E949 TN K-028

# DeToX: LeCroy HV monitor program

David E. Jaffe

September 4, 2002

**Abstract**

DeToX monitors the PMT high voltages for the range stack, barrel veto, barrel veto liner and beam elements. It reports non-standard conditions, compiles a list of channels with non-standard conditions for each run for use in the off-line database and produces ntuples to allow study of the time history of all channels.

## 1 Causes of common error messages

**Any deviation from standard conditions is assumed to constitute an error and is reported as such. Standard conditions are defined as Demon running, all high voltages on and stable with readback voltages within $\pm 10$ volts of the set voltages.**

"Stable" means that the RMS of the difference between the set voltage and readback voltage is less than some threshold. This threshold is typically set at 0.51 volts. The single lowest and highest deviation of the readback from the set voltage is excluded from the calculation of the RMS to reduce sensitivity to outliers. The accumulation of statistics is nominally on a run-by-run basis.

The readback voltages are obtained from the mainframes every 30 seconds and error conditions are reported as soon as they are detected.

The set voltages are obtained once at the beginning of each run and once at the beginning of the inter-run period.

**Known sources of non-serious error messages:**

1. The readback voltage for a single channel occasionally (roughly once per day) deviates from the set voltage by hundreds of volts. It is not clear whether the voltage actually changes or the readback is simply flawed. In any case this condition generates an "out of tolerance" message that can be ignored if the subsequent readback is within tolerance. Subsequent "HV unstable" error messages may appear for this channel if the RMS exceeds the threshold.

2. If the voltage setting for a channel is changed with the Demon panel, DeToX will complain because it obtains the set points from the input file used during Demon's initialization.

3. If any voltage is turned off, DeToX will complain.

4. If Demon is stopped and restarted with new settings (i.e., voltages are turned back on), but the run number does not change, DeToX may falsely report errors because it is unaware of the new settings until a new run number is encountered.

All other error messages should be regarded as serious.

DeToX [1] checks the status of the run about every 30 seconds. When the status changes (a new run starts or a current run ends) DeToX stops accumulating statistics, writes summary files and ntuples and exits. A new DeToX job should start after 30 seconds, if Demon is running, and begin accumulating statistics. So DeToX may persist in reporting errors for a minute or two when the status of the run changes.

## 2 How it works

Demon reads voltages from all mainframes to update the panel. The readback voltages are also written in a file every thirty seconds. The file name is unix-timestamped, i.e. `/e787d/bnlonl/online_logfiles/demon/hv.export.000000997394982` , and contains the voltage and virtual station number (VSN). When Demon starts it creates a database (i.e., `/e787/local/online/demon/log/hv.08-09-01.15:36:48` ) with an obvious time stamp from the files in `/e787/local/online/demon/data/hv.XX` where XX is the two letter acronym of each subdetector (`XX` = RS, BV, ...), that contains the mapping of the VSN to the detector element name, the high voltage channel (mainframe, slot and channel) and voltage setting.

A single DeToX job will first read the latest Demon database file to obtain the correspondence between VSN, detector element, set voltage and channel. This information is stored stored in an ntuple that is written out at the end of the job. DeToX then goes to look for the latest `hv.export.000...` file and obtains the readback voltages from that file. This information is also stored in an ntuple. DeToX tries to answer the following questions:

1. Is the VSN from `hv.export.000...` actually in the database?
2. Is the HV readback present for all channels known in the database?
3. Is the HV set on?
4. Is the HV on according to the readback?
5. Is the HV within tolerance?
6. Is the HV stable?

If the answer to any of the questions in this list is "no", the channel is "bad".

The criteria for the items 1 and 2 are obvious. If a VSN is not in the database (item 1), it could signal a readback error (this has never been observed). Occurances of item 2 have been observed. Occasionally the readback for a channel is not present, these are reported as "garbled" in the Demon log files.

The criteria for items 3 and 4 are $V_{\mathrm{min}} < |V| < V_{\mathrm{max}}$ where $V$ is the set or readback voltage, respectively, and typically $V_{\mathrm{min}} = 50$ and $V_{\mathrm{max}} = 3000$ volts. The criteria for item 5 is typically $|V_{\mathrm{read}} - V_{\mathrm{set}}| \leq 10$ volts. Finally, if the RMS of $V_{\mathrm{read}} - V_{\mathrm{set}}$ exceeds a threshold, typically 0.51 volts, the channel is reported as unstable as described in Section 1.

DeToX obtains the current run number from `/e787d/bnlonl/online_logfiles/dui_log/dui_nrun.txt` and determines the status of the current run from `/e787d/bnlonl/online_logfiles/runlog` . The status of a run is either active (run in progress) or inactive. When a change in status is detected, DeToX outputs two files. One contains ntuples with the set voltages, the run number, all the accumulated readback voltages and the time of each readback. The ntuple file name contains the run number and the time of the DeToX run was started (i.e. `ntuples/detox.42558.997447837.rzn` ). There can be multiple ntuple files with the same run number for several reasons. Inter-run DeToX data will have the run number of the last completed run as will data for runs written to the "trashcan". The second output is an ASCII file containing a one line summary off all bad channels. For data taken during a run, the file name prefix is `hvr`. For inter-run data, the prefix is `hvi`. The file name also contains the run number and the approximate end time of the run (i.e., `RunSum/hvr_sum.042558.0745` ) in hours and minutes. For inter-run data, the file suffix contains the time the file was created in hour, minute and second format `RunSum/hvi_sum.042558.08.50.03` .

Here are the contents of `RunSum/hvr_sum.042558.0745` :

```
    1
*Codes:0=Good,1=HV Off(read),2=HV Off(set),3=Out of Tol.,4=no HV readback,5=Not in database,6=HV unstable,
*   Element VSNhex VSNdec Code Nread  Vset   Vlo   Vhi   meanV  RMS(V) (mean&RMS have lowest & highest readings excluded as of 25jul01)
 BV P 37 1    2194   8596    3   161 -2331 -2353 -2302 -2332.9     7.7
```

The first line is the number of bad channels listed. The next line contains a short description of the code given in the 4th column of the table. The non-obvious column headers in the next line

- VSNhex and VSNdec are the VSN in hexadecimal and decimal, respectively.
- Nread is the number of readbacks,
- Vset is the set voltage,
- Vlo and Vhi are the lowest and highest readback voltages,
- meanV is the average voltage and
- RMS(V) is the RMS of Vset-Vread.

All voltages are in volts. The average and RMS have the single lowest and highest readback voltages excluded. If there are multiple reasons why a channel is bad, the first classification is reported here. For example, Vlo and Vhi both indicate that the voltage was out of tolerance and RMS(V) exceeds the threshold of 0.51 volts. These summary files can be integrated into the CFM database similar to the method currently used for ADC pedestals for off-line analysis.

# 3 What to do when a serious error may have occured

## 3.1 Make sure the error is serious

Repeated, unrelenting error messages not covered in Section 1 usually require action. It is probably useful to obtain a recent time history of bad channels to verify that there is a serious problem. Here's the procedure:

```
cd /949/Test/Demon
run.ReToX N RMSthres FlowThres
```

where `N` is the number of hours that you want to see in the time history, `RMSthres` is the threshold for classifying a channel as bad and `FlowThres` is the minimum number of readbacks with $|V_{\mathrm{read}} - V_{\mathrm{set}}| > 50$ volts for classifying a channel as bad. The default values of 24 hours, 0.51 volts and 10 times, respectively, are usually satisfactory.

Here is a typical output from the script for the default values:

```
 HV database taken from /e787/local/online/demon/production/programs/../../log/hv.08-10-01.09:11:36
 Logfile is log/ReToX.997457473.log and histogram file is hists/retox.997457473.rzn Fri Aug 10 11:31:14 EDT 2001
 Times of first and last input ntuples are Aug 9 12:06 and Aug 10 09:12
```

The script isn't clever enough to insure that you always get to see exactly `N` hours back and the program cannot read the ntuple being filled by the current DeToX job. If the current run is still active, you can force the end of the current DeToX job with the script `Gracefully.End.DeToX` or simply end the run to get the latest data.

The log file lists the ntuple files that were read, the channels that were classified as bad and the name of the postscript file with the time history and other information. Figures 1, 2, 3 , 4 , 5 and 6 show the six postscript pages corresponding to `log/ReToX.997457473.log`. Figure 1 is a "box" plot showing all channels exceeding `RMSthres` with the largest box being the worst channel. Figure 2 shows the average of $V_{\rm set} - V_{\rm read}$, Figure 3 is similar to Figure 1 but shows the elements by subdetector instead of mainframe. Figure 4 shows the distribution of $V_{\rm set} - V_{\rm read}$ for all bad channels with the worst channel in the upper left hand corner. The time histories of these channels are shown in Figure 5 (the ordering of channels on this page is not necessarily the same). In this case that is BV P 37 1, it appears to need replacement. The next worst channel (upper right hand corner) BV P 45 3 was replaced just after the beginning of this data sample. This is clear from Figure 5. BV P 45 3 was performing poorly and the module was replaced at $\sim 2$ hours in the figure. The HV for all other channels in the new module differed by a few volts from the old module, so they were all falsely identified as "bad". The time history of a channel is plotted using HBOOKB so that there is one entry per bin with no empty bins. Entries are plotted at the center of each bin and therefore may not correspond precisely to the time the voltage was actually read.

The final Figure 6 shows the distributions of $V_{\rm set} - V_{\rm read}$ and RMS for all channels. "Bad" channels are shaded. Note the overflow in the lower plot.

### 3.2 Fixing problems

Sometimes unstable or out-of-tolerance channels can be remedied by turning the voltage off and back on with Demon. The second potential remedy is to cycle the power on the mainframe (turn off the voltage with Demon first). The third remedy is to replace the offending module. When replacing the module the main thing is to make sure the cables are correctly reconnected while not causing other problems. Some modules are extremely difficult to replace. Take care not to damage the cables (I broke one connecter) or to dislodge the power cord (I've done this many times).

## 4 More details

### 4.1 Scripting

DeToX is run within the script `run.DeToX` that is normally started by the script `Start.Up.DeToX`. `Start.Up.DeToX` should be run on bnlku9. It checks for a currently running `run.DeToX` process, if one is not found, it starts `run.DeToX`. `run.DeToX` starts the script `get.latest.hv.export.file` and tries to start `Is.Demon.Running`. `get.latest.hv.export.file` writes the name of the latest `hv.export....` file to a local file that is read by DeToX. Then `run.DeToX` executes `DeToX.exe`. If DeToX has not read the latest `hv.export....` file, it does so. If DeToX has read the latest file, then it waits (typically 15 seconds) and tries again. If DeToX finds that the "latest" file has not been updated after many attempts (typically 10), it believes that Demon is no longer running and exits. `run.DeToX` then tries to start a new `DeToX.exe` after checking if Demon is running. This requires that `Is.Demon.Running` be running on the same machine as Demon (currently bnlku3). `Is.Demon.Running` is a script that first checks to see if Demon is running and, if so, puts the time of the latest `hv.export....` file in a file that is read by `run.DeToX` and compared with the current time. If the time of the latest file differs by more than 10 minutes from the current time, `run.DeToX` pauses for 5 minutes and tries again. After 10 such unsuccessful tries, it pops up a xmessage window asking if Demon and `Is.Demon.Running` are actually running, and continues checking to see if Demon is running.

### 4.2 Log files

`Start.Up.DeToX` directs all output of `run.DeToX` into a log file of the form `log/detox.09Aug01.1441.log` . The start date and time are in the file name. The output of each DeToX jobs is directed to a log file of the form `log/DeToX.42558.997475086.log` . The current run and start time (in unix format) are in the file name.

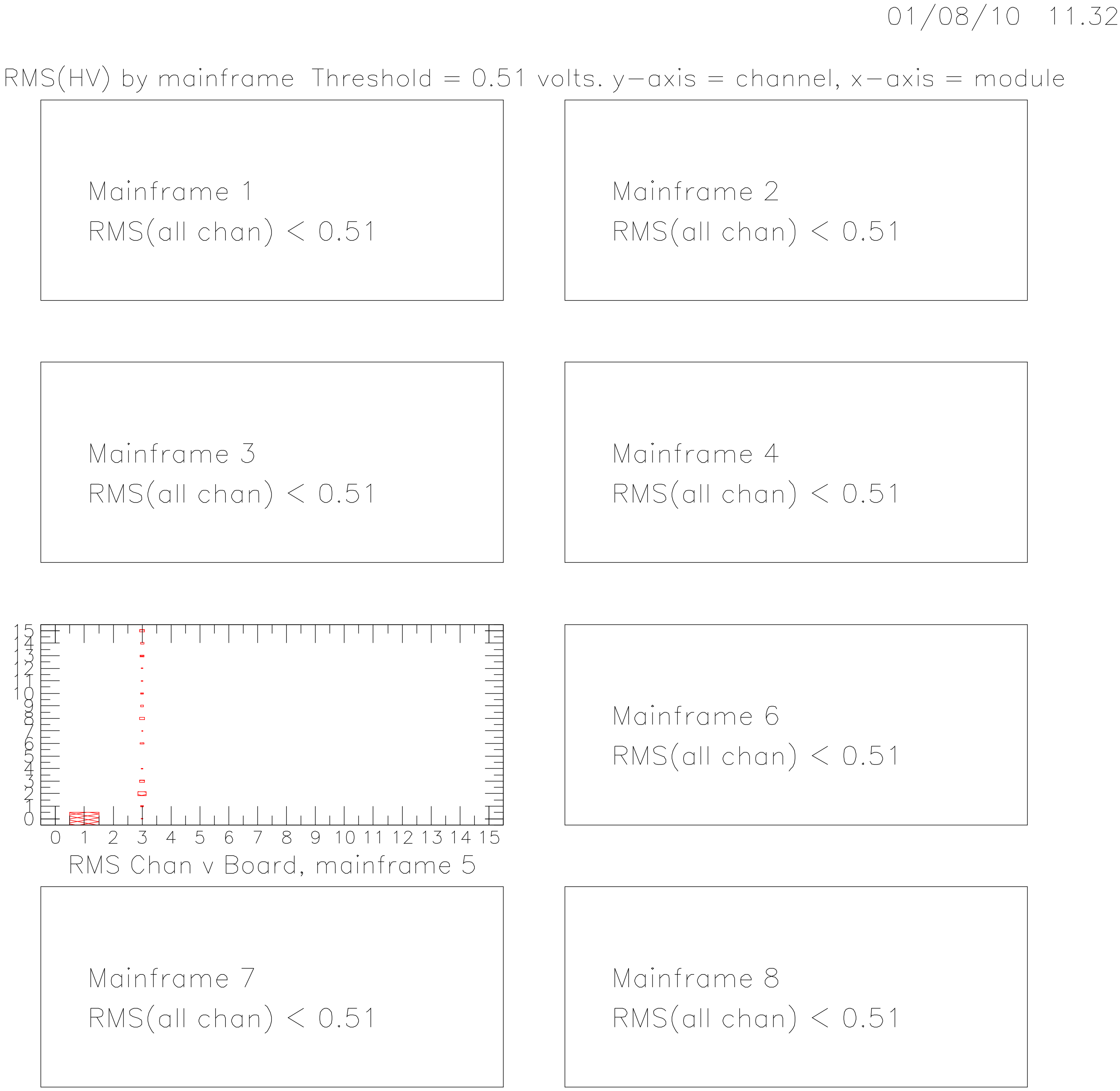


Figure 1: Page 1

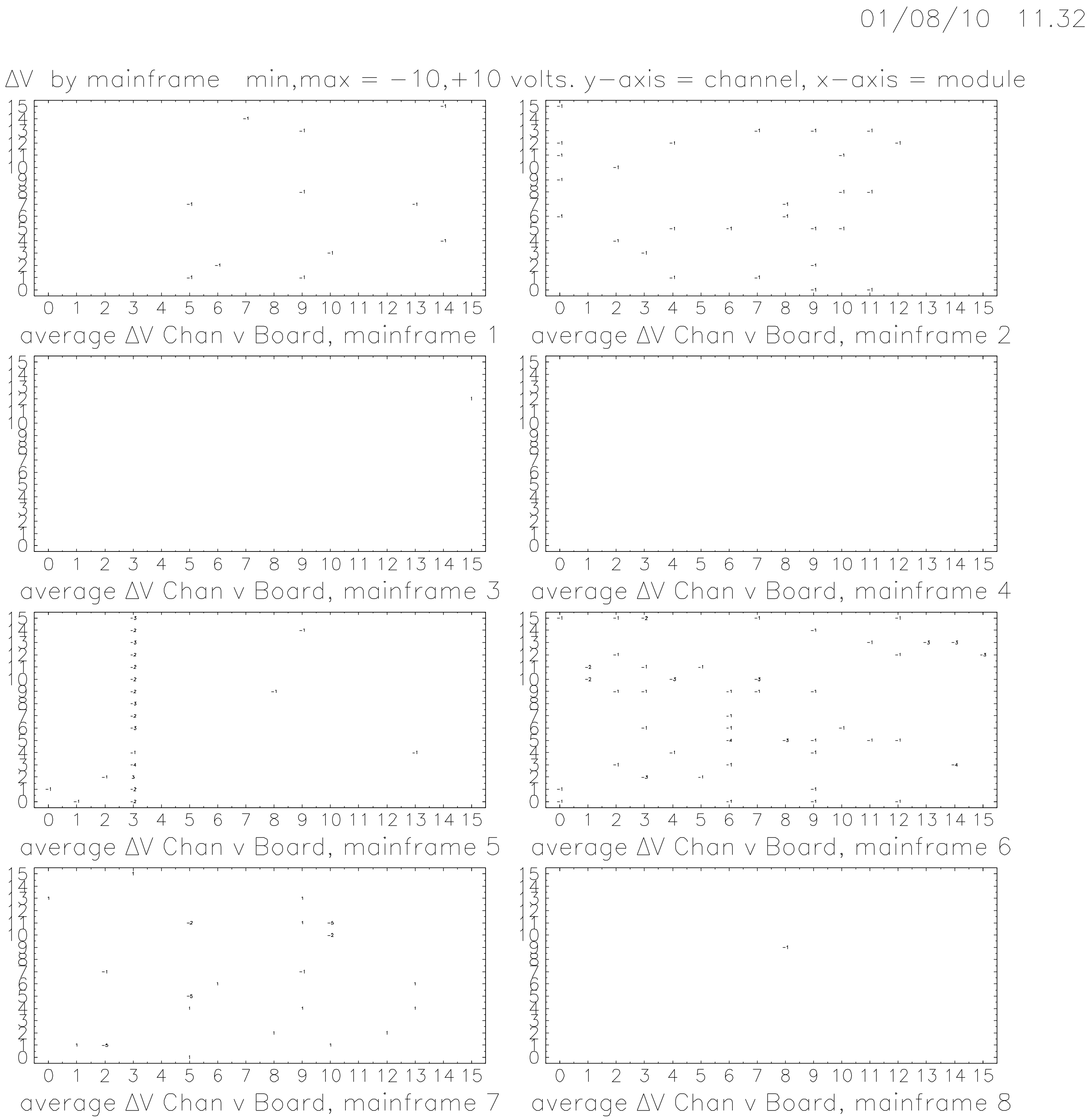


Figure 2: Page 2

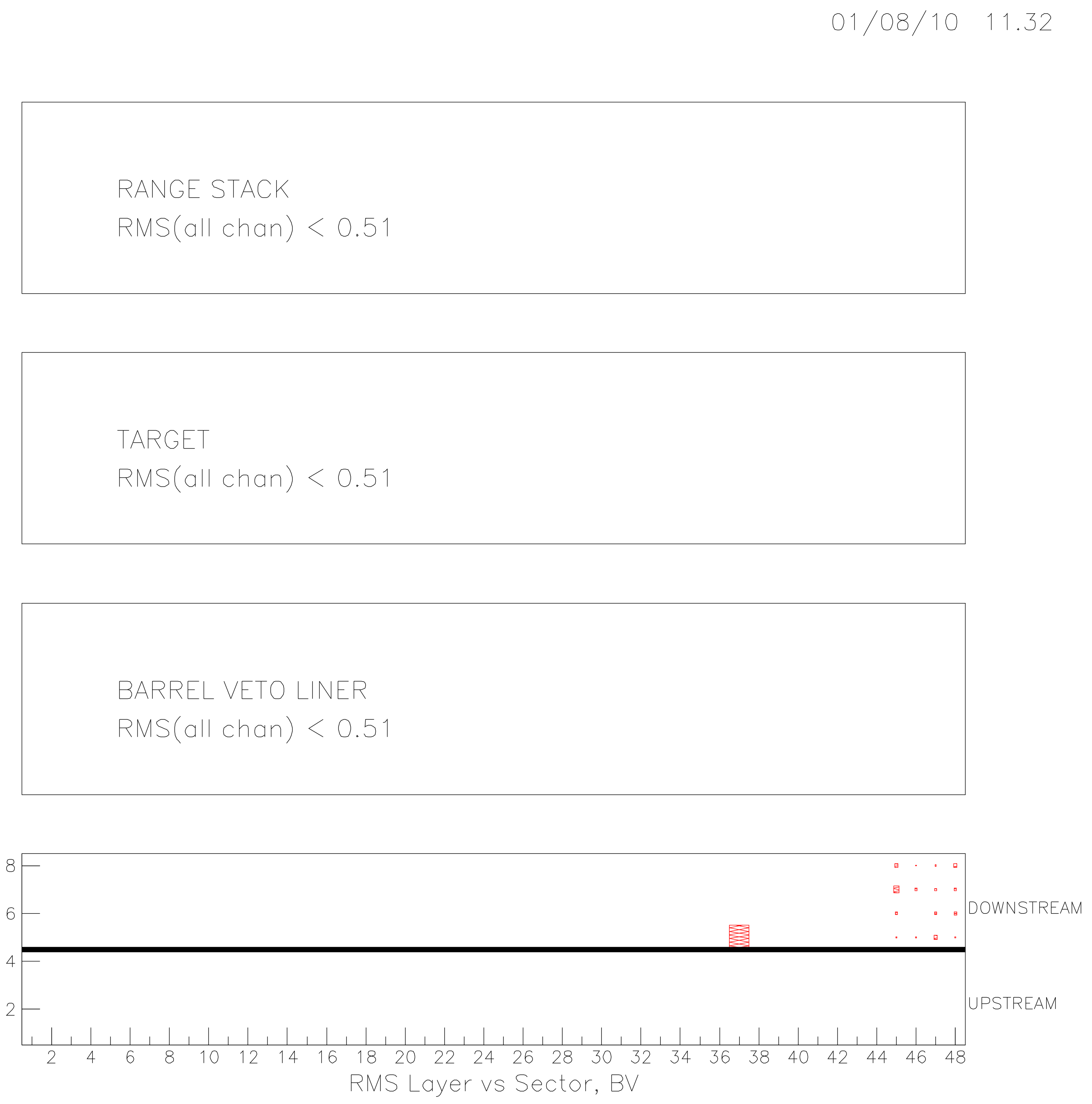


Figure 3: Page 3

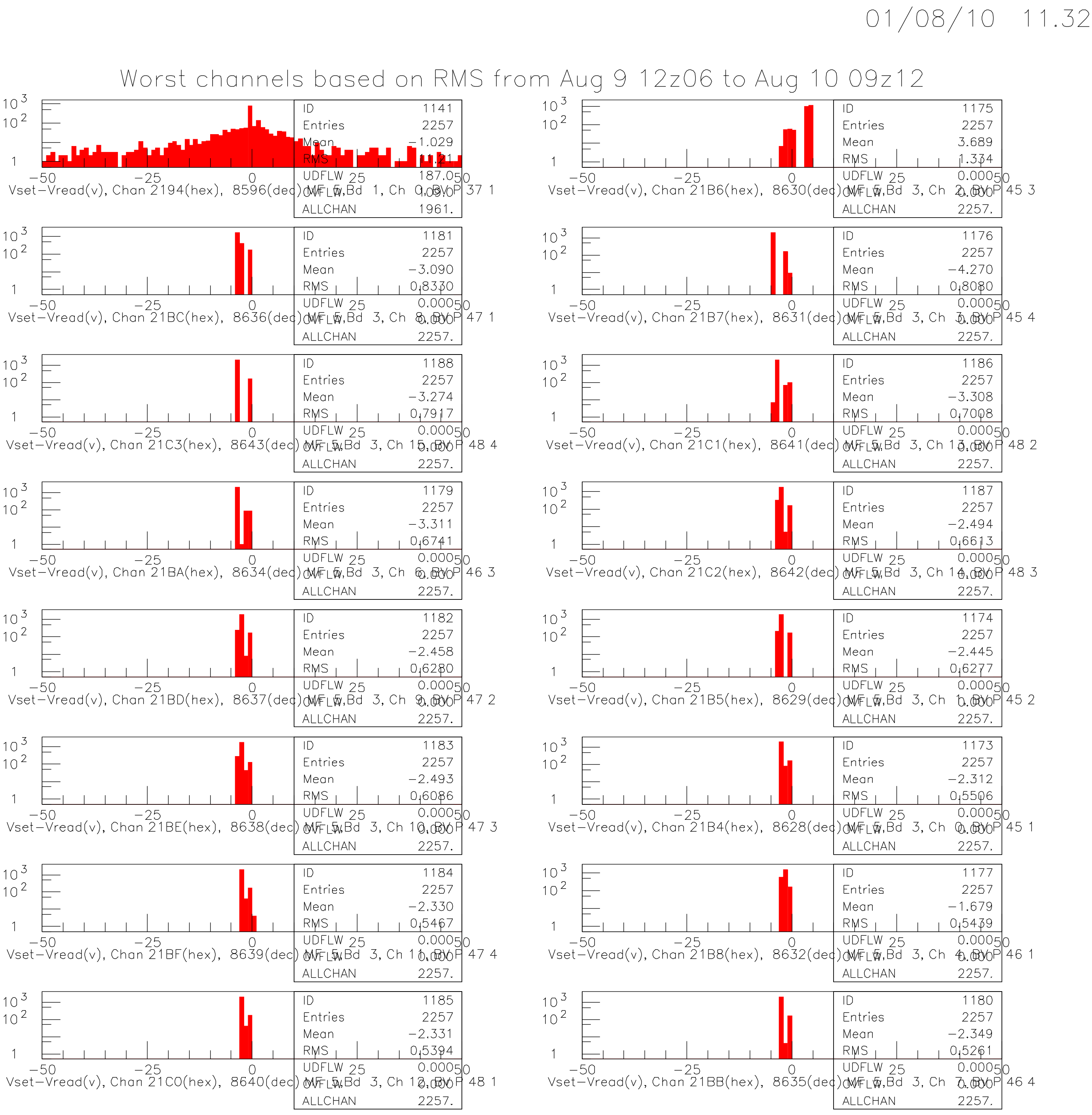


Figure 4: Page 4

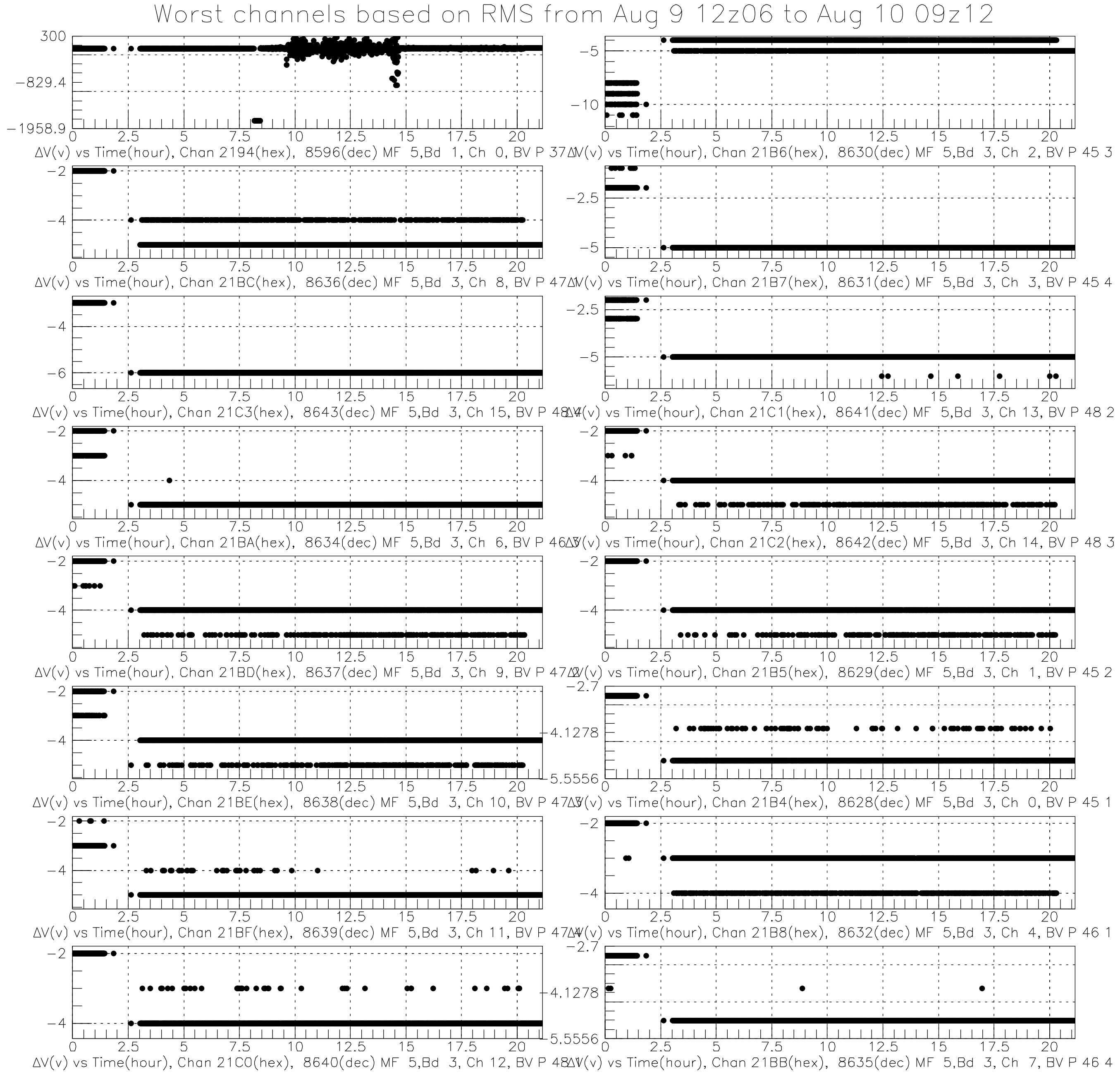


Figure 5: Page 5. $V_{\mathrm{set}} - V_{\mathrm{read}}$ *vs* time for all bad channels. Note that the ordinate autoscales.

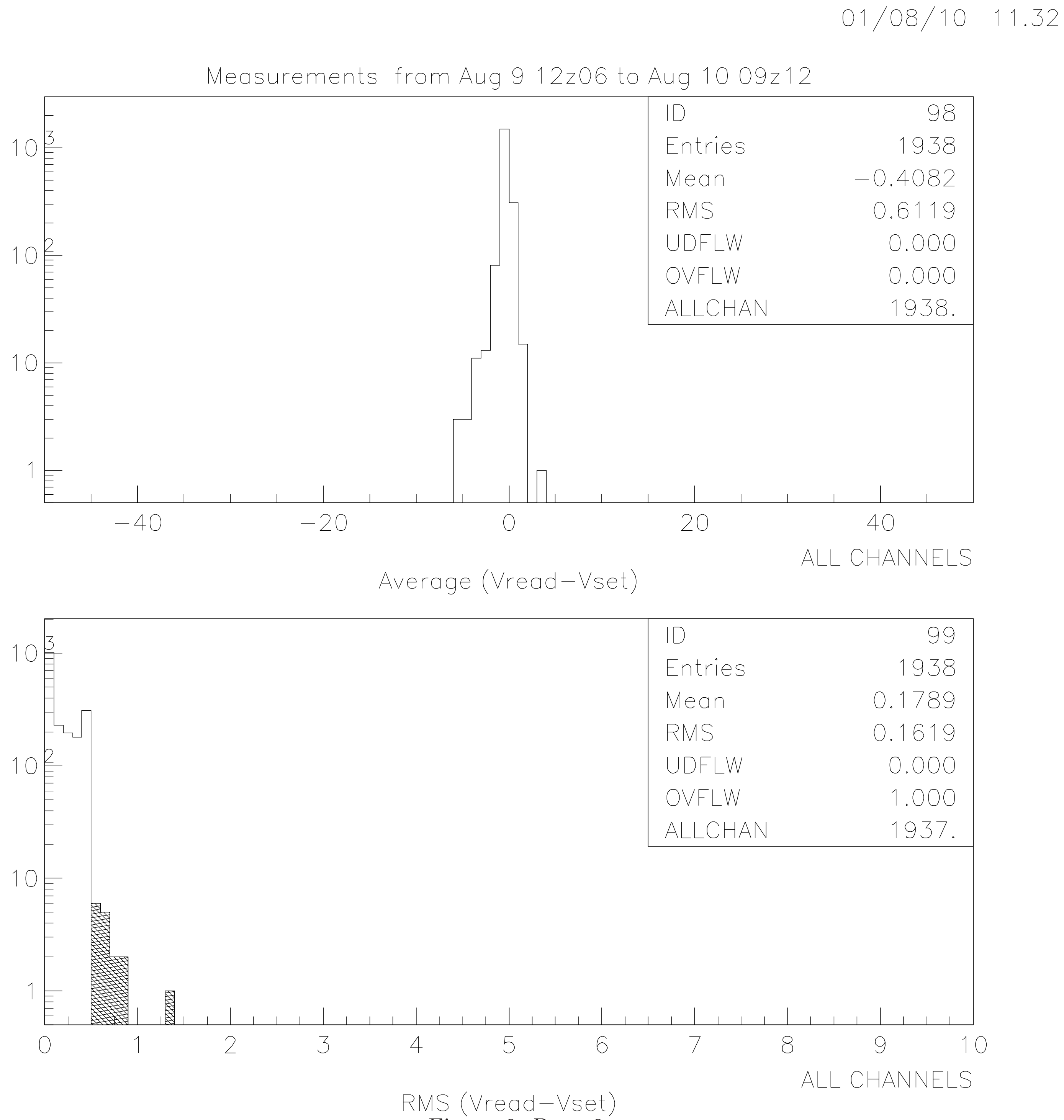


Figure 6: Page 6

### 4.3 Error messages

Here's an example of the pop-up window display:

```
  ***** DeToX HV Error Report 10/08/:1 16.35.52   24th call *****
 HV reading was taken Fri Aug 10 16:35:34 2001
 Nchan HV error message     Run    -1
   1915 Good
     17 HV Off(read)
      6 Out of Tol.
 ----------------- Too many HV channels with errors.
 Full list of bad channels in full.hv.error.file
```

The top line gives the date [1] and time of the report and the number of calls made to the HV checking routine. The next line gives the date and time the HV reading was taken. Needless to say, the two times should be close together. The little table that follows gives the number of channels for each 'HV error message'. If there are less than 20 bad channels, they will be listed in the window. Otherwise, the user has to look in a file. Inter-run data is reported with a run number of -1.

DeToX is clever enough to delete the window with the previous message before popping up a new message, but does not delete its last message if no more errors occur. So check to see that the time of the report is current before you go into a panic.

Besides the file `full.hv.error.file`, you can get more information from the file `current.hv.error.file`. Both these files are updated every 30 seconds by DeToX so don't be surprised when they change.

## 5 Suggestion box

Suggestions are welcome and should be emailed to the author at `djaffe@bnl.gov`.

## References

[1] "DeToX" is a acronym contrived from Demon Tolerance eXamination.

[1] The year is reported as ":1" instead of "01" due to a Y2.001K bug in the CERN routine DATIMH.