

# Table of Contents

|   |          |
|---|----------|
| <b>LHCb RICH CAMERA.....</b>  | <b>1</b> |
| CAMERA -Commissioning and Monitoring Error Reporting Application..... | 1        |
| Where to get it.....  | 1        |
| How it works.....   | 1        |
| Camera server.....  | 1        |
| Sending messages.....   | 1        |
| Server.....   | 1        |
| Client.....   | 1        |
| Redundancy/scalability.....   | 2        |
| Connecting from outside CERN.....                                     | 2        |
| Looking at logged messages.....                                       | 2        |
| Using camera in the control room.....                                 | 3        |
| Algorithms reporting to Camera.....                                   | 3        |
| Camera Gui hints.....   | 3        |
| Presentations and further information.....                            | 3        |

# LHCb RICH CAMERA

Warning : This page is a little out of date ... Its needs updating to reflect the latest changes.

## CAMERA -Commissioning and Monitoring Error Reporting Application

Camera is a network application specifically built for communication and error logging for the commissioning and monitoring. It is based on servers, that handle incoming errors and redistribute them to any number of clients. Error messages contain a header line with the sender name and a severity level and a summary of the error. Additionally, there can be appended data, consisting of a combination of texts timestamps graphs or histograms. The purpose of this is to not only raise an error, but append the exact information, that error was derived from. This differs from DIM, in that the histogram is static and will be saved alongside the error message. (The appendix can contain DIM addresses to inform the shifter of the current status as well.) The whole system is optimised for low latencies.

### Where to get it

Camera is currently part of the `Rich/RichMonitoringTools` package. All the executables are built in the usual CMT way, by running `make` from the `cmt` directory.

### How it works

#### Camera server

The camera server is currently located on `r2ecs01.lbdag.cern.ch` and referred to as `host` in the further description.

#### Sending messages

There is a Gaudi-tool in `Rich/RichMonitoringTools`, that handles the sending of error messages. It can be used inside any Gaudi algorithm or tool. In addition, there are the tools `camwarn` and `camsend` (in `RichMonitoringTools/v*/src/camera`), that send single messages from the command line: e.g. `./camwarn host "myname/0/A simple message"` or `camwarn host "myname/3/A severe error"` This way the system can be used for communication in and with the control room as well. E.g. to state your opinion on a specific error.

#### Server

The job of the server is to strip the additional information from the message and only pass on the header line. this header-line is also stored in a warnings log. The additional information is saved in a different file, and an URL is added to the header line, that indicates where to get the additional information from. A simple tiny webserver comes with camera, whose only job is to serve these extra files. The reason for the stripping is to speed up redistribution especially in cases, where the additional information is very large. The back-end can at some point be a database instead of the filesystem.

#### Client

A simple `telnet` connection and `wget` would be enough to access the information. e.g `telnet host 12346` would connect to the error stream, and `wget host:8888/data/example.dat` would download the extra info. For several reasons, there is a GUI called `camera`. A script, that starts camera is in `/group/rich/scripts/runCamera.sh`. It sets the GUI up to use the cache area (e.g. `$HOME/camcache`). You

might want to delete those files from time to time.

## Redundancy/scalability

For redundancy several servers can be used, so that the next in line is used, when the first one is busy or dead. Also different jobs/subdetectors can report to different servers. To organise data flow and spread the load. The GUI is able to connect to more than one server at the time.

## Connecting from outside CERN

The CERN and LBDAQ networks are setup in a way, that does not allow simple access. You will need two tunnels to be able to connect to the camera server:

on your machine use: `ssh -L 12346:127.0.0.1:12346 lxplus.cern.ch`

then on lxplus: `ssh -L 12346:host:12346 lbgw`

for retrieval of extra info:

on your machine use: `ssh -L 8888:127.0.0.1:8888 lxplus.cern.ch`

then on lxplus: `ssh -L 8888:host:8888 lbgw`

You will have to set some environment variables to configure camera to be aware of this. the following script would do this for you. The commented sections will retrieve a warnings file given as the first argument, and open it before connecting to the server.

```
#!/bin/bash
export CAMPROXY=127.0.0.1
export CAMCACHE=$HOME/camcache
mkdir $CAMCACHE
mkdir $CAMCACHE/data
#rm $1
#wget $CAMPROXY:8888/data/$1
./camera $CAMPROXY # $1
```

## Looking at logged messages

The Camera GUI can read messages from the network and from files. The full options are `camera`. If no arguments are given it is called as `camera 127.0.0.1`. If you want to read in a log file named `warnings.out`, just type `camera NULL warnings.out`. If you want to read the file and the connect to some server type `camera host warnings.out`.

**Rich specific:** To get the most recent log file, you will need to know its name. I usually call them `warnings_ddmmyy.out`, where `ddmmyy` are day month and year. if you have set up the tunnels as described above, you can use `wget 127.0.0.1:8888/data/warnings_020807.out` to get the file from August 2nd 2007. Then open it with `camera NULL warnings_020807.out`. The extra info files will only be available as long as they are not archived yet.

Due to the large number of files in the data directory, a method to archive them has to be found. This is being worked on. The aim is to keep them online for later inspection, but keep the data directory of the camera server small and tidy. expect updates soon.

## Using camera in the control room

There are some scripts in in `/group/rich/scripts` for convenience in the pit:

- `runCamera.sh` runs the GUI set up for the pit. make sure, that you have X forwarding enabled to where ever you are.
- `comment` is used to send a (white) comment to the server. e.g. to state your opinion on a specific error or run conditions. example: `./comment "New run started with SiBias on"` . it will send your username as sender.
- `newfile` closes the old warnings logfile and opens a new one. Use with care once a day. It uses `warnings_ddmmyy.out` as the new filename, where `ddmmyy` is the current date.

## Algorithms reporting to Camera

Most of the reporting can be switched off in the options file. All of the algorithms will report their initialisation, so that the shifter knows, that and when they where started. This list is for what could be reported more than what will be, because it obviously depends on the configuration:

|                    |   |
|--------------------|---|
| RichDAQMon         | reports mismatches of Event IDs and ODIN times  |
| RichHitMapsMonitor | reports number of events seen in the last N seconds   |
| HotPixelReport     | reports hot or dead pixels found, or just a summary when switched off. It generates options files for the <code>SoftPixelMask</code> as well. Turn reporting off, when there is a large testpattern, because it will flood the online display. In that case it will only report how many have been found, after N events.   |
| RichOnlineStatus   | reports changes in SiBias Voltage (will add LHCb.HV and LHCb.LV)  |
| RichHPDCount       | reports too many hits in one HPD when SiBias is "On" on an event by event basis. When <code>SiBias</code> is low it can report when there was no hit at all for too many events in an HPD. To use this safely, a good hot pixel mask has to be used. This can be generated by running the <code>HotPixelReport</code> before the <code>SoftPixelMask</code> for a long enough time (thousands of events). |

## Camera Gui hints

- If things go crazy, just un-check the "=Update=" button. The right panel will stop updating, the left one will go on counting the numbers of messages. Don't forget to turn it back on again. Camera will still receive messages and buffer them, but the update of the right panel is stopped.
- The standard number of maximum lines is set to 50 to allow speedy updates of the right panel. Choose higher numbers to see the full run.
- If there is an "->" arrow left of the text, then there is more. Click on the line to see the extra info. Often it is just a time-stamp.
- Click in the left panel to see only the messages from that sender.
- If you hit the `clear all` button, the messages from "all" senders are deleted, but they will still be visible, if you click on the sender names in the left panel.
- If you are bored with the "good" green messages un-check the `info` button, correspondingly for the `warnings` and `errors` buttons.

## Presentations and further information

- Monitoring with Camera [↗](#) Histogram and Monitoring, 14. Nov. 2007
- CAMERA [↗](#) HPD and Rich monitoring, 24. Oct. 2007
- Rich Monitoring Experience [↗](#) Online, 11. Sep. 2007

-- UlrichKerzel - 07 Dec 2007

This topic: LHCb/RichOperations > RichCamera

Topic revision: r4 - 2008-06-12 - ChristopherRJones



Copyright &© 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)