ATLAS Level-1 Calorimeter Trigger
Software Note 019

18 May 2006
Version 1.0

Monitoring the Cable Tests Runs

Victor Andrei

1 Introduction

This document is intended to describe the current application of the ***ppmMonitoring*** package, which has been developed to provide *monitoring histograms* of parameters specific to each run type. The monitoring package considers all three cable test multi-step runs (PPM DAC, PPM FIFO and PPM PHOS4), and it is already included in the cable test scripts of the L1Calo Online Software. Applications for other monitoring tasks (e.g. rate metering, event data monitoring, slow control, etc) are under development, and reports about their status will be presented in the future.

The present document provides a detailed description of the *monitoring histograms* as well as of the underlying algorithms. Examples of monitoring results for each run type are given in each section. In order to illustrate the functionality of the monitoring package, additional examples are included in the appendix of the document. It should be emphasised that all the *monitoring histograms* presented in this report have been obtained using real PPM event data, taken during the *tile cable tests* at CERN. The histograms are displayed with the PMPpresenter tool, and a detailed documentation about how to run the monitoring package, and display the results with PMPpresenter is provided in the Section 5. Please note that, since the current version of the Online Histogramming server (OH) does not transport the y-axis information of the one-dimensional histograms, the titles on the respective axes were added *off-line*, in order to provide a clear information about the contents of the *monitoring histograms*.

Also, it should be emphasised that the document does not aim to analyse the observed misbehaviours of the system. Some of the examples given here depict such cases, but the goal is to show how the information provided by the monitoring package can be used to detect these misbehaviours, and later perform a diagnostic of the run.

1.1 The ppmMonitoring Package

The ***ppmMonitoring*** package creates and publishes a collection of *monitoring histograms*, based on the information provided by the cable test runs. After each step of a run, the digitized PPM event data is read out by cable test applications, via VME, and published on the **IS** (Information Service) server. When the multi-step run is completed, a dedicated tool retrieves all related data from the IS server, and stores them in appropriate files (for detailed information, please refer to the documents listed in Section 6).

The monitoring package uses these files as inputs to provide histograms of certain quantities specific to each run type. First of all, a decoding application (**ppmDecoder**) is employed to provide the data in a suitable format. Then, characteristic information to each run is extracted, histogrammed and displayed with a dedicated graphic interface (**PMPpresenter**): readout statistics of the PPM channels, pedestal estimation per channel, signal amplitudes, rise time or FWHM. The future developments of the package may use the data available on the IS server, in order to ensure a monitoring of the cable tests at each step in the run. Also, the current version of the monitoring package considers the case of only one PPM, but the next releases should handle the full-crate system.

The following sections attempt to describe the algorithms implemented in the monitoring package, and present examples of monitoring results, as obtained for each run type.

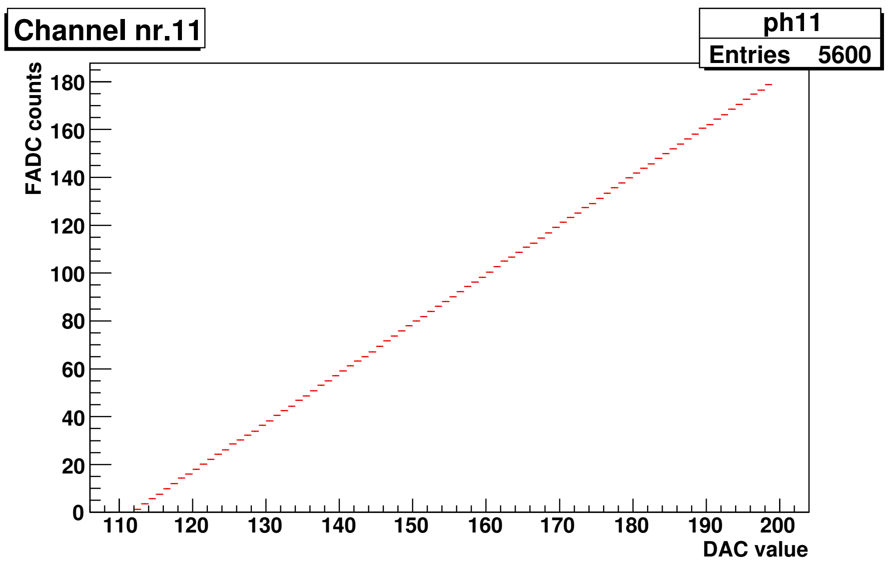

Figure 1: Example of a profile histogram describing the mean of the FADC counts as a function of the DAC value. The bin content, corresponding to the first DAC settings used in the run (from 104 to 111), is zero.

2 The PPM DAC Scan Multi-step Run

The DAC scan run is performed in order to determine the DAC count, required to set the same pedestal value for each PPM channel. This is necessary in order to ensure that each trigger tower signal is digitized with the same zero offset. During the run, the DAC setting is progressively increased in steps of 1 LSB (for an 8-bit DAC range), while reading and storing in an appropriate file (**l1calo-ppX-DAC.dat**) the scan parameter value (DAC setting) and the corresponding *raw data* (typically a set of 20 FADC samplings) from each PPM channel. When the run is completed, a dedicated *cable tests application* checks the linearity of the system (DAC value vs. FADC counts) and determines the DAC setting corresponding to the desired pedestal value (typically 40 FADC counts).

The ***ppmMonitoring*** package uses the file mentioned above as input, to provide histograms of quantities specific to this run type.

2.1 Monitoring the DAC Scan Run

Since the pedestal value and the noise contribution are likely to vary from channel to channel, the linearity of the FADC counts as a function of the DAC value has to be checked for each PPM channel separately. In order to accomplish this, the algorithm (***PpmMonitDacScan*** class) defines a profile histogram for each PPM channel, to compute the mean value of the FADC counts and its RMS, for each DAC setting (see Fig. 1).

For small DAC settings the FADC is in underflow, and thus the digitization returns a zero count. The content of the bins (i.e. mean value of the FADC counts) corresponding to these DAC values, is set to zero, and their range is excluded when applying the linear fit. In order to account for those situations where only the largest noise amplitudes are ”seen” by the digitization, resulting in small fluctuations of the bin content in the range mentioned above, a certain minimum value is chosen for the linear fit: the first DAC value whose corresponding bin content is larger than 5 FADC counts.

For each PPM channel, the ***slope*** parameter is extracted from the linear fit and filled into appropriate *monitoring histograms* (histSlope1, histSlope2). A parameter named ***offset***, and defined as:

$$offset = -\frac{y\text{-}intercept}{slope} \quad [DAC\ value] \ , \tag{1}$$

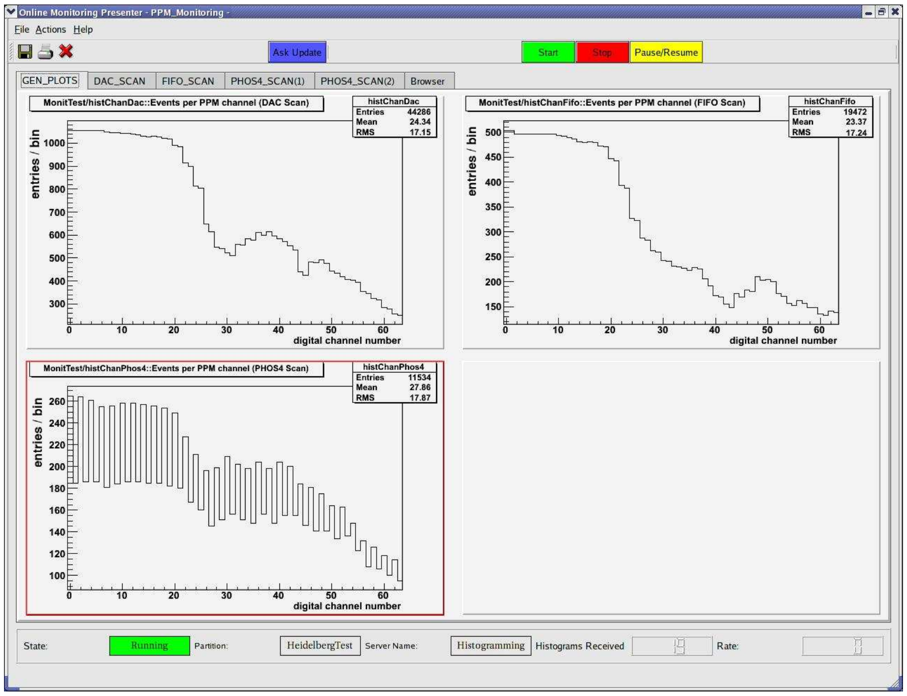

Figure 2: Readout statistics of the PPM channels during the DAC, FIFO and PHOS4 scan multi-step runs.

is computed and filled into separate *monitoring histograms* (histOffset1, histOffset2), where ***y-intercept*** is the other parameter of the linear fit. Mathematically, the *offset* is the *x-intercept*, and in the context of the present analysis it defines the DAC value (as a real number) whose corresponding FADC count is zero. If the linear fit returns a zero value for the *slope* parameter, then the *offset* is initialized with zero in order to avoid errors while filling the *monitoring histograms*, and flag the situation in the same histograms.

2.2 Examples

When the analysis of all 64 channel is completed, the *monitoring histograms* are published on the **OH** (Online Histogramming) server, and then displayed with the **PMPpresenter** tool. Figure 3 shows the results obtained by the monitoring package for a typical DAC scan multi-step run. The *upper* plots describe the *slope* and *offset* values as determined for each PPM channel, while the *lower* plots describe the same values, this time as projected on the *y-axis*. All these histograms are displayed in the same panel (DAC_SCAN) of the PMPpresenter. In addition, the monitoring package (***PpmMonitGenPlots*** class) provides a readout statistics of the PPM channels for the same DAC scan multi-step run (see the *upper left* histogram in Fig. 2), displayed in the GEN_PLOTS panel of PMPpresenter. The histogram shows the number of times a PPM channel was read out during the respective run.

Both panels of histograms, offer to a *shift crew* the possibility to visualise, shortly after the DAC scan multi-step run is completed, histograms of parameters specific to this run type, and to detect any misbehaviour during the respective run. In order to illustrate the kind of misbehaviour that may occur, the overview of another DAC scan run is given in the appendix (see Figures 7 and 8). The *slope* parameter was determined to be zero for the PPM channels 8, 9, 11 and 13, and therefore the *offset* corresponding to these channels was set as well to zero. The situation can be easily explained for channels 9 and 13, by looking at the readout statistics of the respective run: no data was read out from these channels (see Fig. 7). For the other two channels the insufficient readout may be a possible explanation for a zero *slope* value. Also, other two kinds of misbehaviour can be observed for the PPM channels 52 and 63. For the former, a negative and large *slope* parameter was determined,

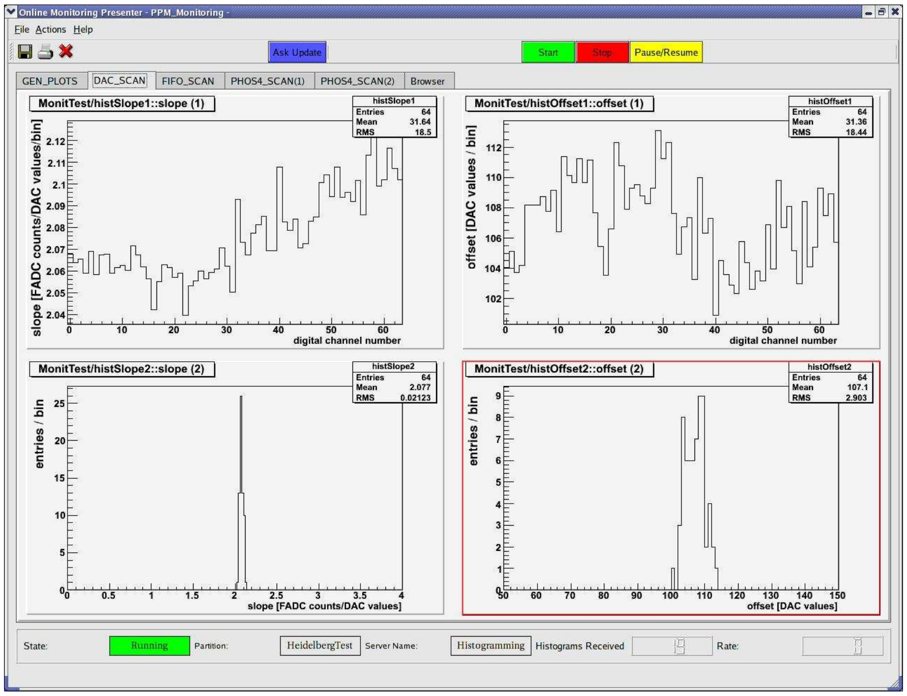

Figure 3: Results of a PPM DAC scan multi-step run. The ***upper*** plots depict the slope and offset values as obtained for each PPM channel, while the ***lower*** plots describe the same values as projected on the y-axis.

which is the consequence of an improper linear fit, due to the insufficient readout for the respective channel. In the other case, the *offset* was determined to be around -110 (*DAC values*), and this is a consequence of the very small (close to zero) *slope* value.

The fall of the distribution with increasing PPM channel number, shown by the *upper left* histogram in Figure 2 is another example of misbehaviour. The *upper* plots displayed in Figure 9 in the appendix, show that, excepting a few channels, the readout of the PPM was uniform during the respective runs.

3 The PPM FIFO Scan Multi-step Run

The FIFO scan multi-step run is performed in order to estimate the pedestal value in each PPM channel, with the DAC setting determined by the DAC scan. During the run, the scan parameter value (the PHOS4 delay, always set to 12 for this run type) and the *raw data* (FADC counts), corresponding to each PPM channel, are read out and saved in an appropriate file **(l1calo-ppX-FIFO.dat)**. This file is used as input by the monitoring package to provide histograms of quantities specific to the FIFO scan multi-step run.

3.1 Monitoring the FIFO Scan Run

The algorithm carried out by the monitoring package (***PpmMonitFifoScan*** class) defines 64 *one-dimensional* histograms to estimate the pedestal value in each PPM channel. During the readout of the file, each histogram is filled with the corresponding FADC counts. When the filling is completed, the mean values and their widths are extracted and filled into appropriate *monitoring histograms* (histFifoPed1, histFifoPed2, histFifoRMSPed1, histFifoRMSPed2), which are then published on the OH server and displayed with PMPpresenter.

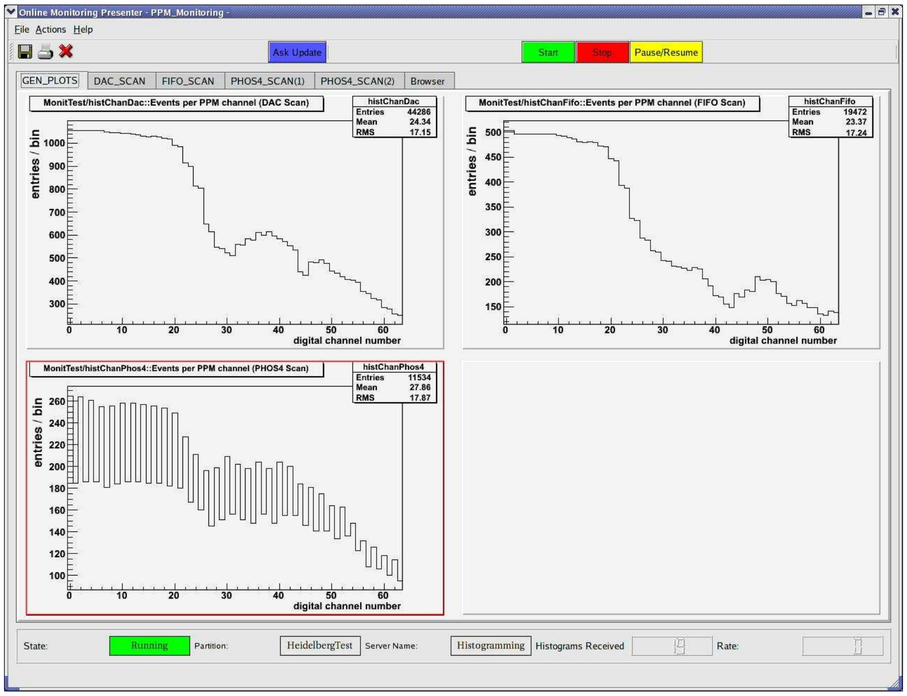

Figure 4: Results of a PPM FIFO scan multi-step run. The ***upper*** histograms depict the mean values of the pedestal and their widths, as determined in each PPM channel. The same values are plotted in the ***lower*** histograms, as projected on the y-axis.

3.2 Examples

Figure 4 shows the results obtained by the monitoring package for a typical FIFO scan multi-step run. The histograms, displayed in the FIFO_SCAN panel of PMPpresenter, offer the possibility to check the proper functioning of the system, in a short time after the run is completed. The *upper* histograms describe the mean pedestal values and their widths as determined for each PPM channel, while the *lower* histograms depict the same values as projected on the y-axis. As illustrated in this example, the distribution of the mean pedestal values (*lower left* plot) is centered close to 40 FADC counts, which is the expected pedestal value (as set after the DAC scan run). The RMS of these values is 0.726 FADC counts, which roughly corresponds to a less than 2% deviation from the same expected value. An example of misbehaviour is illustrated in the *right* plots. The width of the pedestal is in general smaller than 1 FADC count, except for some channels at the top and the bottom of the PPM, that show a noise contribution larger than 1.2 FADC counts.

Additionally, as in the case of the DAC scan run, a readout statistics of the PPM channels for the same FIFO scan multi-step run, is provided as well in the GEN_PLOTS panel of PMPpresenter (Fig. 2, *left* histogram). The examples given in the appendix show a much better readout of the PPM channels (Fig. 9).

4 The PPM PHOS4 Scan Multi-Step Run

The PHOS4 scan multi-step run is performed in order determine the optimal PHOS4 delay for each PPM channel. The run scans through all PHOS4 delays (from 0 to 24 ns, in steps of 1 ns) to find the precise pulse timing of each PPM channel. The results of the complete run, meaning the PHOS4 settings and the FADC counts are saved in appropriate files (**l1calo-pp0-PHOS4_A.dat**, **l1calo-pp0-PHOS4_B.dat**). These files are used as inputs by the monitoring package to provide an overview of the run: pedestal estimation in each PPM channel, signal amplitudes and time characteristics (rise time and FWHM).

4.1 Monitoring the PHOS4 Scan Run

The monitoring algorithm (***PpmMonitPhos4Scan*** class) defines a set of 64 *one-dimensional* histograms to estimate the pedestal in each PPM channel, and a set of 64 profile histograms (FADC counts vs time, 1 ns width for the time bin) to reconstruct the pulse shapes and determine their characteristics.

For pedestal determination, only the first 500 leading samples (equivalent to 20 *bunch-crossings*) are taking into consideration. Each histogram is filled with the corresponding FADC counts, and then the mean values are extracted and filled into appropriate *monitoring histograms* (histPed1, histPed2). The standard deviation of each resulted distribution is extracted as well, but only used later in the algorithm.

The profile histograms are filled with the FADC counts as determined for each PHOS4 setting. Once the pulse shape is reconstructed, the amplitude and the time characteristics of the signal can be extracted. First of all, in each histogram a maximum amplitude is determined and filled into a corresponding *monitoring histogram* (histAmpPed). Remember that this value is not the largest FADC count, but the mean of all FADC counts in the respective bin. Since not all the channels may be supplied with a signal, it is necessary to determine whether the maximum observed amplitude is a real signal amplitude or the maximum noise contribution in the respective channel. Therefore, the following condition is checked:

$$A_{max} > \mu_{ped} + 10 \cdot \sigma_{ped} \quad , \tag{2}$$

where μ_{ped} and σ_{ped} are the mean pedestal value and its standard deviation as determined for a given PPM channel, while A_{max} is the maximum observed amplitude in the same channel. The factor 10 was chosen to ensure that the observed amplitude is well above the noise contribution. The probability to determine an outlier is quite reduced, since the observed amplitude value represents the mean of all FADC counts in the respective bin.

If the above condition is fulfilled, then the time characteristics of the signal are extracted. First of all, the determined pedestal is subtracted from A_{max} in order to obtain an estimate of the signal amplitude:

$$A_{signal} = A_{max} - \mu_{ped} \quad , \tag{3}$$

and the result is filled into a *monitoring histogram* (histAmp). Then, a linear fit is applied on the rising edge of the pulse in order to determine the *rise time* of the signal. The algorithm first searches for the nearest neighbours (in terms of FADC counts) to 10% and 90% of A_{signal}, and then fits a straight line in the range defined by these neighbours. Using the fit parameters, the time difference between the points on the line whose values on the FADC axis are 10% and 90% of A_{signal} is returned, and then filled into appropriate *monitoring histograms* (histRiseTime1, histRiseTime2).

Another time characteristic determined by the algorithm is the *full width at half-maximum* (FWHM) of the signal. From the fit on the rising edge, the time value corresponding to 50% of A_{signal} is extracted. By applying a similar fit on the falling edge, the corresponding time value to 50% of the same amplitude is extracted as well. The FWHM is then given by the difference of the two time values, and filled into appropriate *monitoring histograms* (histFwhm1, histFwhm2).

4.2 Examples

Figures 5 and 6 show the results achieved by the monitoring package for a typical PHOS4 scan multi-step run. The histograms are displayed in the PHOS_SCAN(1) and PHOS4_SCAN(2) panels of PMPpresenter. This run was taken with one cable (C50) connected to the PPM input 1, via TPCC.

The *left* histograms shown in the first panel (Fig. 5) describe the mean pedestal values μ_{ped} as determined for each PPM channel (*up*), and again the same values as projected on y-axis (*down*). The other two histograms depict the maximum observed amplitudes A_{max} (*up*) and the real signal amplitudes A_{signal} (*down*), as determined for each PPM channel. The histograms show that, during this test, only nine PPM channels were supplied with a signal. It can be seen that for channel 5 a larger amplitude, with respect to the other channels, was determined. The mean pedestal value, as determined for this channel, is within one standard deviation (see the pedestal distribution, *lower left* plot), and therefore its subtraction doesn't change the proportionality between the amplitude of this signal and the others. The rise time and FWHM information, displayed in the second panel (see Fig. 6), induce the picture of a steeper and narrower signal in channel 5, and perhaps its shape was determined by external factors (charge injection system, cable delay and attenuation, etc).

In the case of no signal at the inputs of the PPM, the monitoring returns empty histograms for the signal amplitude, rising time and FWHM parameters. An example of such a case is illustrated in the appendix, in Figures 12 and 13.

As for the other two run types, the monitoring provides a readout statistics for the respective PHOS4 scan run, too, displayed in the same GEN_PLOTS panel of PMPpresenter (Fig. 2, *lower* histogram).

5 Running the monitoring package

As mentioned all over this document, the files generated at the end of the cable test multi-step runs are used as inputs by the monitoring package. Therefore the monitoring process should only be started when the multi-step runs are completed. In order to activate the monitoring, one has to run ***ppmhistogramming***, a client application provided in the bin subdirectory of the ppmMonitoring package, with the following arguments:

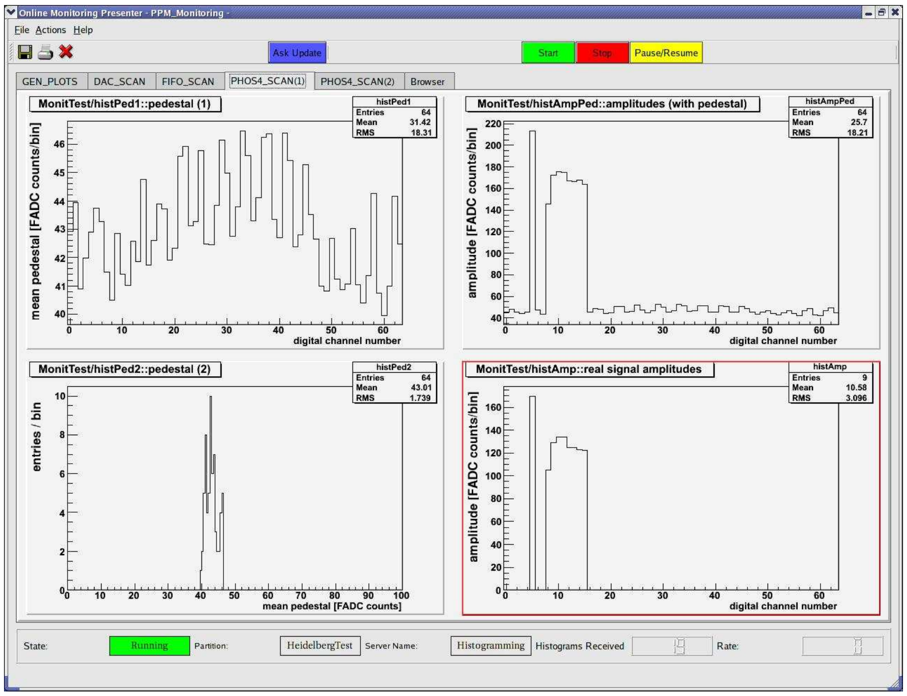

Figure 5: Results of a PHOS4 scan multi-step run (first panel). The ***upper left*** histogram describes the mean pedestal values as determined for each PPM channel, while the ***lower left*** histogram depicts the same values as projected on the y-axis. The ***right*** histograms describe the maximum observed amplitudes A_{max} (*up*) and the real signal amplitudes A_{signal} (*down*) as determined for each PPM channel.

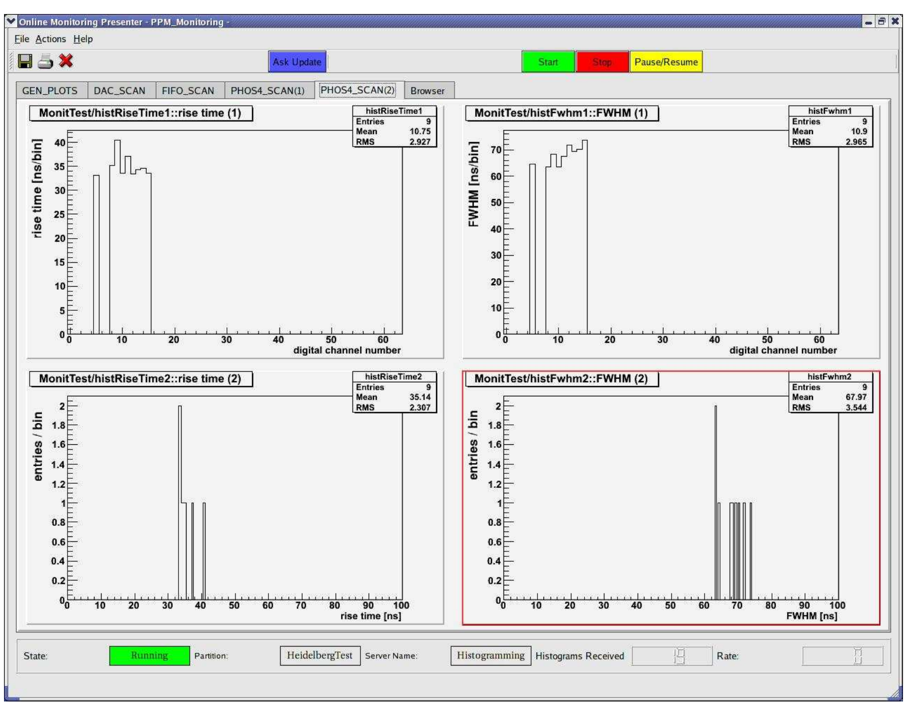

Figure 6: Results of a PHOS4 scan multi-step run (second panel). The ***upper*** histograms describe the rise time and the FWHM values as determined for each PPM channel. The ***lower*** histograms describe the same values as projected on the y-axis.

- **partition (-p)**: a valid IPCPartition. If not given, then the value of $TDAQ_PARTITION variable is retrieved from the system;
- **server (-s)**: name of a valid OH server. Default setting is "Histogramming";
- **provider (-n)**: name of the provider (to be set by the user). Default setting is "MonitTest";
- **path (-P)**: the path where the input VME buffer is stored;
- **file1 (-f)**: the name of the input VME buffer;
- **file2 (-o)**: the name of the second input VME buffer (required only when monitoring PHOS4 scans);
- **scan type (-S)**: an integer between 0 and 10, corresponding to the enumeration list defined in ScanParameter.h (***infraL1Calo*** package). To monitor the cable test runs, one should set this argument as following:
  - ⋆ **10** ⟶ PprDAC scan;
  - ⋆ **7** ⟶ PprFIFO scan;
  - ⋆ **8** or **9** ⟶ PprPHOS4 scan.

  If not given, then the scan type is initialized with 0 (corresponding to "None", in the same list). In the next versions of ppmMonitoring this argument may be removed, since its value can be anyway retrieved from the VME buffer.
- **dump (-d)**: an optional boolean argument, to regulate the dumping of those histograms which are employed by the monitoring package to determine the main characteristics of the run. If given, then the histograms are dumped into ROOT files as following:
  - ⋆ **DAC scan**:
    - → **ppmMonit_dac_profile.root**: 64 profile histograms storing the input signals;
    - → **ppmMonit_dac_ramp.root**: 64 profile histograms storing the FADC vs. DAC ramps;
    - → **ppmMonit_dac_data.root**: file to store the raw data, as structured by the PpmDecChannelData class (see ppmDecoder);
  - ⋆ **FIFO scan**:
    - → **ppmMonit_fifo_profile.root**: 64 profile histograms storing the input signals;
    - → **ppmMonit_fifo_pedestal.root**: 64 one-dimensional histograms storing the pedestal distributions;
    - → **ppmMonit_fifo_data.root**: file to store the raw data, as structured by the PpmDecChannelData class (see ppmDecoder);
  - ⋆ **PHOS4 scan**:
    - → **ppmMonit_phos4_profile.root**: 64 profile histograms storing the input signals;
    - → **ppmMonit_phos4_pedestal.root**: 64 one-dimensional histograms storing the pedestal distributions;
    - → **ppmMonit_phos4_data.root**: file to store the raw data, as structured by the PpmDecChannelData class (see ppmDecoder);

Examples (assuming that the script is using the default server and provider settings):
a) DAC scan:
ppmhistogramming **-P** ∼/inst/tile002/dacscan/run000/ **-f** l1calo-pp0-DAC.dat **-S** 10
ppmhistogramming **-P** ∼/inst/tile002/dacscan/run000/ **-f** l1calo-pp0-DAC.dat **-S** 10 -d
(in the latter case the histograms are dumped into ROOT files)

b) PHOS4 scan:
ppmhistogramming **-P** ∼/inst/tile002/cabling13C/test12/run003/ **-f** l1calo-pp0-PHOS4_A.dat **-o** l1calo-pp0-PHOS4_B.dat **-f** 9

The ***ppmhistogramming*** application sends the path, file and scan type information to a monitoring manager (***PpmMonitROBuffer***), which reads all the buffers stored in the input file(s) and, based on the input scan type argument, dispatches the decoded data (see ***ppmDecoder***) to an appropriate monitoring member. When the filling is completed, the manager collects and delievers the *monitoring histograms* to ***ppmhistogramming***, where ***OHRootProvider*** is employed to export them on the OH server.

5.1 Using PMPpresenter

In order to display the *monitoring histograms* is with ***PMPpresenter***, one needs to create a configuration file to instruct the tool how to display the histograms. A default configuration file for the current application of the ***ppmMonitoring*** package is provided in the *data* subdirectory (***ppm.conf***). Please note that the partition, server and provider names, given as input arguments, **MUST** be the same as those defined in the configuration file. Therefore, when using other arguments than the default ones (see again the previous section), make sure to apply the appropriate changes to the ***ppm.conf*** file.

To use ***PMPpresenter***:

- copy ***ppm.conf*** in your working (sub)directory;
- execute the command: PMPpresenter **-n** PPM_Monitoring **-c** ppm.conf **-l** log &
- click the **Start** button in the ***PMPpresenter*** panel;
- run ***ppmhistogramming*** application as described in the previous section.

6 Related Documents

A list of some related documentation:

- **L1Calo PPM Decoder Documentation** (doxygen web pages)
  http://hepwww.ph.qmul.ac.uk/l1calo/dox/ppmDecoder/html/
- **L1Calo PPM Monitoring Documentation** (doxygen web pages)
  http://hepwww.ph.qmul.ac.uk/l1calo/dox/ppmMonitoring/html/
- **"Monitoring in the Experiment"** (V. Andrei, talk at L1Calo Trigger Joint Meeting, Heidelberg, 16th March 2006)
  http://agenda.cern.ch/askArchive.php?base=agenda&categ=a061030&id=a061030s7t13/transparencies
- **L1Calo Analogue Cable Testing** (wiki page)
  https://uimon.cern.ch/twiki/bin/view/Atlas/L1CaloAnalogueCableTesting#Accounts_at_point_1

- **"CERN PPM Status: Test and Software"** (F. Foehlisch, talk at L1Calo Trigger Joint Meeting, Heidelberg, 15th March 2006)
  http://agenda.cern.ch/askArchive.php?base=agenda&categ=a061030&id=a061030s1t23%2Ftransparencies%2FcernStatus.pdf
- **"PPM Timing-Calibration"** (F. Foehlisch, July 2005)
  http://?!?
- **"Monitoring Requirements"** (E. Eisenhandler and M. Landon, L1Calo Software Notes, version draft 0.4, May 2004)
  http://hepwww.ph.qmul.ac.uk/l1calo/sweb/documents/doclist.html
- **"Tehnical Check-out, Calibration and Monitoring during Data-Acquisition in the ATLAS Level-1 Calorimeter Trigger"** (P. Hanke, revived draft, July 2003)
  http://www.kip.uni-heidelberg.de/atlas/L1/PP_System/L1_Setup_ROut_Jul03.pdf

A Appendix

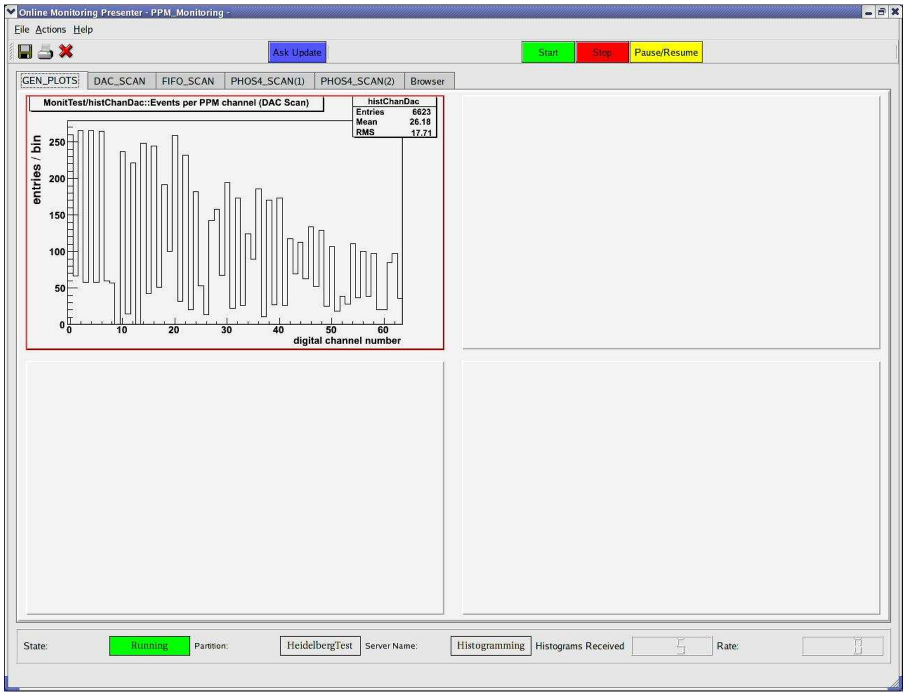

Figure 7: Readout statistics of the PPM channels for a problematic DAC scan multi-step run.

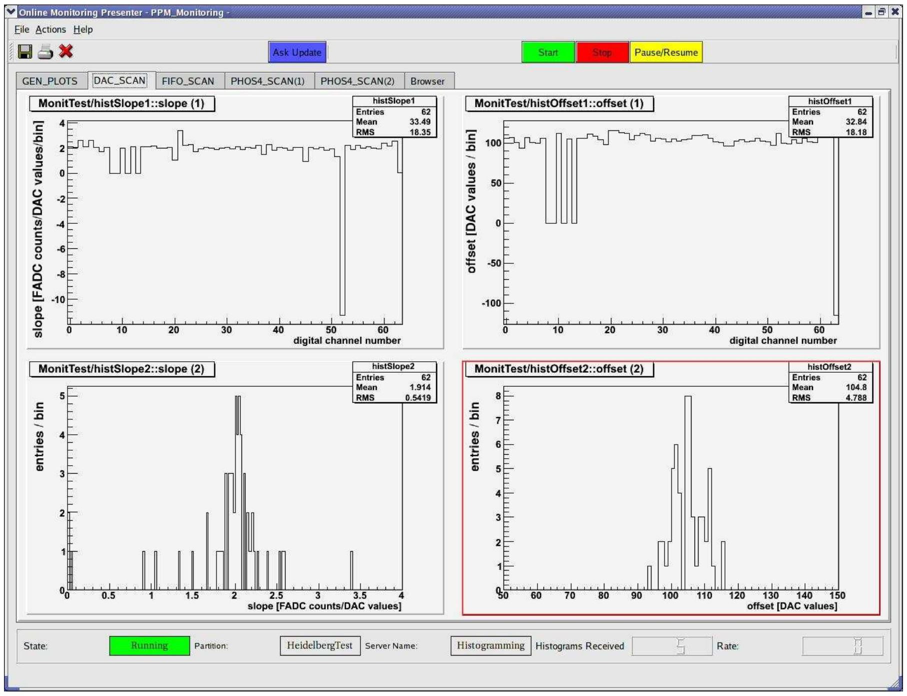

Figure 8: Monitoring results for a problematic DAC scan multi-step run.

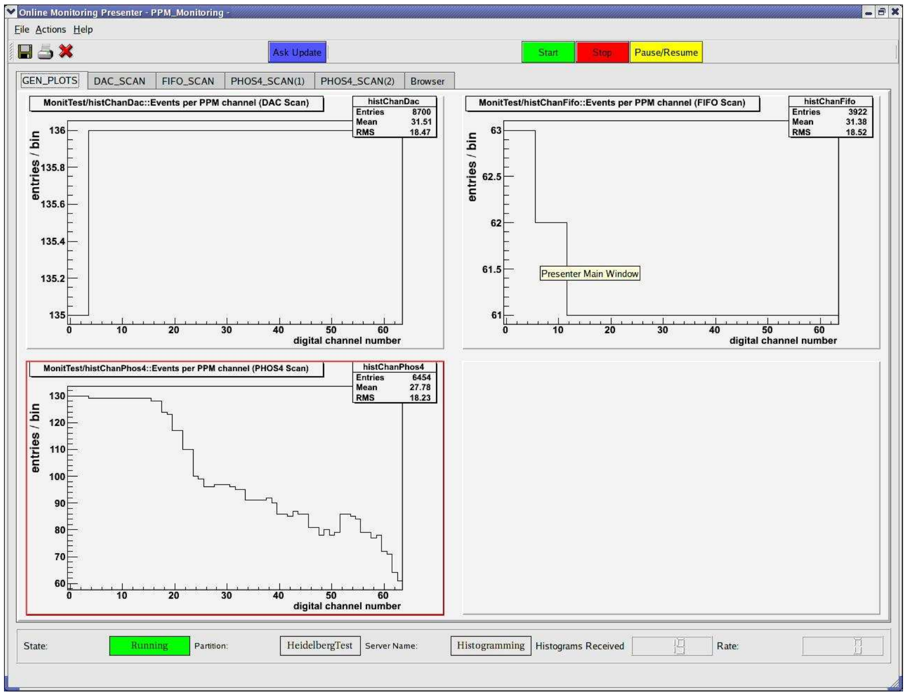

Figure 9: Uniform readout statistics of the PPM channels during the DAC and FIFO scans multi-step runs.

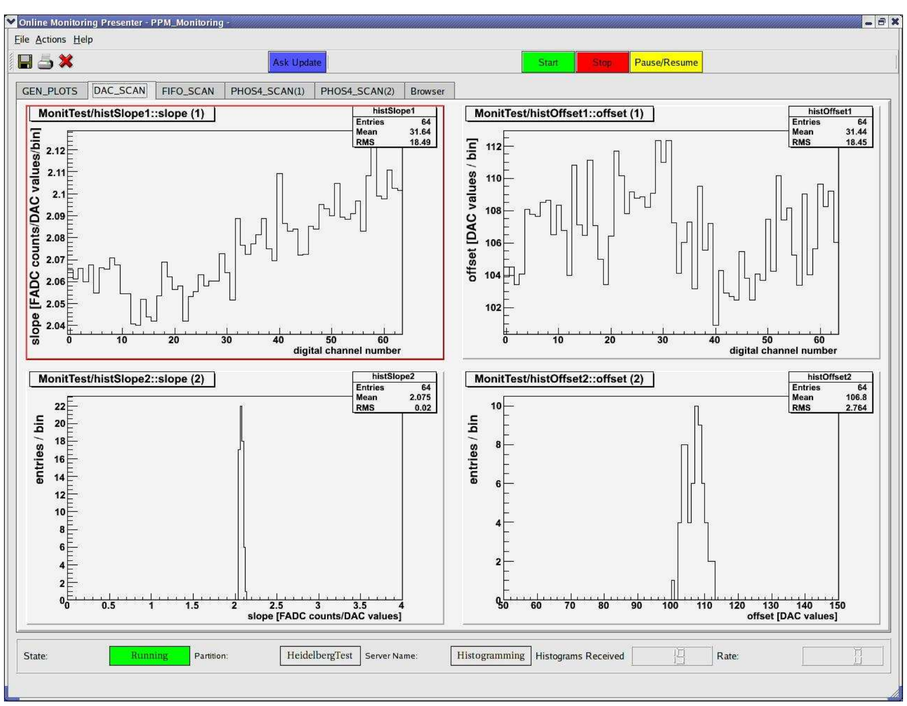

Figure 10: Monitoring results for the DAC scan multi-step run given in Figure 9.

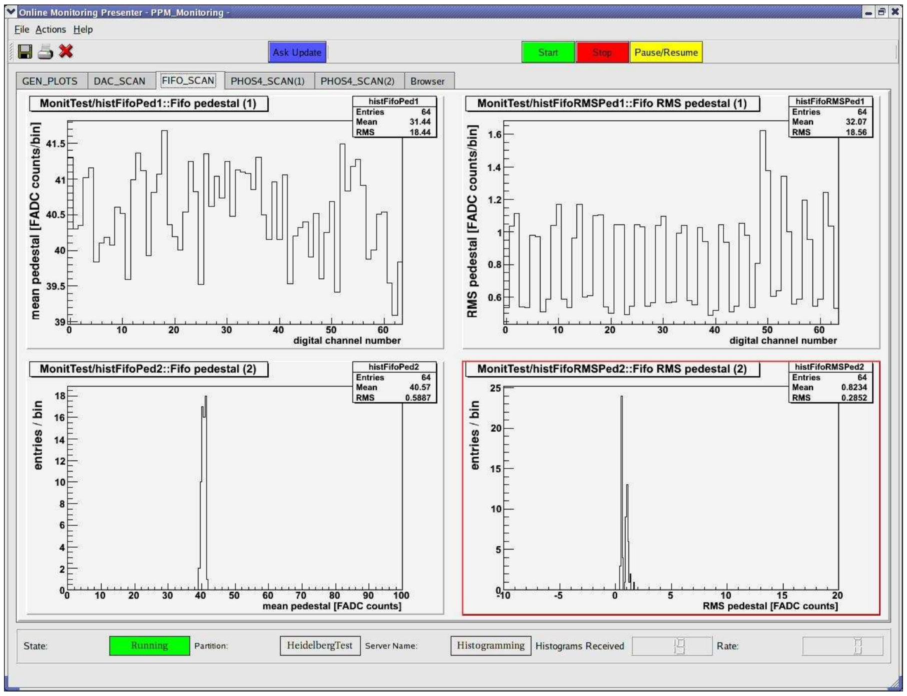

Figure 11: Monitoring results for the FIFO scan multi-step run given in Figure 9.

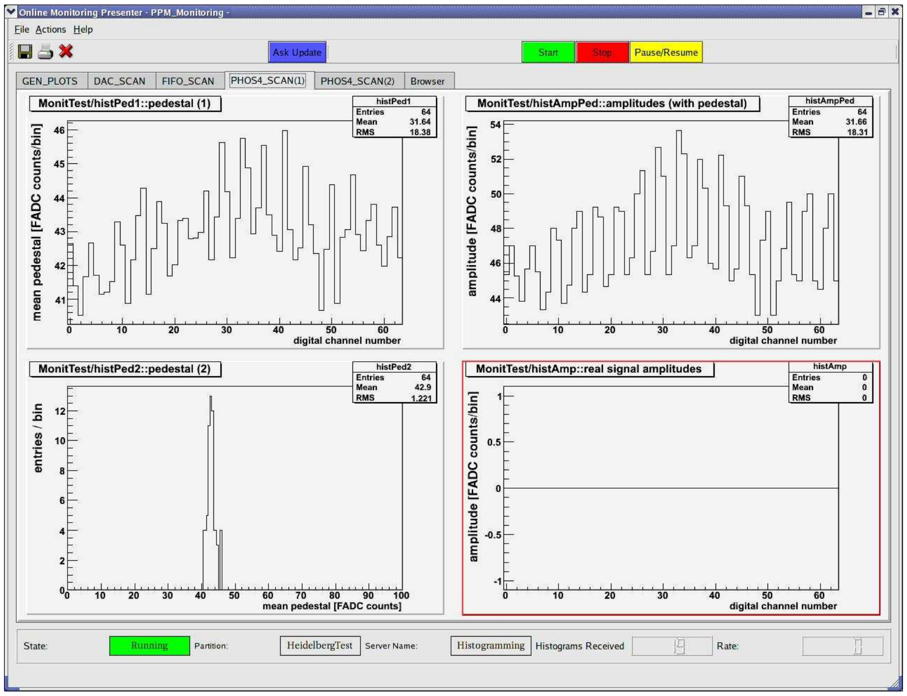

Figure 12: Monitoring results (first panel) for the PHOS4 scan multi-step run given in Figure 9.

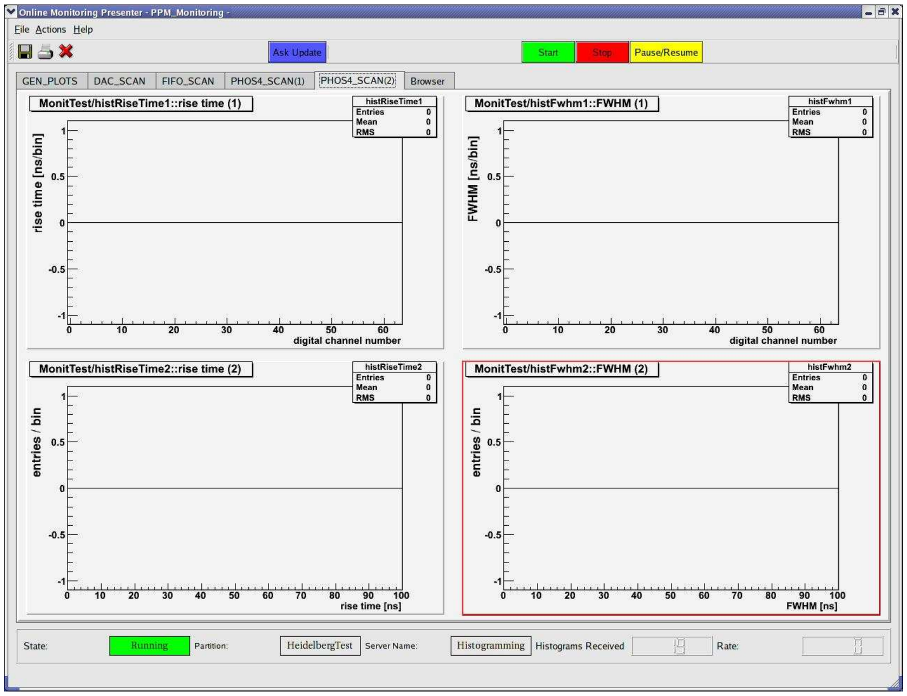

Figure 13: Monitoring results (second panel) for the PHOS4 scan multi-step run given in Figure 9.