How to use BigStackV1

BigStackV1 is a ROOT macro useful for analysing the data which is collected by the online DAQ program, *InoC11*. This program runs and stores the raw data generated in the DAQ PC.

The macro file *BigStackV1.C*, the raw data as well as the output files generated are all stored in a directory named *BigStack* within the home directory of the user *lamps* (Password: lamps123) in the host *ino06.hecr.tifr.res.in (158.144.55.103)*. This host will be referred henceforth as Analysis PC.

You are assumed to be working on or logged into the Analysis PC. Before you start the analysis, you need to copy the data file(s) from the DAQ PC to the Analysis PC. This is how it is done:

```
[lamps@ino06 ~]$ cd BigStack/
[lamps@ino06 BigStack]$ ssh lab@158.144.55.103
lab@158.144.55.103's password: <lab-user>
[lab@c217-gw lab]$ ssh lab@192.168.55.100
lab@192.168.55.100's password: <labuser>
Last login: Thu Aug 16 12:38:49 2007 from 192.168.55.1
[lab@localhost ~]$ cd INODAQ/
[lab@localhost INODAQ]$
```

On the above command prompt, you issue the following command to transfer the data files:

```
scp run<run number>.* lamps@ino06.hecr.tifr.res.in:/home/lamps/BigStack
```

You must of course give the password for the *lamps* user, when prompted. You will come back to Analysis PC by pressing *CTRL+D* twice.

Some times you want to have a look at the data of a run while the run is still in progress in DAQ PC. The data are written into temporary files called *temp.eve* and *temp.mon* in the DAQ PC, during the run. So, in this case you will give the following command to transfer data on-fly into the Analysis PC:

```
scp temp.* lamps@ino06.hecr.tifr.res.in:/home/lamps/BigStack
```

In this case, you will rename the data files on the analysis PC. It is done in the following way for example for run number 76.

```
[lamps@ino06 BigStack]$ mv temp.eve run0076.eve
[lamps@ino06 BigStack]$ mv temp.mon run0076.mon
```

Now, the analysis macro is invoked by the following command:

```
[lamps@ino06 BigStack]$ root -l BigStackV1.C
```

The program then prompts for a number of inputs. Following is a typical snapshot of this session:

```
root [0]
Processing BigStackV1.C...

RPC identification number      [ex: IB01]: IB01
First strip number  (Layer 0)    [ex: TP]: TP
Second strip number (Layer 1)     [ex: 4]: 4
Third strip number  (Layer 2)     [ex: 3]: 3
Fourth strip number (Layer 3)     [ex: 5]: 5
High voltage to the RPC in KV   [ex: 9.6]: 9.8
Gas mixture used      [ex: 95.5:4.5:0.11]: 95.5:4.5:0.11
RPC test run number            [ex: 0002]: 0076
Data type (Event/Monitor)         [ex: e]: e
Start event number to analyse     [ex: 1]: 1
End   event number to analyse   [0 = End]: 0
```

A few points to note here:

1. Default inputs for every query are shown in [brackets]. If your press Enter at a prompt without typing an input, the default is taken.
2. Right now, the strip readout by Layer 1 is assumed to be the main strip. `TP` stands for test pulse input that we feed to Layer 0 for reference.
3. Long inputs such as gas mixture etc. may be entered by cutting and pasting the value shown as the example.
4. Event and monitor data are collected simultaneously by the *InoC11* program, but *BigStackV1* analyses the same separately. Hence the input for *Data type*.
5. Normally you will enter `1` and `0` for start and end event numbers to analyse. These inputs are however more useful, if you want analyse a part of the run.

Then, the analysis will start. If no debugging switches are not turned ON, then the typical output on the screen looks like the following in case of event data analysis:

```
  .
  .
  .
  3300 events processed so far ...
  3400 events processed so far ...
  3500 events processed so far ...

RPC test run number             : 0076
Total number of events processed: 3517
Run summary stored in           : eve0076.sum
ROOT histograms stored in       : eve0076.root
TDC plots stored in             : eve0076.ps

root [1] .
```

The macro will generate a text summary file (.sum) giving salient features of the data, a ROOT compatible file (.root) consisting of various histograms as well as a Postscript file (.ps) containing various plots of general interest. Switches for turning on or off creation of these files are supported. The Postscript file will contain fitted plots of QDC and TDC distributions, QDC-versus-QDC scatter plots, QDC-versus-TDC scatter plots, run time and trigger rate scatter plots.

Please look at a sample PS file *sample-event.pdf*

Following is a typical summary file for the event data analysis:

```
                         RPC test run summary (Event)
                         ----------------------------

          RPC under test              : IB01
          Gas mixture used            : 95.5:4.5:0.11
          High voltage (KV)           : 9.8
          Test run number             : 0076
          Total events                : 3517
          Date and time of first event: 07:07:25 16:11:52
          Date and time of  last event: 07:07:26 09:46:47
          Run time (dddd hh:mm:ss)    : 0000 17:34:55
          Trigger rate (per minute)   :   3.3


          Strip-wise statistics ...

          Strip numbers               :      TP       4       3       5
          Strip hits                  :       0    3075     245     657
          No strip hits               :    3517     442    3272    2860
          Strip efficiencies(%)       :     0.0    87.4     7.0    18.7


          Strip-wise folds & coincidences ...

          No strips hit               :     333 [ 9.5]
          Strips  4 +  3              :    3120 [88.7]
          Strips  4 +  5              :    3143 [89.4]
          Strips  3 +  5              :     836 [23.8]
          Strips  4 +  3 +  5         :    3184 [90.5]
          Strips  4 x  3              :     200 [ 5.7]
          Strips  4 x  5              :     589 [16.7]
          Strips  3 x  5              :      66 [ 1.9]
          Strips  4 x  3 x  5         :      62 [ 1.8]
```

```
Strip-wise timing parameters ...

Strip numbers               :      TP       4       3       5
Mean signal times  (nS)     :  407.30   98.32  406.88  407.32
Timing resolutions (nS)     :    0.15    1.84    0.16    0.17

Strip-wise charge parameters ...

Strip numbers               :      TP       4       3       5
Mean signal charges (pC)    :    6.37   48.09   26.02   23.01
Charge resolutions (pC)     :    0.00   16.85    0.74    1.80
```

If you want to quit the analysis session at this stage, you can give the following command at the ROOT prompt:

```
root [1] .q
```

But, most likely you also want to do monitor data analysis. In that case, instead of quitting the analysis session and doing it all over again, you can type the following command at the ROOT prompt and continue with monitor data analysis

```
root [1] .x BigStackV1.C
```

In case of analysing the monitor data, the inputs asked by BigStackV1 will be similar, except that you will enter a `m` for the *Data type* query and then the last three entries will read like:

```
  Data type (Event/Monitor)          [ex: e]: m
  Start sample number to analyse     [Ex: 1]: 1
  End   sample number to analyse  [0 = End]: 0
```

Following is a typical output on the screen while analysing monitor data:

```
   .
   .
   .
   400 samples processed so far ...
   500 samples processed so far ...
   600 samples processed so far ...

RPC test run number              : 0076
Total number of samples processed: 631
Run summary stored in            : mon0076.sum
ROOT histograms stored in        : mon0076.root
Monitor plots stored in          : mon0076.ps

root [1]
```

The macro will generate a text summary file (.sum) giving salient features of the data, a ROOT compatible file (.root) of various histograms as well as a Postscript file (.ps) containing various plots of general interest. Switches for turning on or off creation of one or more of these files are supported. The Postscript file will contain fitted plots of noise rate histograms as well as noise rate versus sample number scatter plots.

Please look at a sample PS file *sample-monitor.pdf*

Again, following is a typical summary file for the monitor data analysis:

```
                    RPC test run summary (Monitor)
                    ------------------------------

RPC under test              : IB01
Gas mixture used            : 95.5:4.5:0.11
High voltage (KV)           : 9.8
Test run number             : 0076
Total samples               : 632
Date and time of first event: 07:07:25 16:13:16
Date and time of  last event: 07:07:26 09:47:20


Strip-wise noise rate parameters ...

Strip numbers               :      TP       4       3       5
Mean noise rate             :15699.40 6494.14 4820.16 3833.94
Standard deviation          :   70.99  286.09  227.61  118.28
```

You can enter any Linux command from ROOT prompt by using the following syntax:

```
root [1] .! <Linux command>
```

For example, if you want to have a quick glance on event data analysis summary sheet for run number 76, then you will issue the following command on ROOT prompt:

```
root [1] .! more eve0076.sum
```

You can have a detailed look at the plots and distributions by entering an interactive ROOT session by typing `Tbrowser b;`on the ROOT prompt. This command will open a ROOT Object Browser. You will select the .root file that the macro has just created and possibly study in detail various plots and distributions, zoom them, fit them etc. You can also view file of any type, just by double clicking on the filename. ROOT will open the file using the appropriate application program.

Please consult a local help or ROOT's home page (http://root.cern.ch) on how to go about doing these things.

Finally, please send bug reports and your suggestions both on the macro as well as on this document to me at bsn@tifr.res.in.

B.Satyanarayana
11th August 2007