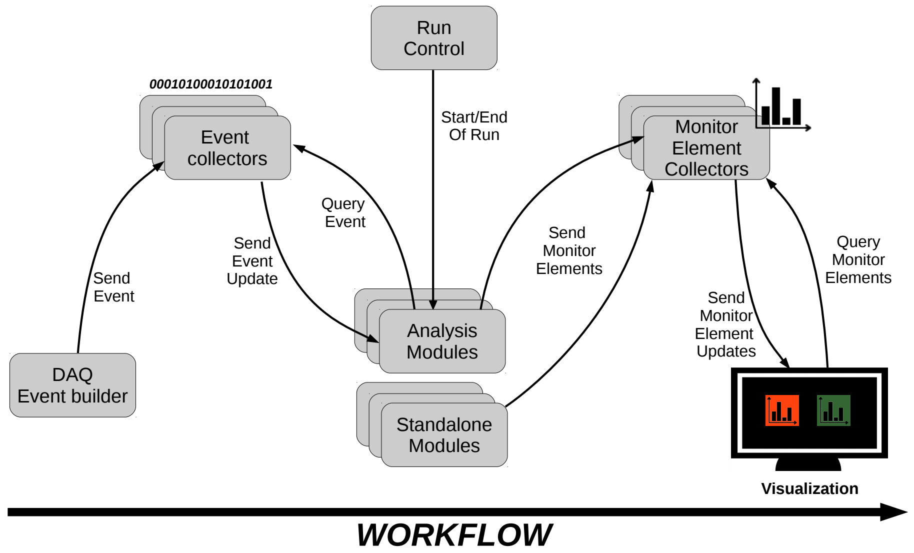
- Basic install : dim, dimjc, jsoncpp, streamlog, DQMCore
- Full install : + DQMViz, LCIO

ILCSOFT mode :

- Basic install : dim, dimjc, jsoncpp, DQMCore
- Full install : + DQMViz

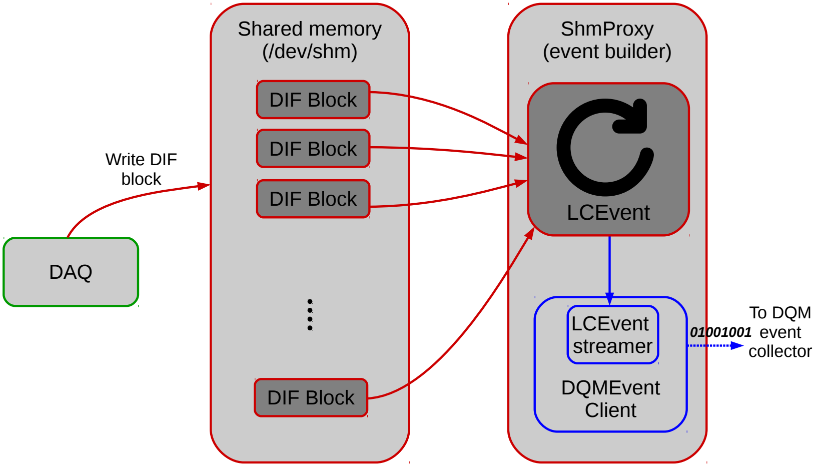
Architecture and API

Global workflow



Architecture and API

DAQ interface example : SDHCAL DAQ interface



DAQ

User specific

DQM framework

Architecture and API

Module applications - analysis module

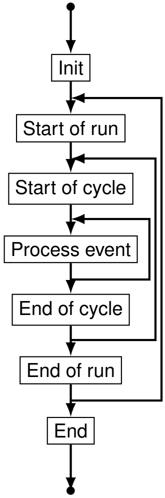
Purpose

- Receive events from a collector server and process them
- Produce monitor elements (histograms, scalars, generic TObject)
- Follow the run control signals (SOR, EOR)

- **Init** : Initialize the application : load dlls, declare services, etc ... Wait for a SOR
- **Start of run** : start cycles loop, open archive
- **Start of cycle** : start a cycle of '*process event*'
- **Process event** : Process incoming event, fill monitor elements, etc ...
- **End of cycle** : send subscribed monitor elements, update archive (opt).
- **End of run** : Wait for SOR, close archive (opt).
- **End** : Clean and exit module.

To implement online DQM analysis, user must implement the `DQMAnalysisModule` interface. A shared library must be build and loaded in the application using the plugin system (`export DQM4HEP_PLUGIN_DLL=libMyModule.so`).

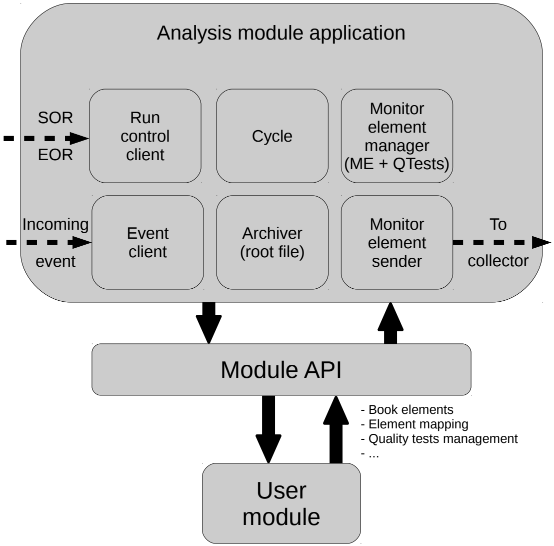
Use `dqm4hep_start_analysis_module` to start an analysis module.



Analysis module application flow

Architecture and API

Module API



Architecture and API

Module API

Monitor element management

- Memory management
- Booking methods (book, delete, from xml)
- Access management via an internal map ("name" -> element)
- Directory management : mkdir, cd, ls, rmdir, pwd

Quality tests

Quality tests used to **evaluate the quality** of a particular monitor element.
Example of quality tests :

- Kolmogorov test with a reference histogram
- χ^2 test after a histogram fit

Quality tests can also be **user defined** (via `DQMQualityTestFactory` class).
Quality tests results (`DQMQualityTestResult` class) stored within monitor element and both **sent to the collector**.
The module API provides functions for :

- User QTests registration
- QTests assignment to monitor elements (also from xml)
- QTest results access (read only)

Architecture and API

Analysis module - Example

```
// ExampleModule.h
class ExampleModule : public DQMAnalysisModule
{
public:
  ExampleModule();  // requiered by plugin system
  ~ExampleModule();

  StatusCode initModule();
  StatusCode readSettings(const TiXmlHandle &
        xmlHandle);
  StatusCode startOfRun(DQMRun *const pRun);
  StatusCode startOfCycle();
  StatusCode processEvent(DQMEvent *const pEvent);
  StatusCode endOfCycle();
  StatusCode endOfRun(DQMRun *const pRun);
  StatusCode endModule();

private:
  DQMMonitorElement    *m_pNHitElement;
};
```

```
// ExampleModule.cc
#include "ExampleModule.h"
// declare plugin in the system
DQM_PLUGIN_DECL( ExampleModule, "ExampleModule" )
// create and enter dir. Book histogram
StatusCode ExampleModule::initModule()
{
  DQMModuleApi::mkdir(this, "/Histograms");
  DQMModuleApi::cd(this, "/Histograms");
  DQMModuleApi::bookIntHistogram1D(this,
        m_pNHitElement, "NHit",
    "Number␣of␣hits", 1200, 0, 1199);
  return STATUS_CODE_SUCCESS;
}
// get lcio event and fill your histogram !
StatusCode ExampleModule::processEvent(DQMEvent *
      const pEvent)
{
  EVENT::LCEvent *pLCEvent =
    pEvent->getEvent<EVENT::LCEvent>();
  if(!pLCEvent)
    return STATUS_CODE_FAILURE;
    // get number of hits from a collection
  m_pNHitElement->get<TH1I>()->Fill(NHit);
  return STATUS_CODE_SUCCESS;
}
```

Architecture and API

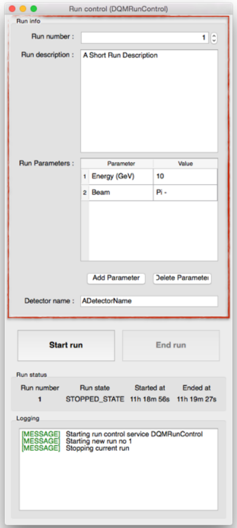
Gui visualisation

Gui interfaces for DQM client developed :

- Run control, job control, online monitoring
- Written with Qt4 framework
- Easily configurable with json and xml.

Architecture and API

Run Control GUI



Run number : 1

Run description : A Short Run Description

Run Parameters :

	Parameter	Value
1	Energy (GeV)	10
2	Beam	Pi -

Detector name : ADetectorName

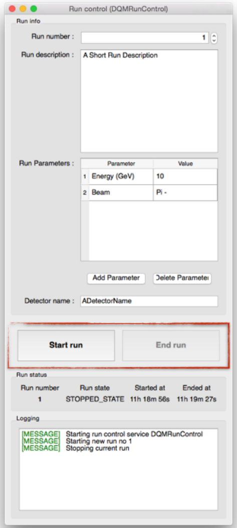
Run number	Run state	Started at	Ended at
1	STOPPED_STATE	11h 18m 56s	11h 19m 27s

```
[MESSAGE] Starting run control service DQMRunControl
[MESSAGE] Starting new run no 1
[MESSAGE] Stopping current run
```

- Parametrisation of run with run number, detector name, run description and parameters

Architecture and API

Run Control GUI



Run number : 1

Run description : A Short Run Description

Run Parameters :

	Parameter	Value
1	Energy (GeV)	10
2	Beam	Pi -

Detector name : ADetectorName

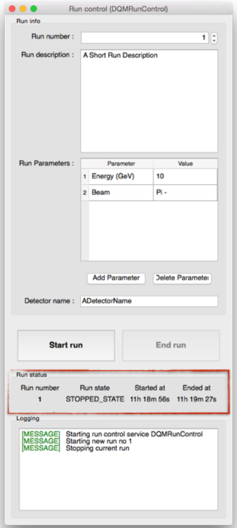
Run number	Run state	Started at	Ended at
1	STOPPED_STATE	11h 18m 56s	11h 19m 27s

```
[MESSAGE] Starting run control service DQMRunControl
[MESSAGE] Starting new run no 1
[MESSAGE] Stopping current run
```

- Parametrisation of run with run number, detector name, run description and parameters
- Send SOR and EOR signals

Architecture and API

Run Control GUI



A Short Run Description

	Parameter	Value
1	Energy (GeV)	10
2	Beam	Pi -

ADetectorName

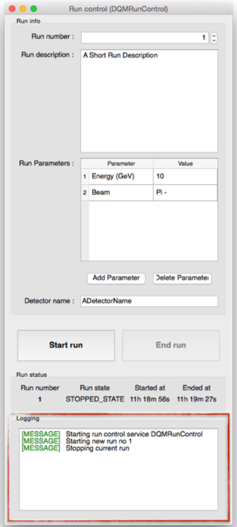
Run number	Run state	Started at	Ended at
1	STOPPED_STATE	11h 18m 56s	11h 19m 27s

```
[MESSAGE] Starting run control service DQMRunControl
[MESSAGE] Starting new run no 1
[MESSAGE] Stopping current run
```

- Parametrisation of run with run number, detector name, run description and parameters
- Send SOR and EOR signals
- Control run status (State, Started/Stopped time)

Architecture and API

Run Control GUI



Run info

Run number : 1

Run description : A Short Run Description

Run Parameters :

	Parameter	Value
1	Energy (GeV)	10
2	Beam	Pi -

Detector name : ADetectorName

Run status

Run number	Run state	Started at	Ended at
1	STOPPED_STATE	11h 18m 56s	11h 19m 27s

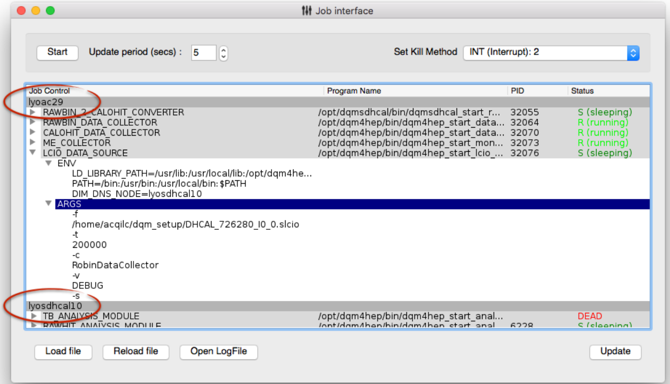
Logging

```
[MESSAGE]  Starting run control service DQMRunControl
[MESSAGE]  Starting new run no 1
[MESSAGE]  Stopping current run
```

- Parametrisation of run with run number, detector name, run description and parameters
- Send SOR and EOR signals
- Control run status (State, Started/Stopped time)
- Every action is logged for easy information overview

Architecture and API

Job Control GUI



Update period (secs) : 5

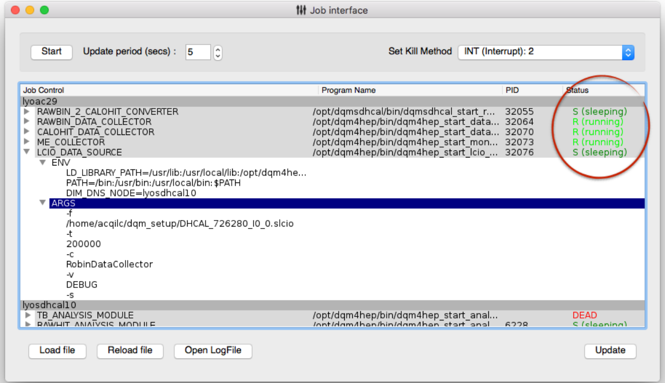
Set Kill Method INT (Interrupt): 2

```
Job Control                              Program Name                              PID     Status
lyoac29
▶ RAWBIN_2_CALOHIT_CONVERTER             /opt/dqmsdhcal/bin/dqmsdhcal_start_r...   32055     S (sleeping)
▶ RAWBIN_DATA_COLLECTOR                  /opt/dqm4hep/bin/dqm4hep_start_data...    32064     R (running)
▶ CALOHIT_DATA_COLLECTOR                 /opt/dqm4hep/bin/dqm4hep_start_data...    32070     R (running)
▶ ME_COLLECTOR                           /opt/dqm4hep/bin/dqm4hep_start_mon...     32073     R (running)
▼ LCIO_DATA_SOURCE                       /opt/dqm4hep/bin/dqm4hep_start_lcio_...   32076     S (sleeping)
   ▼ ENV
        LD_LIBRARY_PATH=/usr/lib:/usr/local/lib:/opt/dqm4he...
        PATH=/bin:/usr/bin:/usr/local/bin:$PATH
        DIM_DNS_NODE=lyosdhcal10
   ▼ ARGS
        -f
        /home/acqilc/dqm_setup/DHCAL_726280_I0_0.slcio
        -t
        200000
        -c
        RobinDataCollector
        -v
        DEBUG
        -s
lyosdhcal10
▶ TB_ANALYSIS_MODULE                     /opt/dqm4hep/bin/dqm4hep_start_anal...              DEAD
▶ RAWHIT_ANALYSIS_MODULE                 /opt/dqm4hep/bin/dqm4hep_start_anal...    6228      S (sleeping)
```

- Load and display a list of applications (Collectors, Modules, etc.) available on different hosts

Architecture and API

Job Control GUI

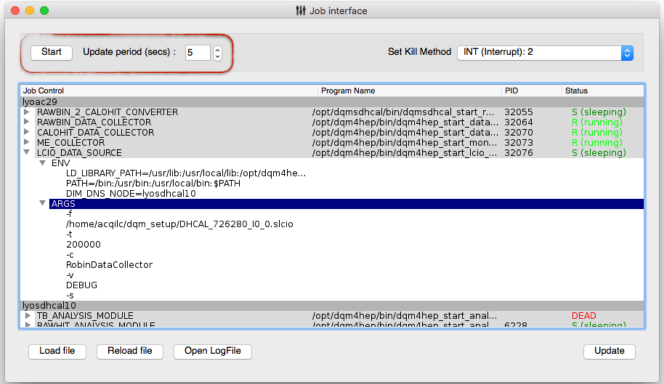


```
Job Control                                Program Name                          PID     Status
lyoac29
▶ RAWBIN_2_CALOHIT_CONVERTER      /opt/dqmsdhcal/bin/dqmsdhcal_start_r...    32055   S (sleeping)
▶ RAWBIN_DATA_COLLECTOR           /opt/dqm4hep/bin/dqm4hep_start_data...     32064   R (running)
▶ CALOHIT_DATA_COLLECTOR          /opt/dqm4hep/bin/dqm4hep_start_data...     32070   R (running)
▶ ME_COLLECTOR                    /opt/dqm4hep/bin/dqm4hep_start_mon...      32073   R (running)
▼ LCIO_DATA_SOURCE                /opt/dqm4hep/bin/dqm4hep_start_lcio_...    32076   S (sleeping)
  ▼ ENV
      LD_LIBRARY_PATH=/usr/lib:/usr/local/lib:/opt/dqm4he...
      PATH=/bin:/usr/bin:/usr/local/bin:$PATH
      DIM_DNS_NODE=lyosdhcal10
  ▼ ARGS
      -f
      /home/acqilc/dqm_setup/DHCAL_726280_I0_0.slcio
      -t
      200000
      -c
      RobinDataCollector
      -v
      DEBUG
      -s
lyosdhcal10
▶ TB_ANALYSIS_MODULE              /opt/dqm4hep/bin/dqm4hep_start_anal...             DEAD
```

- Load and display a list of applications (Collectors, Modules, etc.) available on different hosts
- Displays informations(Name, Host, PID, Status, etc.) about applications

Architecture and API

Job Control GUI

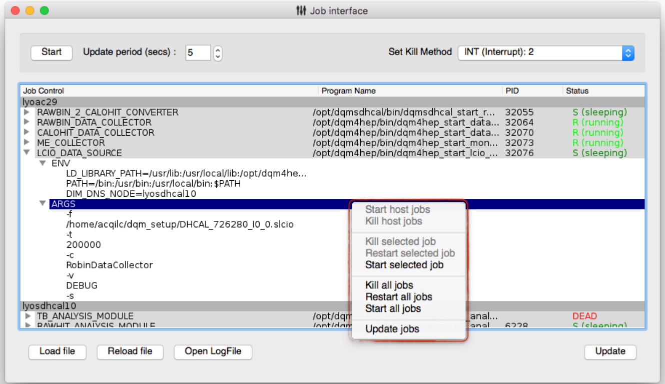


```
Job Control                          Program Name                              PID     Status
lyoac29
▶ RAWBIN_2_CALOHIT_CONVERTER         /opt/dqmsdhcal/bin/dqmsdhcal_start_r...   32055   S (sleeping)
▶ RAWBIN_DATA_COLLECTOR              /opt/dqm4hep/bin/dqm4hep_start_data...    32064   R (running)
▶ CALOHIT_DATA_COLLECTOR             /opt/dqm4hep/bin/dqm4hep_start_data...    32070   R (running)
▶ ME_COLLECTOR                       /opt/dqm4hep/bin/dqm4hep_start_mon...     32073   R (running)
▼ LCIO_DATA_SOURCE                   /opt/dqm4hep/bin/dqm4hep_start_lcio_...   32076   S (sleeping)
  ▼ ENV
      LD_LIBRARY_PATH=/usr/lib:/usr/local/lib:/opt/dqm4he...
      PATH=/bin:/usr/bin:/usr/local/bin:$PATH
      DIM_DNS_NODE=lyosdhcal10
  ▼ ARGS
      -f
      /home/acqilc/dqm_setup/DHCAL_726280_I0_0.slcio
      -t
      200000
      -c
      RobinDataCollector
      -v
      DEBUG
      -s
lyosdhcal10
▶ TB_ANALYSIS_MODULE                 /opt/dqm4hep/bin/dqm4hep_start_anal...            DEAD
▶ RAWHIT_ANALYSIS_MODULE             /opt/dqm4hep/bin/dqm4hep_start_anal...    6228    S (sleeping)
```

- Load and display a list of applications (Collectors, Modules, etc.) available on different hosts
- Displays informations(Name, Host, PID, Status, etc.) about applications
- Infos can be updated in "real time"

Architecture and API

Job Control GUI

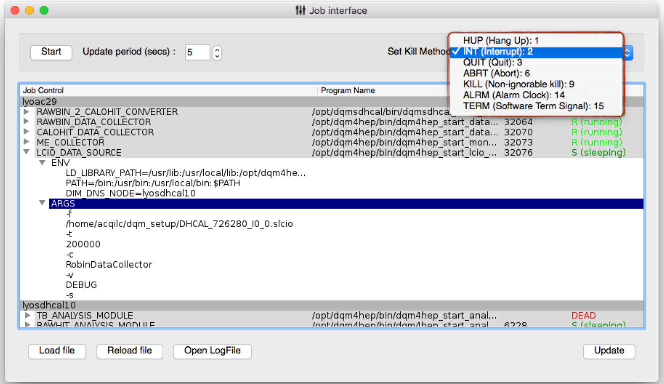


Job Control	Program Name	PID	Status
lyoac29			
▶ RAWBIN_2_CALOHIT_CONVERTER	/opt/dqmsdhcal/bin/dqmsdhcal_start_r...	32055	S (sleeping)
▶ RAWBIN_DATA_COLLECTOR	/opt/dqm4hep/bin/dqm4hep_start_data...	32064	R (running)
▶ CALOHIT_DATA_COLLECTOR	/opt/dqm4hep/bin/dqm4hep_start_data...	32070	R (running)
▶ ME_COLLECTOR	/opt/dqm4hep/bin/dqm4hep_start_mon...	32073	R (running)
▼ LCIO_DATA_SOURCE	/opt/dqm4hep/bin/dqm4hep_start_lcio_...	32076	S (sleeping)
▼ ENV			
LD_LIBRARY_PATH=/usr/lib:/usr/local/lib:/opt/dqm4he...			
PATH=/bin:/usr/bin:/usr/local/bin:$PATH			
DIM_DNS_NODE=lyosdhcal10			
▼ ARGS			
-f			
/home/acqilc/dqm_setup/DHCAL_726280_I0_0.slcio			
-t			
200000			
-c			
RobinDataCollector			
-v			
DEBUG			
-s			
lyosdhcal10			
▶ TB_ANALYSIS_MODULE	/opt/dqm... anal...		DEAD
▶ RAWHIT_ANALYSIS_MODULE	/opt/dqm... anal...	6228	S (sleeping)

- Load and display a list of applications (Collectors, Modules, etc.) available on different hosts
- Displays informations(Name, Host, PID, Status, etc.) about applications
- Infos can be updated in "real time"
- Manage Applications (Start/Kill/Restart) with contextual menu

Architecture and API

Job Control GUI

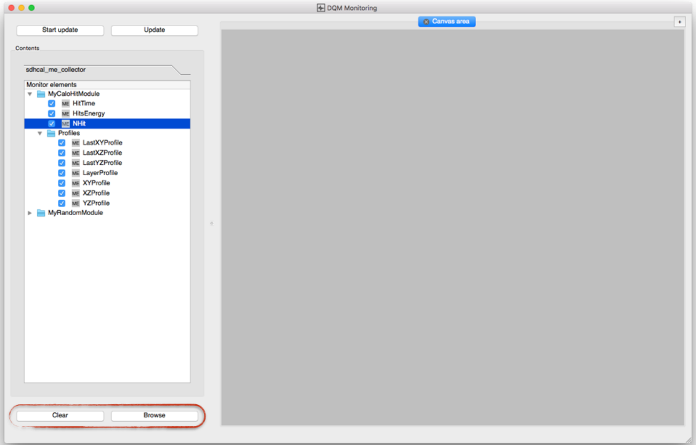


```
Job Control                          Program Name
lyoac29
▶ RAWBIN_2_CALOHIT_CONVERTER        /opt/dqmsdhcal/bin/dqmsdhca
▶ RAWBIN_DATA_COLLECTOR             /opt/dqm4hep/bin/dqm4hep_start_data...  32064  R (running)
▶ CALOHIT_DATA_COLLECTOR            /opt/dqm4hep/bin/dqm4hep_start_data...  32070  R (running)
▶ ME_COLLECTOR                      /opt/dqm4hep/bin/dqm4hep_start_mon...   32073  R (running)
▼ LCIO_DATA_SOURCE                  /opt/dqm4hep/bin/dqm4hep_start_lcio_... 32076  S (sleeping)
   ▼ ENV
        LD_LIBRARY_PATH=/usr/lib:/usr/local/lib:/opt/dqm4he...
        PATH=/bin:/usr/bin:/usr/local/bin:$PATH
        DIM_DNS_NODE=lyosdhcal10
   ▼ ARGS
        -f
        /home/acqilc/dqm_setup/DHCAL_726280_I0_0.slcio
        -t
        200000
        -c
        RobinDataCollector
        -v
        DEBUG
        -s
lyosdhcal10
▶ TB_ANALYSIS_MODULE                /opt/dqm4hep/bin/dqm4hep_start_anal...         DEAD
▶ RAWHIT_ANALYSIS_MODULE            /opt/dqm4hep/bin/dqm4hep_start_anal...  6228   S (sleeping)
```

- Load and display a list of applications (Collectors, Modules, etc.) available on different hosts
- Displays informations(Name, Host, PID, Status, etc.) about applications
- Infos can be updated in "real time"
- Manage Applications (Start/Kill/Restart) with contextual menu
- Kill method can be adjusted

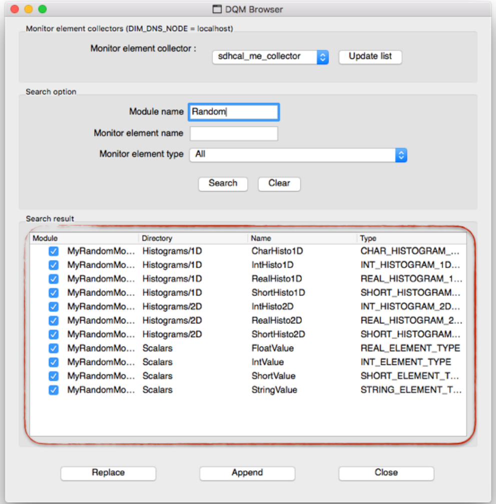
Architecture and API

Monitoring Gui + Browser



Architecture and API

Monitoring Gui + Browser



Monitor element collector : sdhcal_me_collector

Module name: Random

Monitor element name:

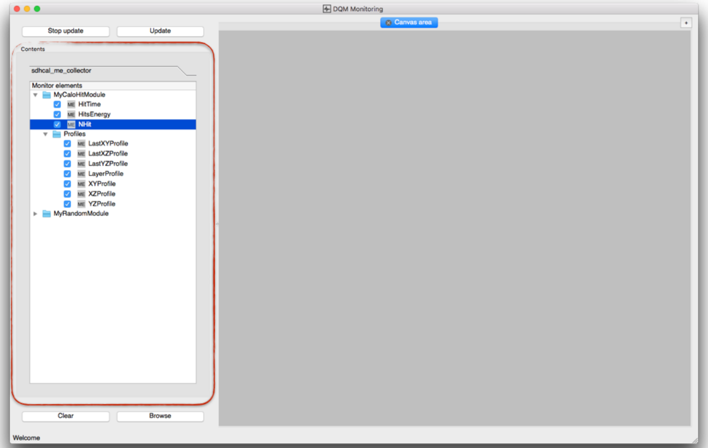
Monitor element type: All

	Module	Directory	Name	Type
✓	MyRandomMo...	Histograms/1D	CharHisto1D	CHAR_HISTOGRAM_...
✓	MyRandomMo...	Histograms/1D	IntHisto1D	INT_HISTOGRAM_1D...
✓	MyRandomMo...	Histograms/1D	RealHisto1D	REAL_HISTOGRAM_1...
✓	MyRandomMo...	Histograms/1D	ShortHisto1D	SHORT_HISTOGRAM...
✓	MyRandomMo...	Histograms/2D	IntHisto2D	INT_HISTOGRAM_2D...
✓	MyRandomMo...	Histograms/2D	RealHisto2D	REAL_HISTOGRAM_2...
✓	MyRandomMo...	Histograms/2D	ShortHisto2D	SHORT_HISTOGRAM...
✓	MyRandomMo...	Scalars	FloatValue	REAL_ELEMENT_TYPE
✓	MyRandomMo...	Scalars	IntValue	INT_ELEMENT_TYPE
✓	MyRandomMo...	Scalars	ShortValue	SHORT_ELEMENT_T...
✓	MyRandomMo...	Scalars	StringValue	STRING_ELEMENT_T...

- Browser to build histograms selections to display
- Search Function to refine selection

Architecture and API

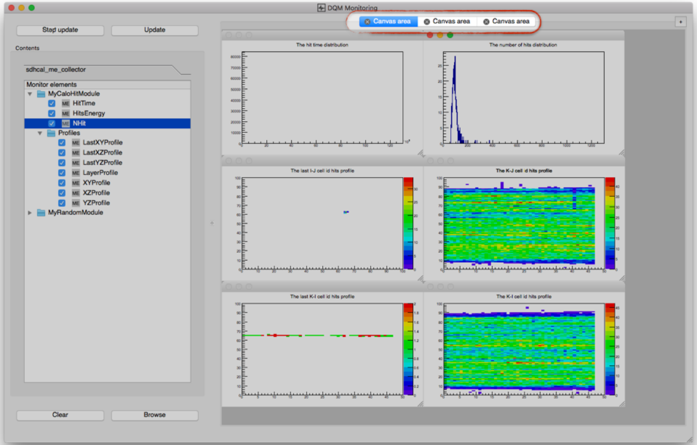
Monitoring Gui + Browser



- List of histograms added from Browser

Architecture and API

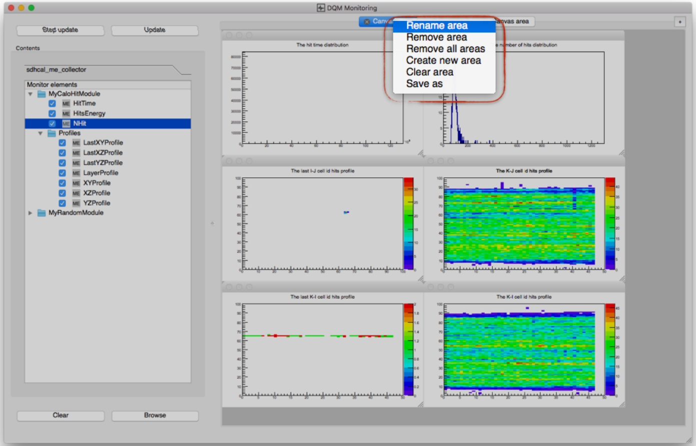
Monitoring Gui + Browser



- Multiple canvas area available

Architecture and API

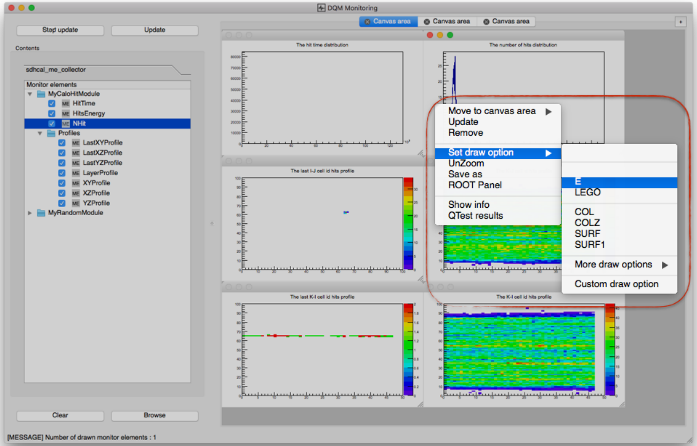
Monitoring Gui + Browser



- Multiple canvas area available

Architecture and API

Monitoring Gui + Browser

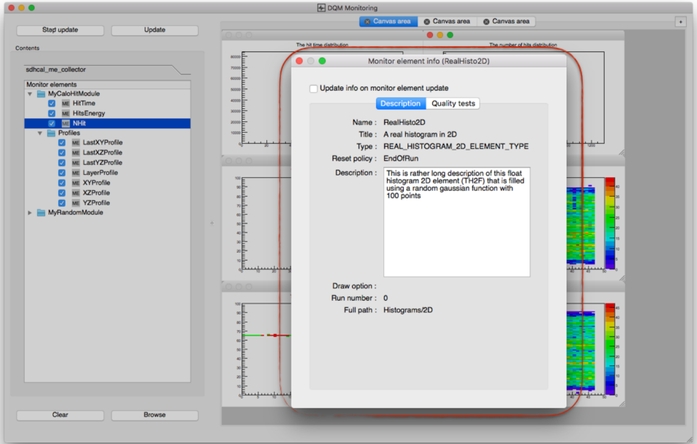


[MESSAGE] Number of drawn monitor elements : 1

- Multiple canvas area available
- Real ROOT histograms (Can be fitted, zoomed, etc.)

Architecture and API

Monitoring Gui + Browser

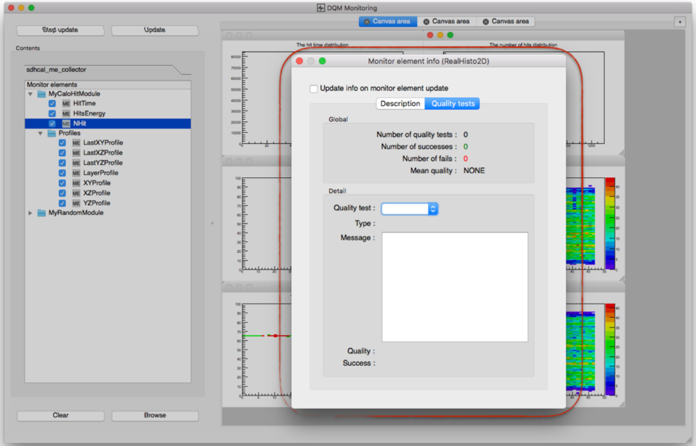


Name : RealHisto2D
Title : A real histogram in 2D
Type : REAL_HISTOGRAM_2D_ELEMENT_TYPE
Reset policy : EndOfRun
Description : This is rather long description of this float histogram 2D element (TH2F) that is filled using a random gaussian function with 100 points
Draw option :
Run number : 0
Full path : Histograms/2D

- Multiple canvas area available
- Real ROOT histograms (Can be fitted, zoomed, etc.)
- Histograms descriptions and Quality

Architecture and API

Monitoring Gui + Browser



Monitor element info (RealHisto2D)

Update info on monitor element update

Global

Number of quality tests : 0
Number of successes : 0
Number of fails : 0
Mean quality : NONE

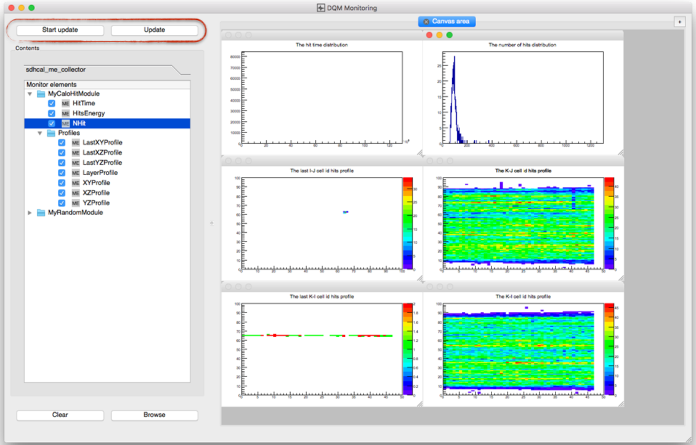
Detail

Quality test :
Type :
Message :
Quality :
Success :

- Multiple canvas area available
- Real ROOT histograms (Can be fitted, zoomed, etc.)
- Histograms descriptions and Quality

Architecture and API

Monitoring Gui + Browser



- Multiple canvas area available
- Real ROOT histograms (Can be fitted, zoomed, etc.)
- Auto Update

Conclusion and plans

Conclusions and plans

Conclusion and plans

- Independent processes decoupled and linked using networking.
- Plugins (modules, data streaming) to configure and run the system.
- Tools for data feed in the system from the DAQ (event client interface)
- GUIs to control/monitor the system.
- Tests are OK but need numbers !
- ILCSOFT release ?

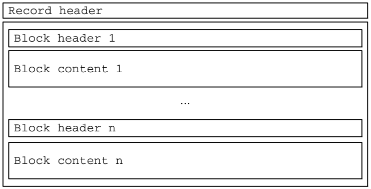
Current work and plans

- Full implementation of SDHCAL DQM
- Combined ECAL test-beam -> combined ECAL-HCAL DQM as proof of concept
- EUDAQ binding (T. Coates, Sussex, UK)

Backups

xdrstream and xdrlcio

- xdrstream github : `https://github.com/DQM4HEP/xdrstream`
  Serialization with XDR format to read/write raw data into different devices (file, buffer, socket, user defined)



  Globally -> SIO implementation :
  - without static API (SIO_blockManager, SIO_streamManager, etc)
  - with different device implementation
  - external standalone package

  For the moment, only buffer implementation (`xdrstream::BufferDevice`)
- xdrlcio github : `https://github.com/DQM4HEP/xdrlcio`
  LCIO edm serialization using xdrstream.
  •`XdrLcio::writeEvent(const EVENT::LCEvent *pLCEvent, xdrstream::IODevice *const pDevice)`
  and
  •`XdrLcio::readNextEvent(xdrstream::IODevice *const pDevice)`
  + many useful functions