GEM Light DQM Reference Guide

Robert King, Mykhailo Dalchenko
/ CMS / Texas A&M

Contents

1   COMPONENTS

1.1   OVERVIEW

There are multiple parts to the GEM Light Data Quality Monitoring software. They can be summarized in the graphic below, and are explained in further detail in the following subsections.
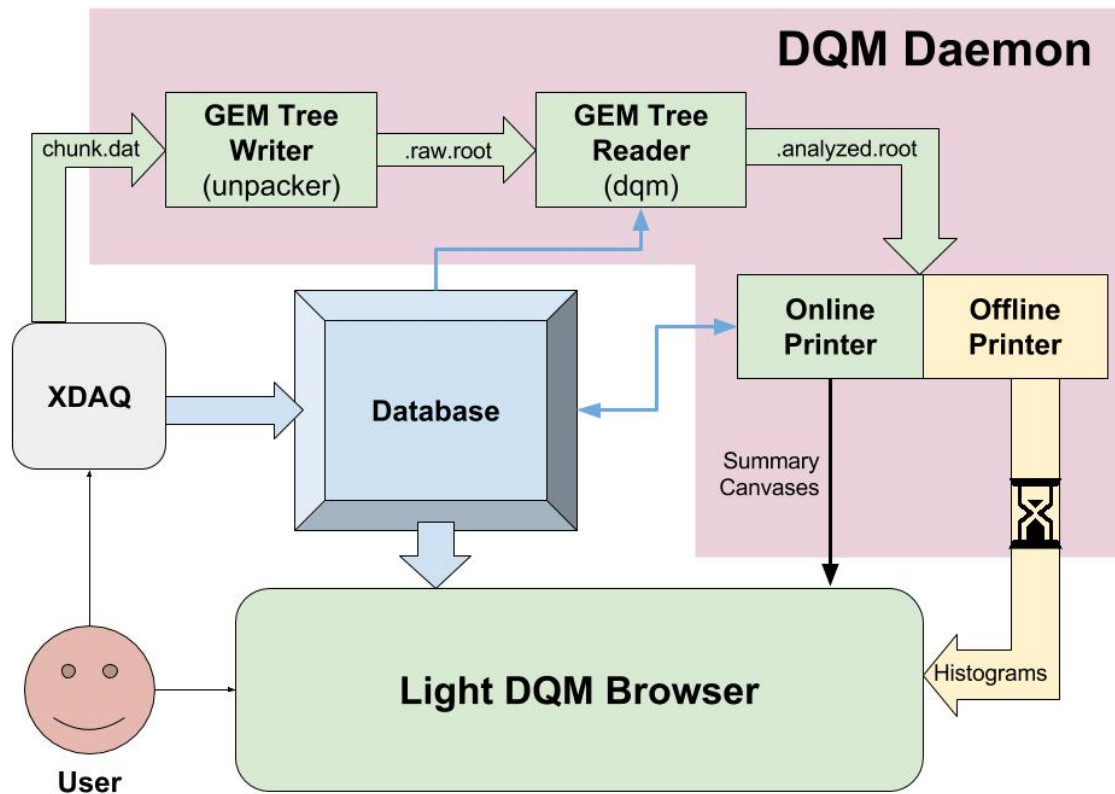
FIGURE 1: LIGHT DQM SYSTEM OVERVIEW

1.2   GEM TREE WRITER (UNPACKER)

Takes raw data file, processes it into a GEM tree, and puts it into a raw.root file to be processed by DQM separately.

1.3   GEM TREE READER (DQM)

Takes GEM tree in raw.root file, processes the tree, and fills histograms with useful information in analyzed.root file.

1.4   GEM TREE ONLINE PRINTER (ONLINEPRINTER)

Uses analyzed.root to quickly print summary canvases containing useful information to view while data is being taken to monitor status.

1.5   GEM TREE OFFLINE PRINTER (GTPRINTER)

Loops through all histograms and canvases in analyzed.root and prints each one as .JSON files to be viewed in the LDQM Browser. Takes significantly longer than the online printer (currently on the order of several minutes depending on size of .raw.root file).

1.6   DAEMON

As data is produced by the XDAQ application, the daemon can be run in the background to process this data through the DQM steps automatically, essentially making the DQM software completely online. This is done through chunking - where data is produced in chunks (of 1000 events each), then processed a single chunk at a time and combined with the previous chunks. The end result is the ability to see DQM histograms in the Light DQM Browser during data-taking.

   The DQM Daemon automatically begins when the Light DQM Browser is started. It looks for data chunks from the most recent run in the database and immediately processes them if it finds any.

1.7   LIGHT DQM BROWSER

View printed DQM histograms in interactive .JSON format with a user-friendly, organized GUI.

1.8   DATABASE

Currently using MySQL database hosted on gem904daq01. Plan to transition to Central Database soon. Database holds all run information:

- Run Name

- Run Number

- Run Date

- Run Type

- Run Station

- Hardware Configuration (for each run)

- Hardware State (auto-updating)

2 GEM LIGHT DQM

2.1 CONFIGURATION AND ENVIRONMENTAL VARIABLES

ADD $BUILD_HOME TO .BASHRC (OR EQUIVALENT)

```
export $BUILD_HOME=/home/...
```

   The most important environmental variable is $BUILD_HOME. Set this variable to wherever you plan to checkout all of the GEM DQM Code. All other environmental variables should be available from scripts inside the software packages. Proper connection to the Database is required for all parts of the Light DQM. The database stores mandatory information about runs and hardware configurations which the code cannot work without. The code will automatically try to connect to the Database for you.

2.2 GETTING CODE - GITHUB

2.2.1 REPOSITORIES

GEM Online Software Suite: https://github.com/cms-gem-daq-project/cmsgemos.git
GEM Light DQM: https://github.com/cms-gem-daq-project/gem-light-dqm.git
Light DQM Browser: https://github.com/cms-gem-daq-project/ldqm-browser.git

2.2.2 COMPILING

COMPILING ONLINE SOFTWARE SUITE

```
cd $BUILD_HOME
git clone https://github.com/cms-gem-daq-project/cmsgemos.git
source $BUILD_HOME/cmsgemos/setup/paths.sh
cd cmsgemos
make all -j8

Ready to run
```

COMPILING DQM SUITE

```
cd $BUILD_HOME
git clone https://github.com/cms-gem-daq-project/gem-light-dqm.git
source $BUILD_HOME/cmsgemos/setup/paths.sh
cd gem-light-dqm
source compile.sh
```

Ready to run

LIGHT DQM BROWSER - NO COMPILATION NECESSARY

```
cd $BUILD_HOME
git clone https://github.com/cms-gem-daq-project/ldqm-browser.git
source $BUILD_HOME/cmsgemos/setup/paths.sh
source $BUILD_HOME/ldqm-browser/setup.sh
cd ldqm-browser/LightDQM
```

Ready to run

The light DQM browser does not require any compilation, but you **must** source setup.sh before running the browser.

2.3 XDAQ APPLICATION

Talk to Jared Sturdy... needs its own reference

2.4 RUNNING LIGHT DQM

The light DQM can either be called manually by running the individual files through the DQM components illustrated in Figure 1, or can be done automatically using the DQM Daemon.

2.4.1 USING DAEMON (RECOMMENDED)

The Daemon is a process that runs in the background when one configures the Light DQM Browser:

RUNNING LIGHT DQM DAEMON/BROWSER

```
source $BUILD_HOME/cmsgemos/setup/paths.sh
source $BUILD_HOME/ldqm-browser/setup.sh
cd $BUILD_HOME/ldqm-browser/LightDQM
python2.7 manage.py runserver <MACHINE>:<PORT> >& logname.log &
```

> ### EXAMPLE FOR CODE RUN ON MACHINE GEM904DAQ01 AND PORT 5017
>
> ```
> python2.7 manage.py runserver gem904daq01.cern.ch:5017 >& test.log &
> ```

The DQM Daemon automatically begins running when the Light DQM Browser is configured in the code above. It will output some information to the log file which you can view to verify that it is running.

While data is being taken, you can view summary histograms in the Light DQM Browser. More information on the Light DQM Browser is in the next section.

2.4.2 MANUALLY

To manually run data through the light DQM, use the binary files in the following order:

> ### DATA FILE (.DAT) TO RAW ROOT FILE (.RAW.ROOT)
>
> ```
> $BUILD_HOME/gem-light-dqm/gemtreewriter/bin/<1>/<2>/unpacker run.dat sdram
> ```
>
> Tab-complete the following (should only have one option)
>
> <1> XDAQ_OS - Machine-dependent, determined by compiler
>
> <2> XDAQ_PLATFORM - Machine-dependent, determined by compiler

This will automatically create the .raw.root file in the same location. This file contains the GEM Tree.

> ### RAW ROOT FILE (.RAW.ROOT) TO ANALYZED ROOT FILE (.ANALYZED.ROOT)
>
> ```
> $BUILD_HOME/gem-light-dqm/dqm-root/bin/<1>/<2>/dqm run.raw.root
> ```

This will automatically create the .analyzed.root file in the same location. This file contains the Light DQM histograms.

> ### ANALYZED ROOT FILE (.ANALYZED.ROOT) TO PRINTED HISTOGRAMS
>
> ```
> $BUILD_HOME/gem-light-dqm/dqm-root/bin/<1>/<2>/gtprinter run.analyzed.root
> ```

This will automatically print all of the histograms in a directory structure that matches the hardware configuration.

3 LIGHT DQM BROWSER

3.1 RUNNING

The code to start the light DQM browser is in the ldqm-browser repository. Starting the light DQM browser will also start the DQM daemon, if you have already begun the Light DQM Daemon, do not run this code again:

RUNNING LIGHT DQM DAEMON/BROWSER

```
source $BUILD_HOME/cmsgemos/setup/paths.sh
source $BUILD_HOME/ldqm-browser/setup.sh
cd $BUILD_HOME/ldqm-browser/LightDQM
python2.7 manage.py runserver <MACHINE>:<PORT> >& logname.log &
```

EXAMPLE: CODE RUN ON MACHINE GEM904DAQ01 AND PORT 5017

```
python2.7 manage.py runserver gem904daq01.cern.ch:5017 >& test.log &
```

EXAMPLE: ADDRESS TO VIEW IN INTERNET BROWSER

```
gem904daq01.cern.ch:5017/main
```

We recommend using **Firefox** to view the Light DQM Browser. It is much better at handling the backend software structure we currently use.

3.2 VIEWING

Make sure that you add **/main** to the end of your server address to view the browser! Otherwise you will see a screen like the one shown below:
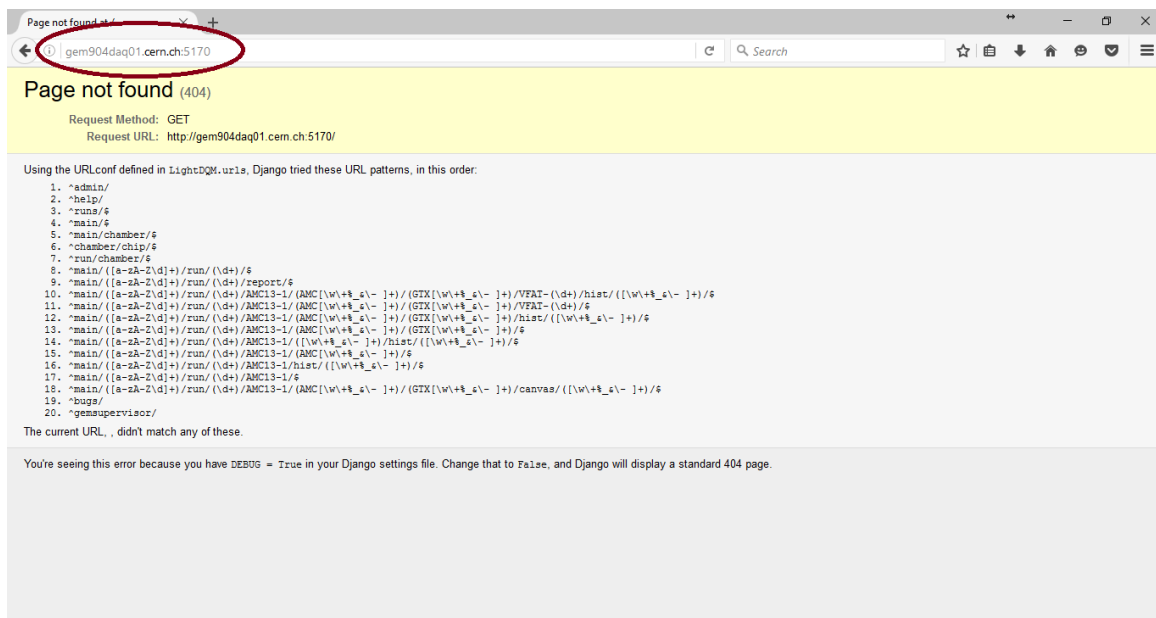

FIGURE 2: LIGHT DQM BROWSER - PAGE NOT FOUND (NO /MAIN)

3.3 MAIN

```
{server_address}/main
```

This will bring you to the main screen, which is blank. It looks like the figure below. To select a run, click on **Run List** in the upper left hand corner.
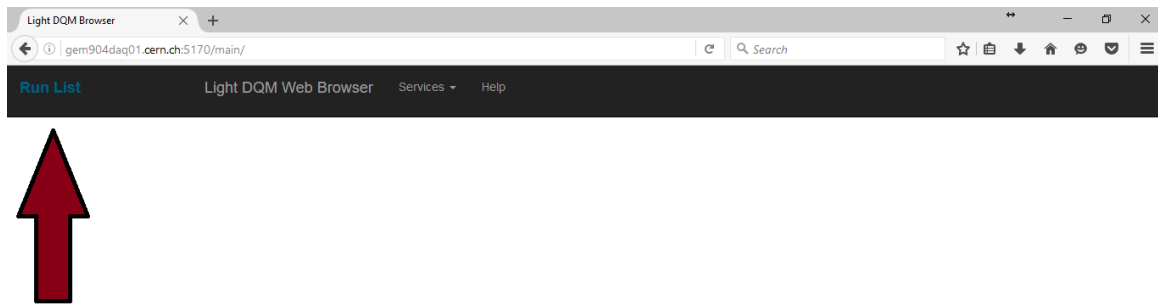

FIGURE 3: LIGHT DQM BROWSER - MAIN

3.4 RUNLIST

```
{server_address}/main -> "Run List"
```

Clicking on **Run List** will open a sidebar, as shown in the image below, which will display all runs currently in the database.

FIGURE 4: LIGHT DQM BROWSER - RUN LIST

3.5   RUN

Clicking on the Run you are interested in will bring you to the Run view, shown below. It will show you all the chambers associated with this run and their associated AMC, colored based on their status.
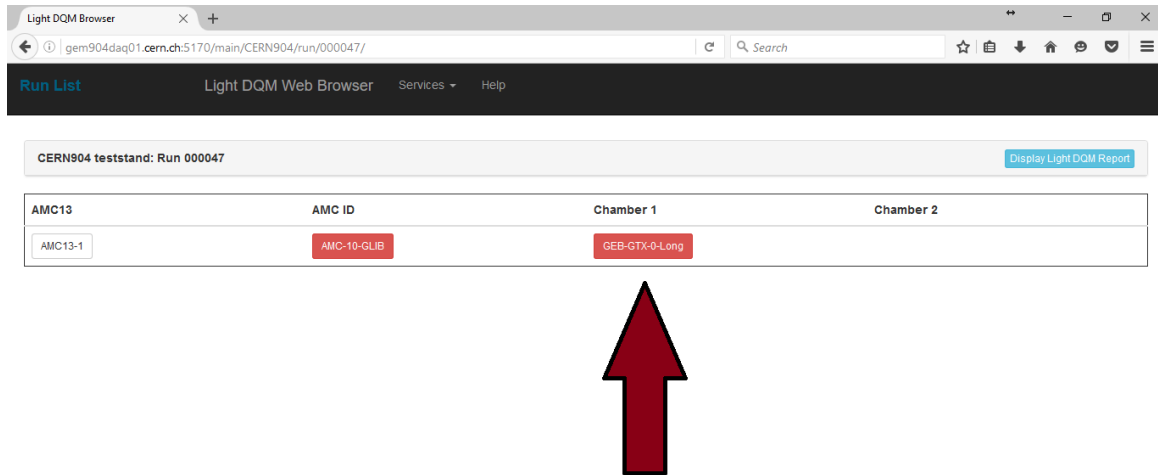

FIGURE 5: LIGHT DQM BROWSER - RUN VIEW

As you can see in this run, there is an error associated with the GTX-0 chamber connected to the AMC (GLIB) in slot 10 in the uTCA crate, indicated by its red-colored button.

3.6   HARDWARE

Clicking on any piece of hardware will bring you to histograms specific to that hardware from the run. These histograms are described in more detail in the next section.

3.7   SUMMARY CANVASES

Summary Canvases, the only fully online part of the DQM currently, are sorted per GEB, and can be found by clicking on any Chamber. The buttons to view them are dark blue.

4   HISTOGRAMS

4.1   AMC13

The following image shows the histograms specific to the AMC13. These include:

- Control_Bit5 - Should always fill bin 5

FIGURE 6: LIGHT DQM BROWSER - AMC13-SPECIFIC HISTOGRAMS

- Control_BitA - Should always fill bin 10

- Evt_ty

- LV1_id

- Bx_id

- Source_id

- CalTyp

- nAMC - number of AMCs for each event

- OrN

- CRC_amc13

- Blk_Not

- LV1_idT

- BX_idT

- EvtLength

- CRC_cdf

4.2   AMC

The following image shows the histograms specific to the AMC13. These include:

- L1A

- BX

- Dlength

- FV

  "AMCnum", "L1A", "BX", "Dlength", "FV", "Rtype", "Param1", "Param2", "Param3", "Onum", "BID", "GEMDAV", "Bstatus", "GDcount", "Tstate", "ChamT", "OOSG", "CRC", "L1AT", "DlengthT"];

- Control_Bit5 - Should always fill bin 5

- Control_BitA - Should always fill bin 10

- Evt_ty

- LV1_id

- Bx_id

- Source_id

- CalTyp

- nAMC - number of AMCs for each event

- OrN

- CRC_amc13

- Blk_Not

- LV1_idT

- BX_idT

- EvtLength

- CRC_cdf

FIGURE 7: LIGHT DQM BROWSER - AMC-SPECIFIC HISTOGRAMS

4.3 GEB/OPTOHYBRID

The following image shows the histograms specific to the AMC13. These include:

- Control_Bit5 - Should always fill bin 5

- Control_BitA - Should always fill bin 10

- Evt_ty

- LV1_id

- Bx_id

- Source_id

- CalTyp

- nAMC - number of AMCs for each event

- OrN

- CRC_amc13
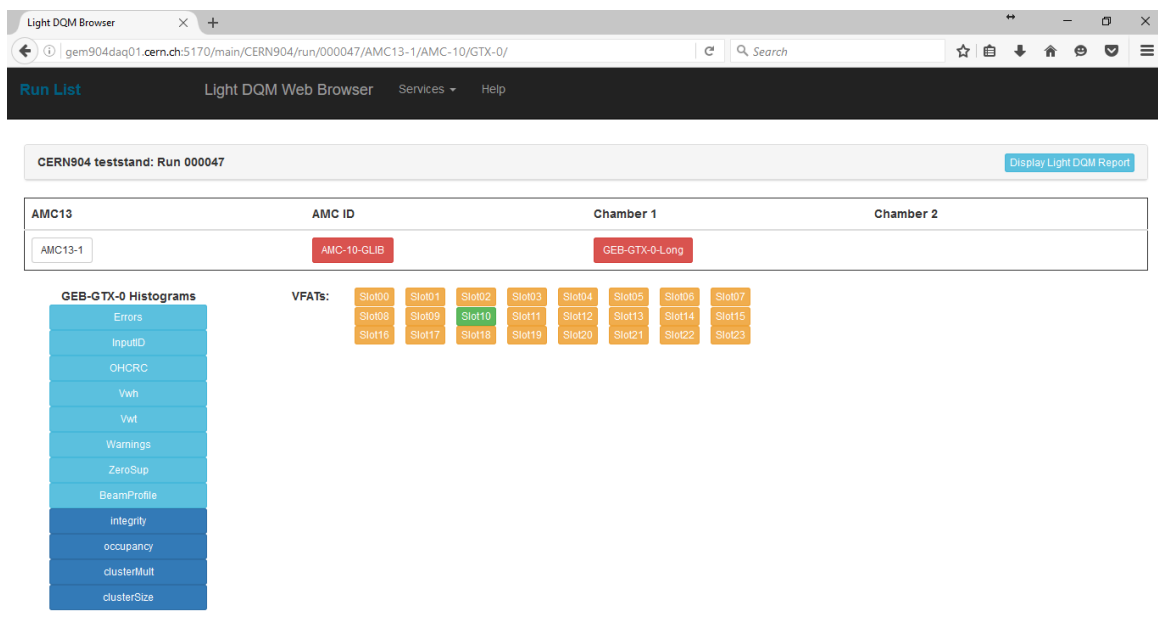
- Blk_Not

- LV1_idT

- BX_idT

- EvtLength

- CRC_cdf

FIGURE 8: LIGHT DQM BROWSER - GEB-SPECIFIC HISTOGRAMS

4.4   SUMMARY CANVASES

FIGURE 9: LIGHT DQM BROWSER - SUMMARY CANVASES (DARK BLUE BUTTONS)

4.5   VFAT

The following image shows the histograms specific to the AMC13. These include:

- Control_Bit5 - Should always fill bin 5

- Control_BitA - Should always fill bin 10

- Evt_ty

- LV1_id

- Bx_id

- Source_id

- CalTyp

- nAMC - number of AMCs for each event

- OrN

- CRC_amc13

- Blk_Not

- LV1_idT

- BX_idT

- EvtLength

- CRC_cdf

FIGURE 10: LIGHT DQM BROWSER - VFAT-SPECIFIC HISTOGRAMS

4.6 SUMMARY CANVASES

5 TROUBLESHOOTING

Is your $BUILD_HOME correct?

Did you source $BUILD_HOME/cmsgemos/setup/paths.sh?

Did you source $BUILD_HOME/ldqm-browser/setup.sh?

5.1 CONTACT INFORMATION

Issues? Contact:

Robert King - robert.david.king@cern.ch

Mykhailo Dalchenko - mykhailo.dalchenko@cern.ch