

# Quality Assessment for the NA61/SHINE experiment at CERN

Alexander Sadovsky

*sadovsky@inr.ru*

*Institute for Nuclear Research RAS*  
for the NA61/SHINE Collaboration  
prepared  
for

XXXI INTERNATIONAL WORKSHOP  
“Neutrino physics at accelerators”  
DLNP, JINR, Dubna,  
January 27-29, 2009

- NA61 upgrade
- Offline (DST) QA
- Online QA
- Web-interface
- Bookkeeping



# NA61/SHINE upgrade

## NA61/SHINE experimental setup

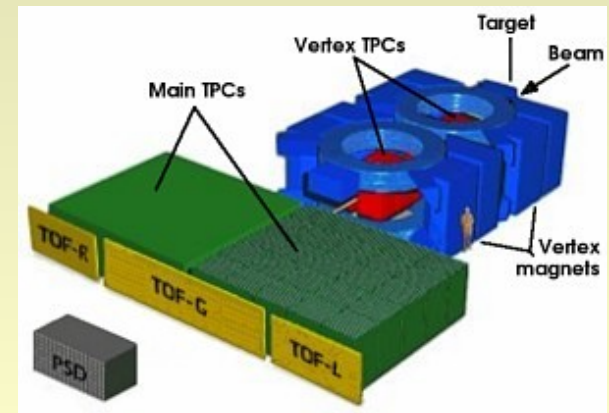
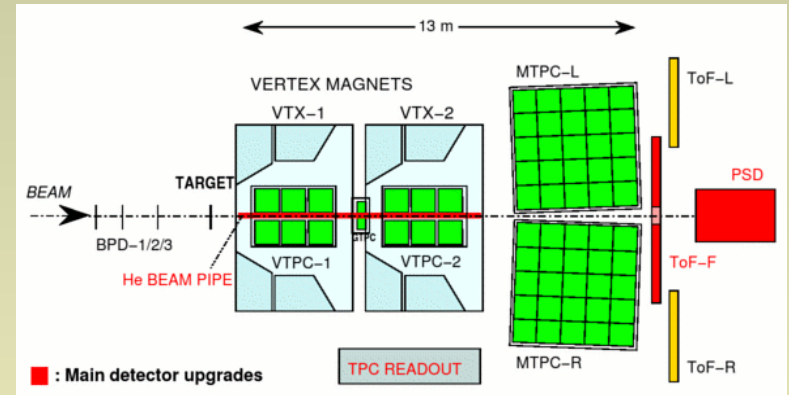
- 4 large volume TPCs + 1 gap TPC (63 subdetectors)
- 3 Time-of-Flight detectors
- 6 Beam Position Detectors
- 1 Projectile Spectator Detector

## Experimental runs

- 2007 /physics/ ~600 data files (p+C)  
    TDC readout & DAQ-upgrade (TPC event rate~100 Hz i.e. NA49x10)
- 2008 /DAQ-test/ ~100 data files (LHC incident)
- 2009 /physics/ ~10000 data files are expected (at least)

## Problem to keep under control:

- Detector control plots: [hist]x[file]  $\approx$  [100]x[10000]
- Possible only with semiautomatic tools
- Quick view-access to any (prepared in advance) QA plot
- Finally, DST file validation is needed!



# Offline DST QA

**histogram history** for DST QA was designed to allow for:

- help during DST preparation (DST="Data Summary Tape")
- time stability control
- expert decisions on a standard data representation view

It assumes:

- that experts provide a set of ~ (1...12) QA plots for each detector
- these histograms are saved per each DST file to a corresponding ROOT-histogram file
- GUI program (*hhistory*) is used to control long term detector stability
- decision is taken based on (logbook entries & expert opinions) for a limited set of files
- semiautomatic control in case of numerous amount of DST files with help of stability control

Long term aims:

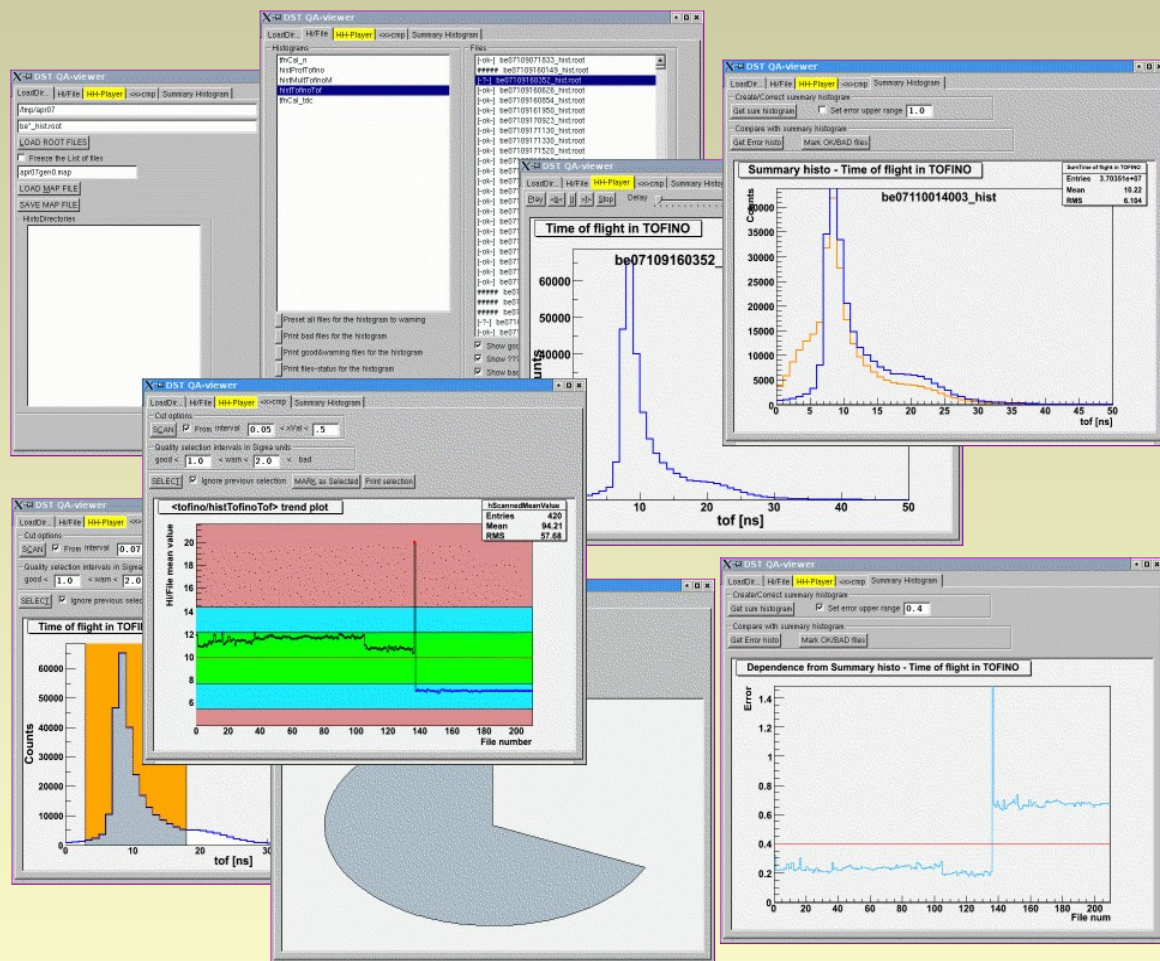
- incorporate DST-QA into the WEB-server with tree-like navigation (~bookkeeping)
- quick view-access to any (prepared in advance) QA plot
- DST file validation

# “hhistory” for offline-QA

What is it for?

- + light-weight GUI application
- + Linux/ROOT based
- + general purpose tool
- + histogram player
- + semiautomatic
- + <x> comparator
- + 1-dm shape control
- + 2-dm shape control
- + pattern ~ expert selectable
- + collaborating usage
- + all kinds of printouts
- + since 2003 ([HADES@GSI](#))

Helps in understanding of what was happening during past experiment

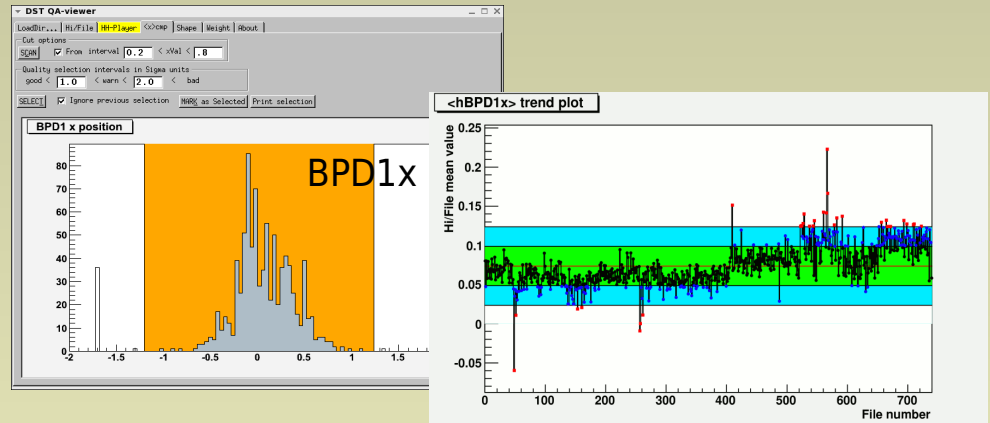


2007-2008 developments: in collaboration with T.Solovieva /diploma thesis/

# Examples of work

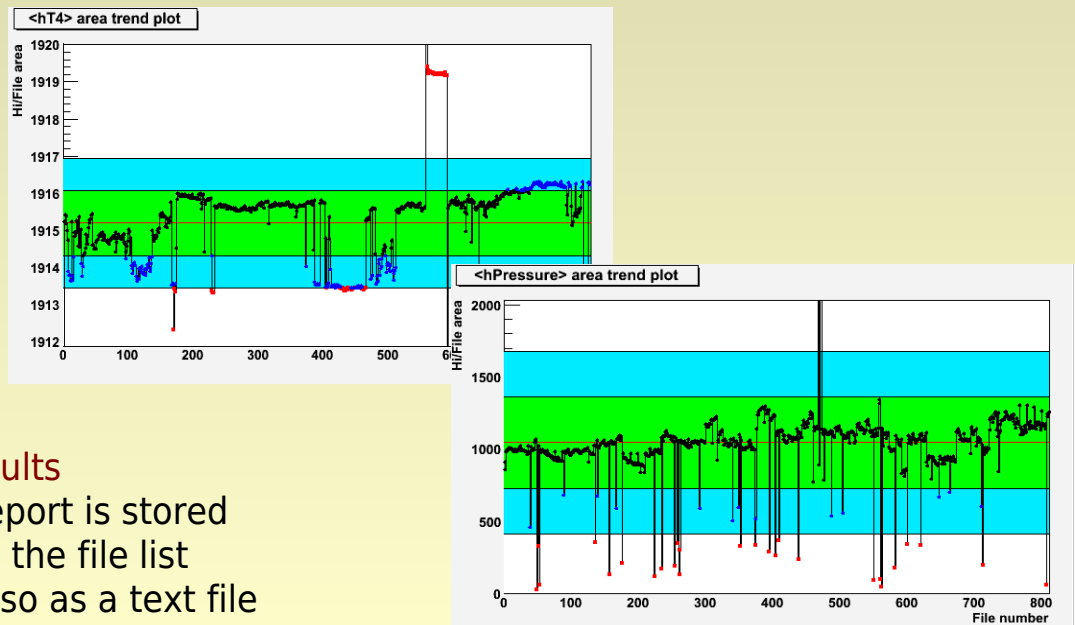
BPD example

peak mean value distribution  
and its trend plot



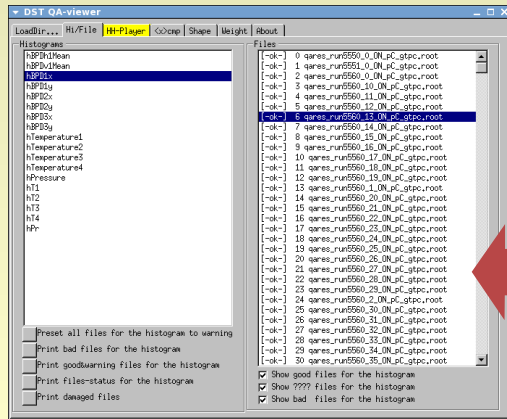
Pressure and temperature

examples of trend plots  
as seen from 2007'exp data



Results

report is stored  
in the file list  
also as a text file



# hhistory for online-QA

## Online? – rather - fast off-line as a solution for online quality control

Not to be mixed up with slow control

Created for shift crew, to help in realizing of drastic changes of experimental data flow

New DAQ: each file after 10-20s (delay up to ~3-5 minutes for full chain DAQ→histogr→Online-QA)

May involve deeper level of data analysis compared to “real online-control”

## Histogram history GUI-application was extended to allow for:

Display the last available DAQ-file status (pre-analyzed into a set of control histograms)

Quick look up to the histogram status reverse in time (file)

Creation of trend plot for any 1-dm histogram “on-fly” by user request

- Mean value comparator
- Kolmogorov-Smirnov test (1-dm and 2-dm)
- Chi<sup>2</sup> test

Specify multiple layouts (4x4) and display them

- in built-in window or
- in pop-up windows

## Status on Oct-2008:

Beta tests with 2007-data in DAQ→QA imitated mode

Version 3.14: several bugfixes

Installed at [runna61@na61pc002](mailto:runna61@na61pc002)

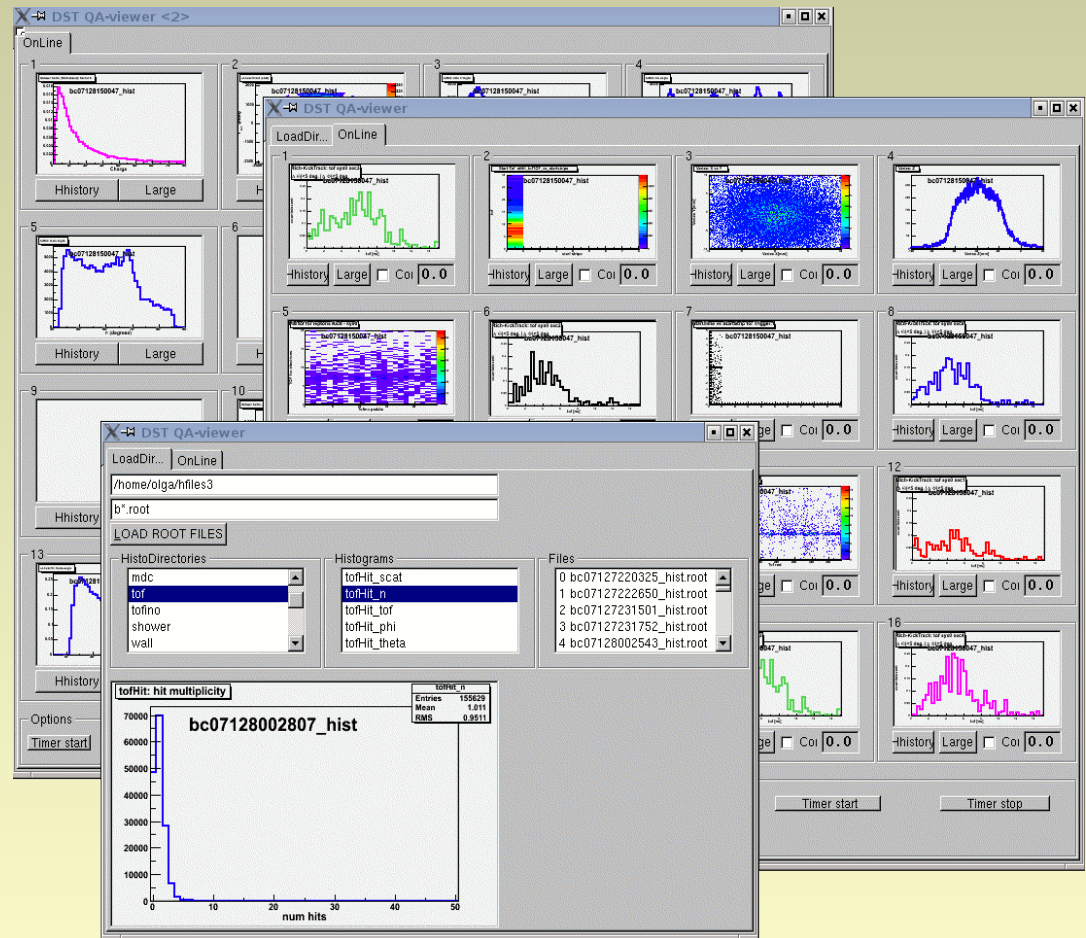
Unfortunately was not really tested in real experiment

# From offline- to online- QA

## Expansion towards online-QA

- + for NA61/SHINE
- + online option (fast offline)
- + data control for shift crew
- + histogram shape history
- + latency < (1..5) min depending on DAQ-QA flow
- + multiple online windows
- + page layout editor
- + interaction with offline module at any time

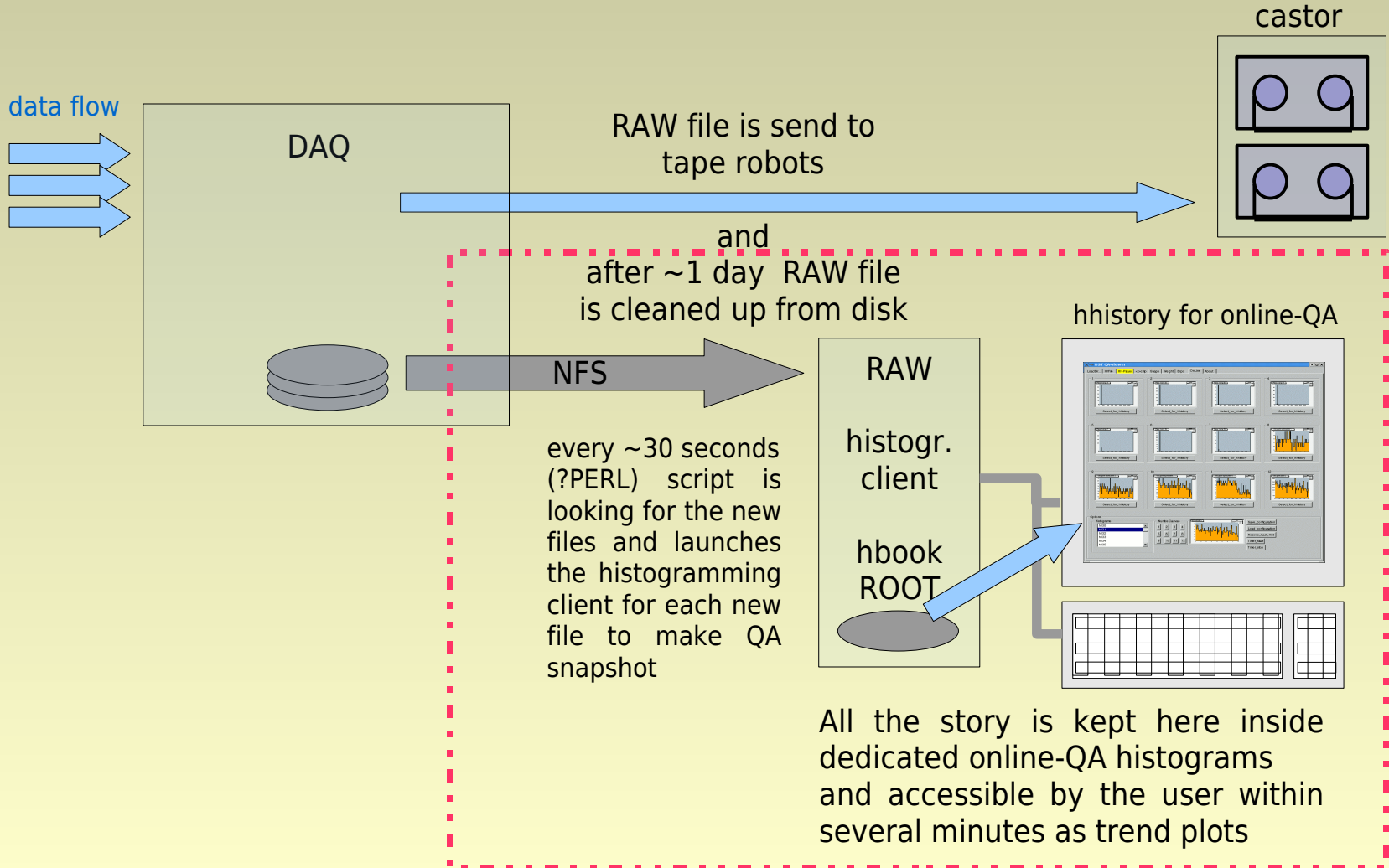
Combines offline problem recognition power for just in time application during data flow at experimental run



2008-2009 developments: in collaboration with O.Busygina /diploma thesis/

# Online QA functionality\*

DAQ→QA data flow (\*priv. communication with A.Laszlo, to be tested in real experiment)





# Online QA extension

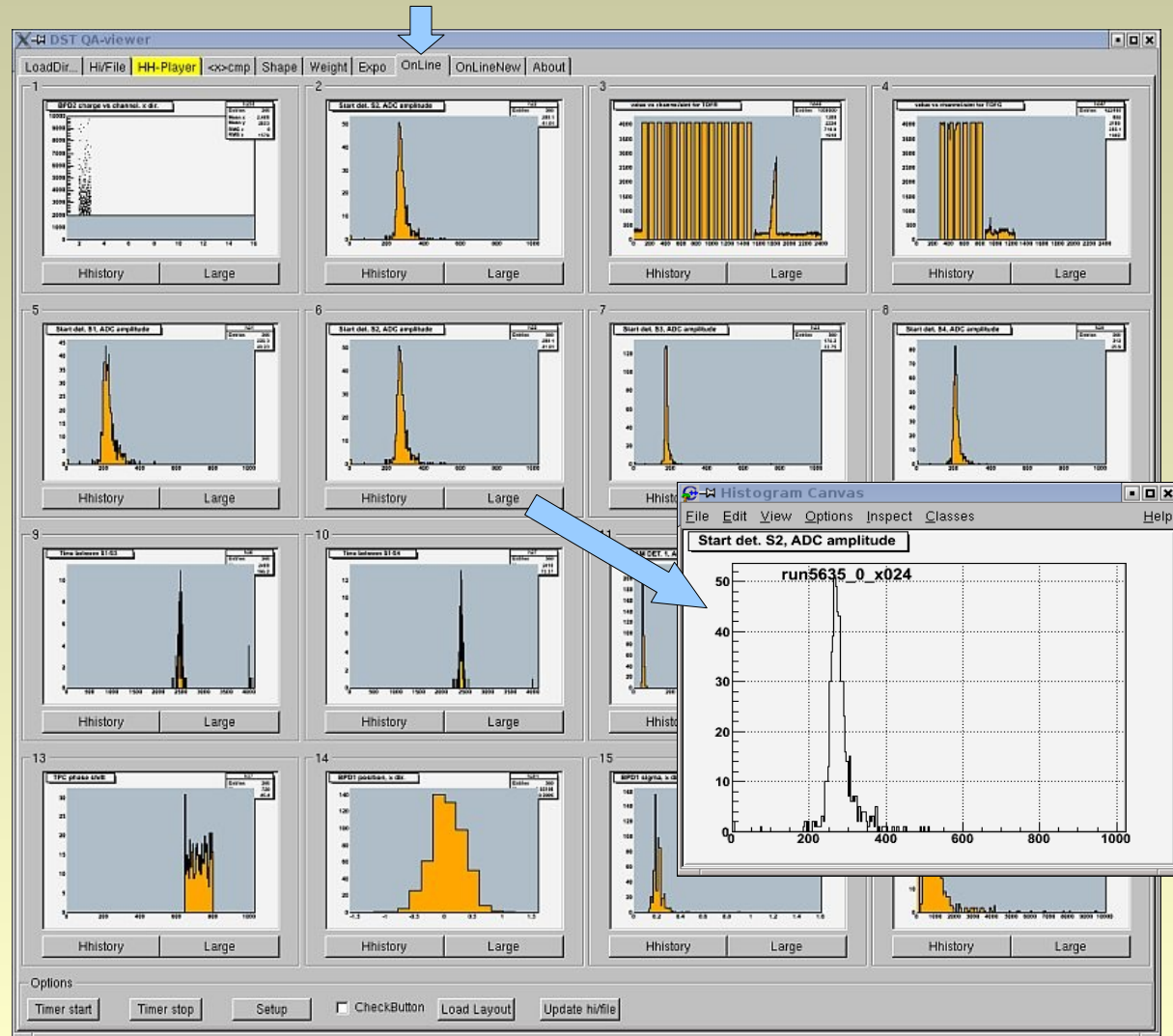
Prototype v3.14:

supports for:

- edit/load layout (16 hists),
- GUI histogram selection,
- show enlarged view,
- saving configuration and
- online update (from the latest ROOT file)

Allows “history” investigation of for an individually-selected plot.

**Diploma project:**  
**Olga Busygina**



# New layout / open window

The image displays a software interface with several windows and a central configuration window. The windows are arranged in a grid-like fashion, showing various plots and histograms. The central window is titled "DST QA-viewer" and contains a menu bar with options like "LoadDir...", "Hi/File", "HH-Player", "<x>cmp", "Shape", "Weight", "Expo", "OnLine", "OnLineNew", and "About". Below the menu bar, there is a "Saved files" list containing several configuration files, including "qaOnline\_BPDpositioSigr", "qaOnline\_MAIN.cfg", "qaOnline\_New-Configura", "qaOnline\_StartBeamQA.c", and "qaOnline\_TOF-L-R-G.cfg". The "Options" section includes a "Create new Online-Window" button. The "Options for Online-Window" section has a "HistoDirectories" field, a "Histograms" list with items like "h23", "h24", "h25", "h26", "h27", "h31", "h32", "h33", "h37", "h51", "h52", "h201", "h211", "h221", and "h231", and a "Viewer" section with a small plot. The "NumberCanvas" section has a grid of buttons from 1 to 16, with "h221" selected. The "Configuration" section at the bottom has a "Set name configuration:" field with the value "New-Configuration" and a "Save Layout" button. Three blue arrows point from the "DST QA-viewer" window to the "BPDpositioSigma", "StartBeam", and "New-Configuration" windows, indicating the process of opening or switching windows.

# Improving NA61 web-site

NA61/SHINE web site

<https://na61.web.cern.ch/na61/xc/>

What is it for?

Small redesign of the overall view:

larger fonts

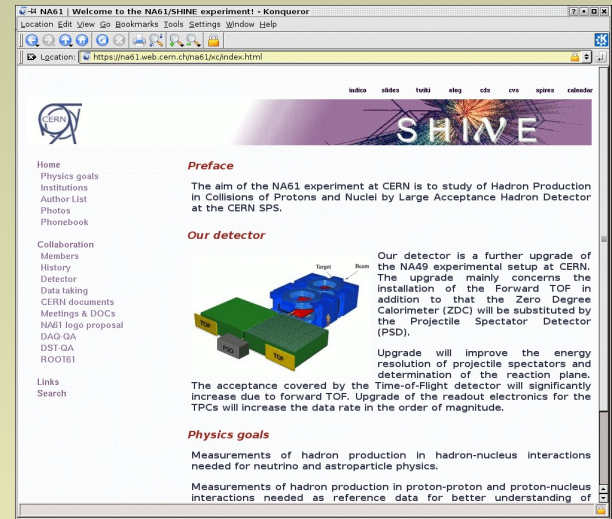
less freedom for picture positioning

Easy import of simple HTML texts from anybody

e.g. `~/user/public/www/somefile.html`

Run auxiliary CGI scripts on CERN server (for all kind of work which is suitable for a web-server)

- + keep all links in one place (not redoing standard tools like: TWiki, indico, ...)
- + online documentation generation
- + bookkeeping organization
- + possibility to organize centralized parameter storage
- + nearly online/offline monitoring for run/DAQ/DST status from outside
- + requests from external databases if necessary
- + possibility of simplified distributed documentation
- not for multiuser work at a low cost (rather use dedicated systems TWiki, CVS/CVN etc.)
- not for online/remote/slow control (due to possible network problems)
- not for CPU-intensive jobs

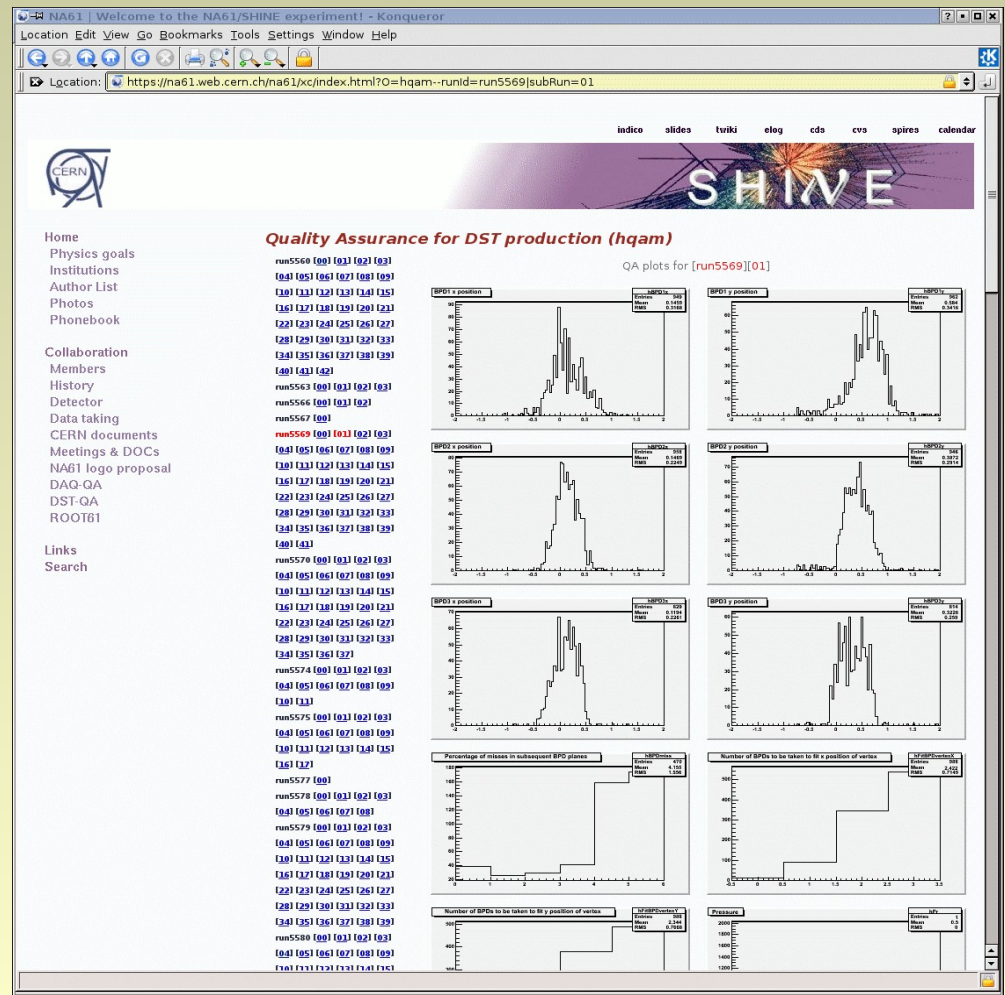


# Offline-QA web interface

## NA61/SHINE website development

- CERN hosted under UNIX
  - /-/ storage limit to 800Mb
  - needs external disk capacity DB?
- self made CMS
- employs file system, CGI/HTML/CSS
- Best suited for QA access from outside CERN
- Offline-QA
- DST-batch-jobs log-QA parser
- ? Bookkeeping
  - /requires ~ 10-50Gb disk space/

Shall it be read-open outside the collaboration?



# Log parser for batch jobs

The screenshots show the following components:

- Table 1: Summary of log files**

file-name	info	is file loaded	number of events	number of failures	number of errors	number of warnings	reading errors
run5600_0.raw	log	yes	521	0	4	0	0
run5597_0.raw	log	yes	1105	0	4	0	0
run5597_0.raw_x001	log	yes	1081	0	4	0	0
run5597_0.raw_x002	log	yes	331	0	4	0	0
run5581_0.raw	log	yes	1090	0	4	0	0
run5581_0.raw_x001	log	yes	1087	0	4	0	0
run5581_0.raw_x002	log	yes	1089	0	4	0	0
run5581_0.raw_x003	log	yes	1071	0	4	0	0
run5581_0.raw_x004	log	yes	1089	0	4	0	0
run5581_0.raw_x005	log	yes	1085	2	43	6	0
run5581_0.raw_x006	log	yes	1091	2	43	8	0
run5581_0.raw_x007	log	yes	1057	1	23	4	0
run5581_0.raw_x008	log	yes	1051	0	4	0	0
run5581_0.raw_x009	log	yes	189	0	4	0	0
run5611_0.raw	log	yes	1094	1	24	4	0
run5611_0.raw_x001	log	yes	1047	1	23	1	0
run5611_0.raw_x002	log	yes	1058	0	4	0	0
run5611_0.raw_x003	log	yes	1074	0	4	0	0
run5611_0.raw_x004	log	yes	1092	0	4	0	0
run5611_0.raw_x005	log	yes	1093	0	4	0	0
run5611_0.raw_x006	log	yes	1087	0	4	0	0
run5611_0.raw_x007	log	yes	780	1	24	1	0
run5625_0.raw	log	yes	1157	0	4	0	0
run5625_0.raw_x001	log	yes	1130	0	4	0	0
run5625_0.raw_x002	log	yes	1131	0	4	0	0
run5625_0.raw_x003	log	yes	384	0	4	0	0
Summary for 26 files	---	---	25064	8	260	24	0

- Form:** A list of runIDs (run5603 to run5626) with checkboxes and a 'Select/deselect' button.
- Details View:** Shows debugging information for a selected file, including 'General job information', 'Used run period(s)', and a list of log events.

## Log parser of job results on batch farm

- Convenient access to each file
- Very limited disk space
- Relatively slow for simultaneous parsing of many files on default CERN's web-server computer

# Bookkeeping

## Task(s) from physics to data:

It is good to remember conditions which were used while producing the Data Summary Tapes (DST) as well as for simulation.

It is convenient to have possibility to check the data taking conditions and performance of certain detectors for any beam time starting from “physics” and ending by the Quality plots, calibration parameters etc.

## Possible solution:

We can organize central gate responsible for the user interface between a standard computation task and the user. As soon this is done we have possibility to trace what jobs and under which parameter conditions were computed.

This can be realized as a web-form page which is creating certain start-up script which is used as a 1-st step of any analysis. If this is done conveniently this will also be profitable for the end user to use that form instead of programming start-up script from scratch every time.

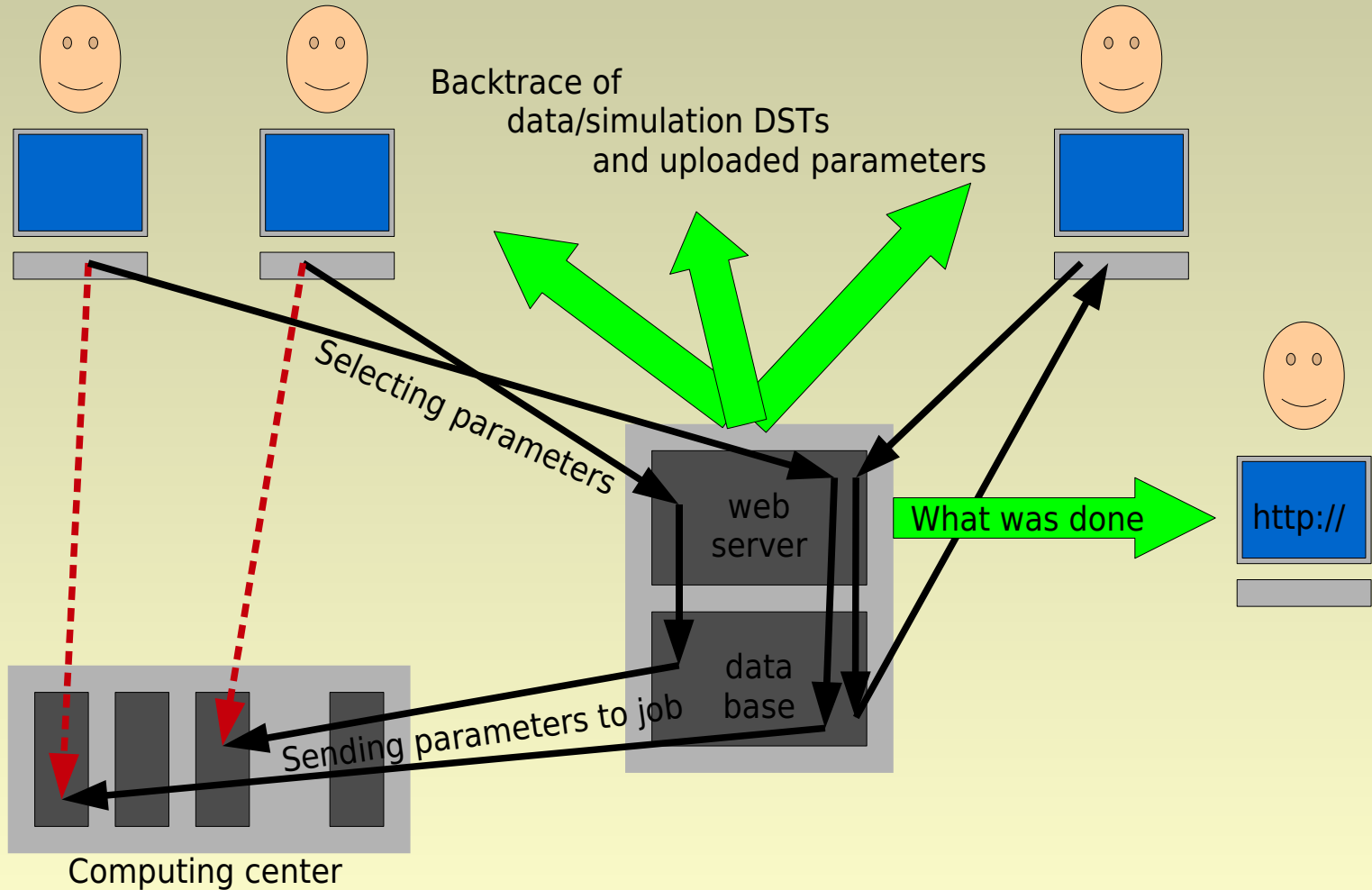
## Problems:

Computations are done on various computers  
Possibly at different research centers (GRID)

On the early stage calibration- (and also simulation-) parameters might be stored/created not at central repository.

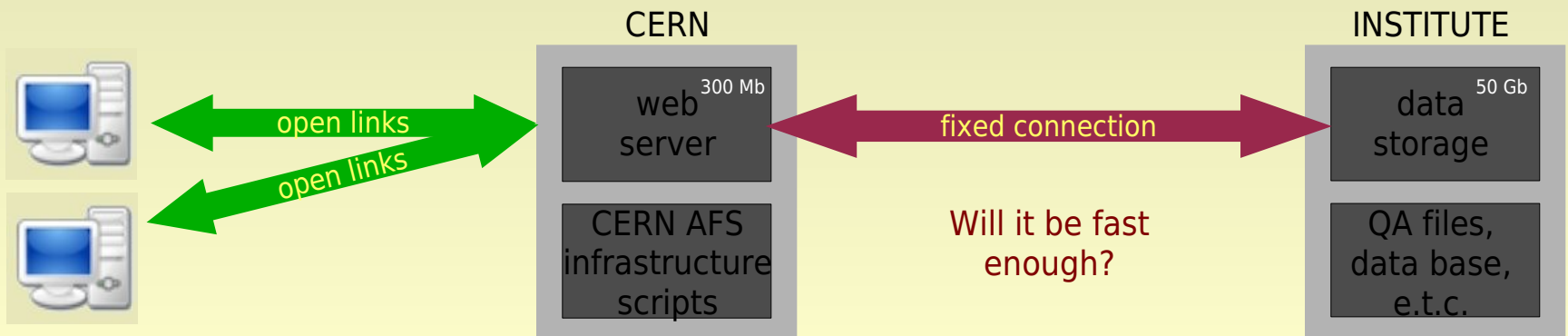
It's hard to convince people to commit detailed documentation.

# Bookkeeping proposal



# Web server limitations

- **CERN rules:**
  - Own AFS quota < 800Mb
  - Access to other users within AFS if directory has proper permission
  - No access to tape robots
  - Own PC located in the office is not secured against AC power loss.
  - Own PC (web+data base server) can not be located at comp.center.
- **Possible solution (to be tested):**
  - Remote computing center for data storage (also for the data base)
  - Web server information exchange using ip↔ip connection
  - bookkeeping test to be done.





# Current status

- **DST batch-log-parser:**  
shell and web supported (still some problems with permissions)  
demonstrated possibility of work, to be tuned into common interface/DB.
- **Offline DST QA:**  
more histograms to be added by detector experts.
- **Online QA:**  
integration with offline QA, bugfixing,  
real test to be done in 2009.
- **Web-site/QA:**  
prototypes are ready,  
external data-storage is under consideration,  
bookkeeping test to be done.

# BACKUP SLIDES

# NA61 data taking plans

2009: August 12 to November 16

p+C at 31 GeV/c	3 weeks (T2K)
$\pi$ +C at 158, 300 GeV/c	2 weeks (C-R)
p+p at 6 energies	6 weeks (SIM)
p+p at 158 GeV/c	2 weeks (high pT)

2010:

p+p at 158 GeV/c	11 weeks (high pT)
------------------	--------------------

2011:

30+30 at 6 energies	6 weeks (SIM)
p+Pb at 158 GeV/c	6 weeks (high pT)

2012:

10+10 at 6 energies	6 weeks (SIM)
p+Pb at 6 energies	6 weeks (SIM)

2013:

100+100 at 6 energies	6 weeks (SIM)
-----------------------	---------------

6 energies are: 10, 20, 30, 40, 80, 158 GeV/c.

Expected data yield per year ~ 12 weeks ~ 25000 files (of 1Gb size) per year!

**Strong demands to QA!**

# Log-file parser for DST prod.

https://test-na61qa.web.cern.ch/test-na61qa/cgi-bin/ii.html

## Test-NA61qa

Batch-log-QA

### NA61 batch QA

Here you can specify the batch output and see corresponding diagnostics generated by an automatic process.

batch output directory: (standard UNIX form)

password: (letters and digits)

Click OK to start

file-name	is file loaded from tape
run5602_0.raw	log yes
run5597_0.raw	log yes
run5597_0.raw_x001	log yes
run5597_0.raw_x002	log yes
run5582_0.raw	log yes
run5582_0.raw_x001	log yes
run5582_0.raw_x002	log yes
run5582_0.raw_x003	log yes
run5624_0.raw	log yes
run5624_0.raw_x001	log yes
run5624_0.raw_x002	log yes
run5624_0.raw_x003	log yes
run5624_0.raw_x004	log yes
run5624_0.raw_x005	log yes

Summary for 14 files

Page loaded.

run5603  
 run5601  
 run5598  
 run5584  
 run5583  
 run5597  
 run5602  
 run5582  
 Select/deselect all

Page loaded.

```
Location: /bin/ii.html?O=batchQA--selectRunDebug=/afs/cern.ch/na61/Production/wwwdir/run5603/LSFJOB_965962/STDOUT=2156
2151 DSPACK 1.602, 1 Aug 2007 (dsdd, server: prodna61_18755_lxb8584)
2152 DSPACK 1.602, 1 Aug 2007 (dsread, server: prodna61_18755_lxb8584)
2153 Read one event
2154
2155
2156
2157
2158
2159
Run: 5603 Event: 00059
2160 DSPACK 1.602, 1 Aug 2007 (dsdd, server: prodna61_18755_lxb8584)
2161 DSPACK 1.602, 1 Aug 2007 (tblbla, server: prodna61_18755_lxb8584)
2162 vt1 christina correction is /afs/cern.ch/na61/Software/pro/RES_CORR_ZG/init/vt1.corr
2163 vt2 christina correction is /afs/cern.ch/na61/Software/pro/RES_CORR_ZG/init/vt2.corr
2164 mtl christina correction is /afs/cern.ch/na61/Software/pro/RES_CORR_ZG/init/mtl.corr
2165 mtr christina correction is /afs/cern.ch/na61/Software/pro/RES_CORR_ZG/init/mtr.corr
2166 Open: /afs/cern.ch/na61/Software/pro/RES_CORR_ZG/init/vdrift_2007.txt
2167 christina par file is /afs/cern.ch/na61/Software/pro/RES_CORR_ZG/init/vt1.corr
Page loaded.
```

```
Location: /afs/cern.ch/na61/Production/wwwdir/run5603/LSFJOB_965962/STDOUT
General job information
Submitted run period(s):
EXECUTE started: 1 times
converter started: 1 times
See details:
Executing error events:
00001 event 00002 event 00003 event 00004 event 00005 event 00006 event 00007 event 00008 event 00009 event 00010 event 00011
event 00012 event 00013 event 00014 event 00015 event 00016 event 00017 event 00018 event 00019 event 00020 event 00021 event 00022 event 00023 event 00024 event 00025
event 00026 event 00027 event 00028 event 00029 event 00030 event 00031 event 00032 event 00033 event 00034 event 00035 event 00036 event 00037 event 00038 event 00039 event 00040 event 00041
event 00042 event 00043 event 00044 event 00045 event 00046 event 00047 event 00048 event 00049 event 00050 event 00051 event 00052 event 00053 event 00054 event 00055 event 00056
event 00057 event 00058 event 00059 event 00060 event 00061 event 00062 event 00063 event 00064 event 00065 event 00066 event 00067 event 00068 event 00069 event 00070 event 00071
event 00072 event 00073 event 00074 event 00075 event 00076 event 00077 event 00078 event 00079 event 00080 event 00081 event 00082 event 00083 event 00084 event 00085 event 00086
event 00087 event 00088 event 00089 event 00090 event 00091 event 00092 event 00093 event 00094 event 00095 event 00096 event 00097 event 00098 event 00099 event 00100 event 00101
event 00102 event 00103 event 00104 event 00105 event 00106 event 00107 event 00108 event 00109 event 00110 event 00111 event 00112 event 00113 event 00114 event 00115 event 00116
event 00117 event 00118 event 00119 event 00120 event 00121 event 00122 event 00123 event 00124 event 00125 event 00126 event 00127 event 00128 event 00129 event 00130 event 00131
event 00132 event 00133 event 00134 event 00135 event 00136 event 00137 event 00138 event 00139 event 00140 event 00141 event 00142 event 00143 event 00144 event 00145 event 00146
event 00147 event 00148 event 00149 event 00150 event 00151 event 00152 event 00153 event 00154 event 00155 event 00156 event 00157 event 00158 event 00159 event 00160 event 00161
event 00162 event 00163 event 00164 event 00165 event 00166 event 00167 event 00168 event 00169 event 00170 event 00171 event 00172 event 00173 event 00174 event 00175 event 00176
event 00177 event 00178 event 00179 event 00180 event 00181 event 00182 event 00183 event 00184 event 00185 event 00186 event 00187 event 00188 event 00189 event 00190 event 00191
event 00192 event 00193 event 00194 event 00195 event 00196 event 00197 event 00198 event 00199 event 00200 event 00201 event 00202 event 00203 event 00204 event 00205 event 00206
event 00207 event 00208 event 00209 event 00210 event 00211 event 00212 event 00213 event 00214 event 00215 event 00216 event 00217 event 00218 event 00219 event 00220 event 00221
event 00222 event 00223 event 00224 event 00225 event 00226 event 00227 event 00228 event 00229 event 00230 event 00231 event 00232 event 00233 event 00234 event 00235 event 00236
event 00237 event 00238 event 00239 event 00240 event 00241 event 00242 event 00243 event 00244 event 00245 event 00246 event 00247 event 00248 event 00249 event 00250 event 00251
event 00252 event 00253 event 00254 event 00255 event 00256 event 00257 event 00258 event 00259 event 00260 event 00261 event 00262 event 00263 event 00264 event 00265 event 00266
ROOT-started
```

# histogramming client

Since *hhistory* does nothing but displaying and analysing histograms from ROOT-files located as a certain directory one needs a client which is creating and filing such histograms.

Currently *check\_raw\_zoltan* -client is used to create control online-QA histograms

- + was already used before
- + works
- does not include control plots for some detectors (TPCs?)
- needs some modifications when data structure changes (new DAQ)
- written in C with support for HBOOK/PAW, plots has to be converted to ROOT format

? May be we shall stick to DSPACK related version also for online-QA

- + no need in two independent conversions of RAW data into the ready-to-use format
- + will almost merge with offline-QA and will need less manpower to support
- must be slower, but the cost is CPU which is in the end cheaper compared to bug hunting
- it can be, that some important low-level information is not transmitted into DSPACK (?)