

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

23 février 2016



Université Claude Bernard



Lyon 1



1 Overview and packaging

2 Architecture and API

3 Conclusion and plans

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 DQMVis requires Qt and ROOT **compiled with *-enable-qt* option**.

Standalone mode :

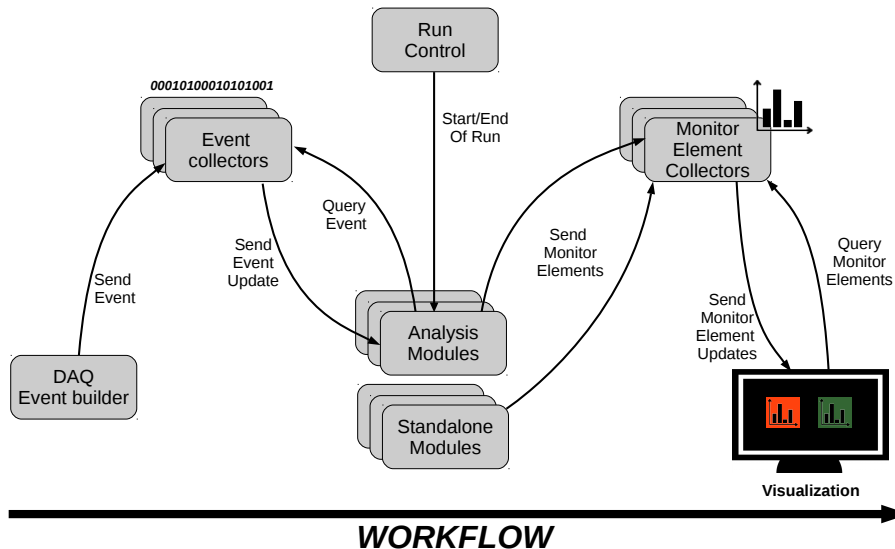
- Basic install : dim, dimjc, jsoncpp, streamlog, DQMCore
- Full install : + DQMVis, LCIO

ILCSOFT mode :

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

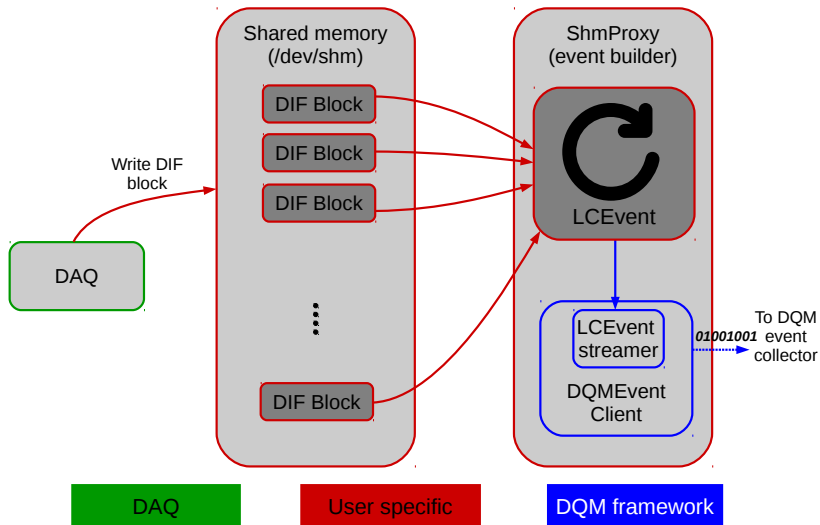
Architecture and API

Global workflow



Architecture and API

DAQ interface example : SDHCAL DAQ interface



Architecture and API

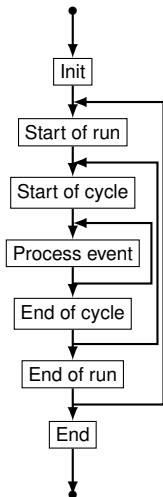
Module applications - analysis module

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`).

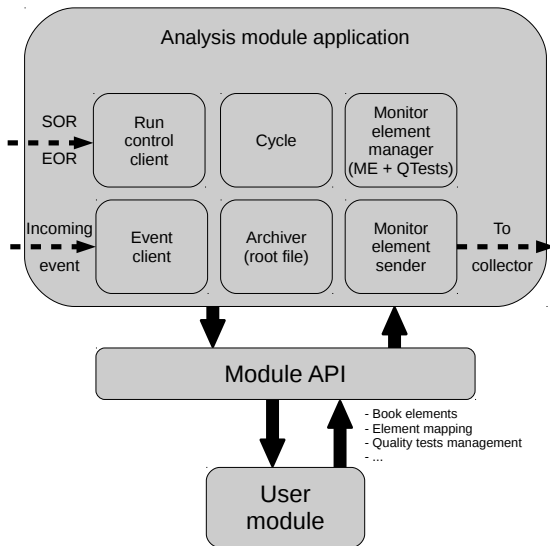
Use `dqm4hep_start_analysis_module` to start an analysis module.



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 DQMAalysisModule
{
public:
    ExampleModule(); // required 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:
    DQMModuleElement *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<TH1>()->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 control (DQMRunControl)

Run info

Run number :

Run description : A Short Run Description

Run Parameters :

| | Parameter | Value |
|---|--------------|-------|
| 1 | Energy (GeV) | 10 |
| 2 | Beam | P1 - |

Detector name :

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

Architecture and API

Run Control GUI

Run control (DQMRunControl)

Run info

Run number : 1

Run description : A Short Run Description

| Run Parameters : | Parameter | Value |
|------------------|--------------|-------|
| 1 | Energy (GeV) | 10 |
| 2 | Beam | P1 - |

Add Parameter Delete Parameter

Detector name : ADetectorName

Start run End run

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

Architecture and API

Run Control GUI

Run control (DQMRunControl)

Run info

Run number : 1

Run description : A Short Run Description

Run Parameters :

| Parameter | Value |
|----------------|-------|
| 1 Energy (GeV) | 10 |
| 2 Beam | P1 - |

Add Parameter Delete Parameter

Detector name : A:DetectorName

Start run End run

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)

Architecture and API

Run Control GUI

Run control (DQMRunControl)

Run info

Run number : 1

Run description : A Short Run Description

Run Parameters :

| Parameter | Value |
|----------------|-------|
| 1 Energy (GeV) | 10 |
| 2 Beam | P1 - |

Add Parameter Delete Parameter

Detector name : A:DetectorName

Start run End run

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

The screenshot shows a window titled "Job interface" with a control panel at the top and a table of processes below. The control panel includes a "Start" button, an "Update period (secs)" dropdown set to 5, and a "Set Kill Method" dropdown set to "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/DHICAL_726280_i0_0.slcio | | | |
| -t | | | |
| 200000 | | | |
| -c | | | |
| RobinDataCollector | | | |
| -v | | | |
| DEBUG | | | |
| -s | | | |
| lyosdhcal10 | | | |
| ▶ TB_ANALYSIS_MODULE | /opt/dqm4hep/bin/dqm4hep_start_anal... | | DEAD |
| ▶ RWBHIT_ANALYSIS_MODULE | /opt/dqm4hep/bin/dqm4hep_start_anal... | 6328 | S (sleeping) |

At the bottom of the window, there are buttons for "Load file", "Reload file", "Open LogFile", and "Update".

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

Architecture and API

Job Control GUI

The screenshot shows a window titled "Job interface" with a control panel at the top. The control panel includes a "Start" button, an "Update period (secs)" field set to 5, and a "Set Kill Method" dropdown menu set to "INT (Interrupt): 2".

The main area displays a table with the following columns: Job Control, Program Name, PID, and Status. The table lists several processes under the "lyoac29" job control:

| Job Control | Program Name | PID | Status |
|------------------------------|---|-------|--------------|
| ▶ RAWBIN_2_CALOHIT_CONVERTER | /opt/dqmsdhcal/bin/dqmsdhcal_start_f... | 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) |

Below the table, the "ENV" and "ARGS" sections are expanded, showing environment variables like LD_LIBRARY_PATH, PATH, and DIM_DNS_NODE, and command-line arguments like /home/acqilc/dqm_setup/DHICAL_726280_I0_0_slcio and 200000. At the bottom of the table, a process under "lyosdhcal10" is shown as "DEAD".

At the bottom of the window, there are buttons for "Load file", "Reload file", "Open LogFile", and "Update".

- 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

The screenshot shows a window titled "Job interface" with a "Start" button and a "Set Kill Method" dropdown set to "INT (Interrupt): 2". The "Update period (secs)" is set to 5. Below is a table of job control information:

| 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/DHICAL_726280_i0_0.slcio | | |
| -t | 200000 | | |
| -c | RobinDataCollector | | |
| -v | DEBUG | | |
| -s | | | |
| lyosdhcal10 | | | |
| ▶ TB_ANALYSIS_MODULE | /opt/dqm4hep/bin/dqm4hep_start_anal... | 6328 | DEAD |
| ▶ RAWBIN_ANALYSIS_MODULE | /opt/dqm4hep/bin/dqm4hep_start_anal... | | S (sleeping) |

At the bottom, there are buttons for "Load file", "Reload file", "Open LogFile", and "Update".

- 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

The screenshot shows a window titled "Job interface" with a "Job Control" section. At the top, there are controls for "Start", "Update period (secs) : 5", and "Set Kill Method INT (Interrupt): 2". Below this is a table of jobs:

| 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/acqlic/dqm_setup/DHICAL_726280_10_0_slcio | | |
| -t | 200000 | | |
| -c | RobinDataCollector | | |
| -v | DEBUG | | |
| -s | | | |
| lyosdhcal10 | | | |
| ▶ TB_ANALYSIS_MODULE | /opt/dqm... | anal... | DEAD |
| ▶ RAWBIN_ANALYSIS_MODULE | /opt/dqm... | anal... 6328 | S (cleaning) |

A contextual menu is open over the "ARGS" section, listing actions: Start host jobs, Kill host jobs, Kill selected job, Restart selected job, Start selected job, Kill all jobs, Restart all jobs, Start all jobs, and Update jobs.

At the bottom of the window, there are buttons for "Load file", "Reload file", "Open LogFile", and "Update".

- 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

The screenshot shows a window titled "Job interface" with a "Start" button and an "Update period (secs)" set to 5. A "Set Kill Method" dropdown menu is open, showing options: HUP (Hang Up): 1, QUIT (Interrupt): 2 (selected), QUIT (Quit): 3, ABRT (Abort): 6, KILL (Non-ignorable kill): 9, ALRM (Alarm Clock): 14, and TERM (Software Term Signal): 15.

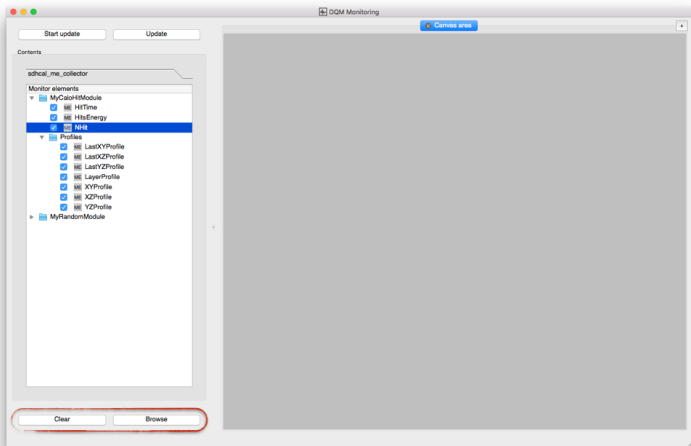
| Job Control | Program Name | PID | Status |
|------------------------------|--|-------|--------------|
| lyac29 | | | |
| ▶ RAWBIN_2_CALOHIT_CONVERTER | /opt/dqmsdhcal/bin/dqmsdhca... | | |
| ▶ RAWBIN_DATA_COLLECTOR | /opt/dqm4hep/bin/dqm4hep_start_data... | 32064 | R (running) |
| ▶ CALOHIT_DATA_COLLECTOR | /opt/dqm4hep/bin/dqm4hep_start_data... | 32070 | R (running) |
| ▶ ME_COLLECTOR | /opt/dqm4hep/bin/dqm4hep_start_mon... | 32073 | R (running) |
| ▼ LCLG_DATA_SOURCE | /opt/dqm4hep/bin/dqm4hep_start_lclg... | 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 | | | |
| -j | /home/acqilc/dqm_setup/DHICAL_726280_I0_0.slcio | | |
| -t | 200000 | | |
| -c | RobinDataCollector | | |
| -y | DEBUG | | |
| -s | | | |
| lyosdhcal10 | | | |
| ▶ TB_ANALYSIS_MODULE | /opt/dqm4hep/bin/dqm4hep_start_anal... | | DEAD |
| ▶ RAWBIN_ANALYSIS_MODULE | /opt/dqm4hep/bin/dqm4hep_start_anal... | 6228 | S (sleeping) |

Buttons at the bottom: Load file, Reload file, Open LogFile, Update.

- 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 collectors (DIM_DNS_NODE = localhost)

Monitor element collector :

Search option

Module name

Monitor element name

Monitor element type

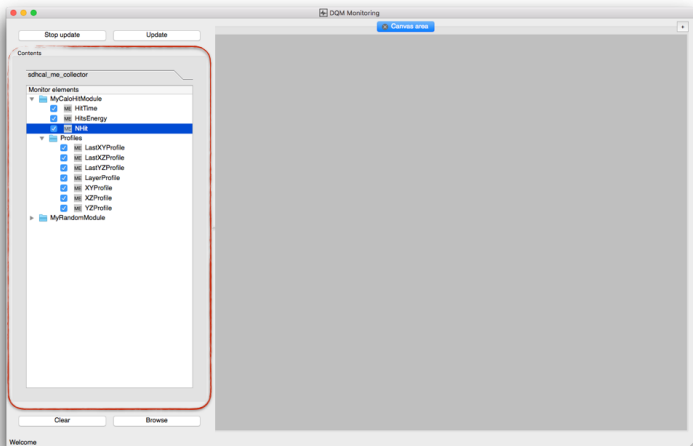
Search result

| Module | Directory | Name | Type | |
|-------------------------------------|---------------|---------------|---------------|---------------------|
| <input checked="" type="checkbox"/> | MyRandomMo... | Histograms/1D | Char-Histo1D | CHAR_HISTOGRAM_... |
| <input checked="" type="checkbox"/> | MyRandomMo... | Histograms/1D | Int-Histo1D | INT_HISTOGRAM_1D... |
| <input checked="" type="checkbox"/> | MyRandomMo... | Histograms/1D | Reall-Histo1D | REAL_HISTOGRAM_1... |
| <input checked="" type="checkbox"/> | MyRandomMo... | Histograms/1D | Short-Histo1D | SHORT_HISTOGRAM... |
| <input checked="" type="checkbox"/> | MyRandomMo... | Histograms/2D | Int-Histo2D | INT_HISTOGRAM_2D... |
| <input checked="" type="checkbox"/> | MyRandomMo... | Histograms/2D | Reall-Histo2D | REAL_HISTOGRAM_2... |
| <input checked="" type="checkbox"/> | MyRandomMo... | Histograms/2D | Short-Histo2D | SHORT_HISTOGRAM... |
| <input checked="" type="checkbox"/> | MyRandomMo... | Scalars | FloatValue | REAL_ELEMENT_TYPE |
| <input checked="" type="checkbox"/> | MyRandomMo... | Scalars | IntValue | INT_ELEMENT_TYPE |
| <input checked="" type="checkbox"/> | MyRandomMo... | Scalars | ShortValue | SHORT_ELEMENT_T... |
| <input checked="" type="checkbox"/> | MyRandomMo... | Scalars | StringValue | STRING_ELEMENT_T... |

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

Architecture and API

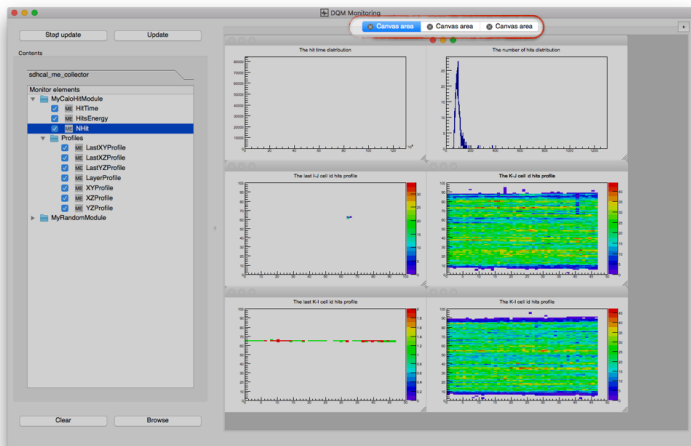
Monitoring Gui + Browser



- List of histograms added from Browser

Architecture and API

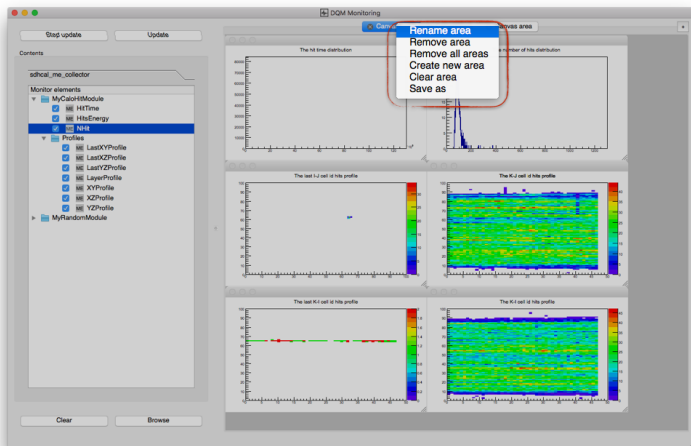
Monitoring Gui + Browser



- Multiple canvas area available

Architecture and API

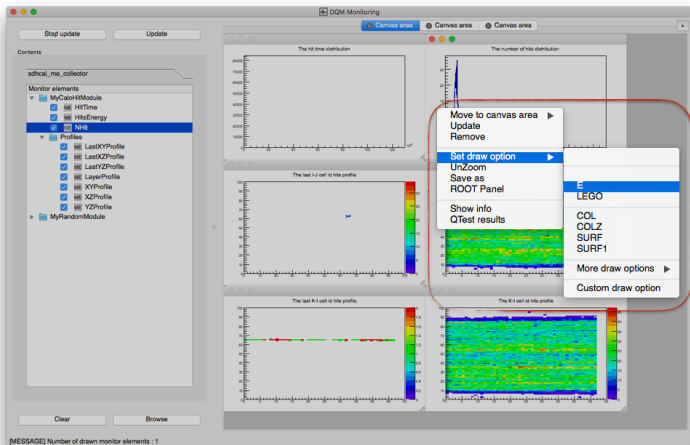
Monitoring Gui + Browser



- Multiple canvas area available

Architecture and API

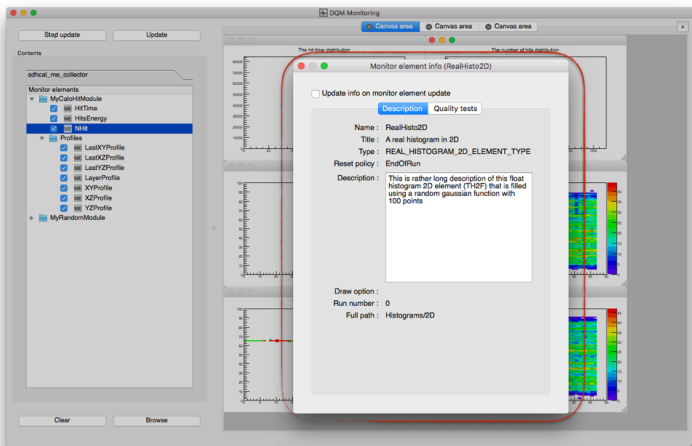
Monitoring Gui + Browser



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

Architecture and API

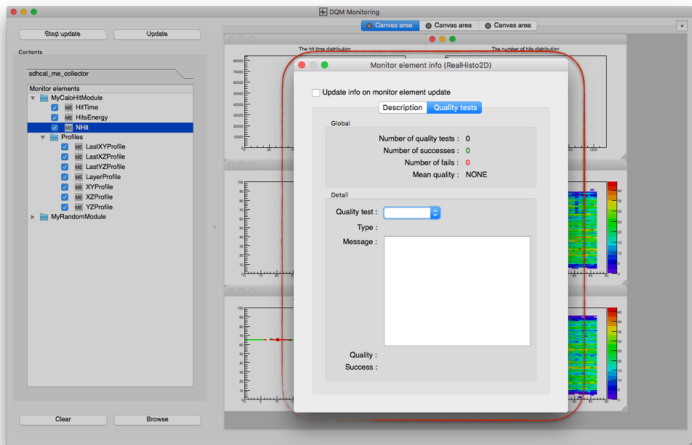
Monitoring Gui + Browser



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

Architecture and API

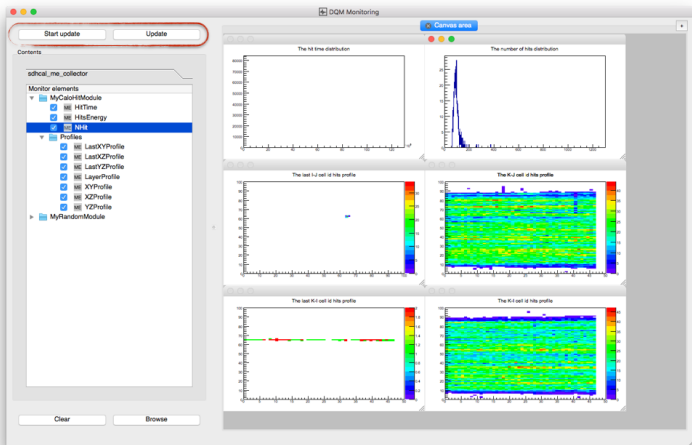
Monitoring Gui + Browser



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

- 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 ?

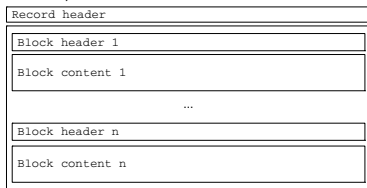
Current work and plans

- 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