

## Section 1 Essential elements to start “DAQ-1” IGUI

A lot of ENVIRONMENT variables need to be defined in order to start “DAQ-1”.

- A** Examples of bash scripts can be found in “maia” in /root/TestStand/daq17/.rod.sh and /root/TestStand/daq17/rod.sh. One is supposed to do

```
source rod.sh          or
```

```
. rod.sh
```

before trying to start “DAQ-1”.

- B** Examples of csh/tcsh scripts can be found in maia in /home/kinyip/.rod.csh and /home/kinyip/daq17/rod.csh. One is supposed to do

```
source rod.csh
```

before trying to start “DAQ-1”.

- C** For a non-superuser ( $\$USER \neq \text{“root”}$ ), the \$HOME directory should be the same in both local host(“maia”) and remote host (“bert”). The essential definitions that the remote host absolutely needs (in order for “pmg\_agent” to run) is called at login (such as my “.rod.csh”). The rest is put in “rod.csh”. Though one can in principle define everything at login, it is better not to call everything at login.

- D** However, for the superuser “root”, the disk/directory where \$HOME sits is different between local and remote hosts. In local host (maia), we do not need to define anything at login and so in principle all the shell definitions can be put in rod.sh. I still separate them into two parts for clarity though rod.sh actually calls .rod.sh. In the remote host (“bert”), only .rod.sh is needed and I actually just put them in /root/.bashrc.

To start “DAQ-1” IGUI, do

```
play_daq          your_partition_name
```

where your\_partition\_name can be “PartitionRod1” for our ROD setup.

## Section 2 Databases

- A** Partition database files (such as PartitionRod1.data.xml and Rod.data.xml) are in the directory \$TDAQ\_DB\_PATH/data-flow/partitions. (A few more, which we do not use at the moment, can also be found in \$TDAQ\_DB\_PATH/online/partitions.
- B** To edit databases, one can
- i.** confdb\_edit\_data.sh or
  - ii.** confdb\_gui -p path/your\_partition\_xml\_filename or
  - iii.** simply text edit the corresponding .xml files if it is just a simple name change etc.
- C** When saving database files after editing using confd\_edit.sh, one has to save with the option “extended xml” so that the database file are sort of text readable.

## Section 3 “pmg\_agent” running in a remote CPU board

- A** “DAQ-1” needs to start a “pmg\_agent” in the remote CPU board such as “bert”. This is possible only if the remote CPU board “bert” is able to get access to the same “DAQ-1” installation as is in the local machine.
- B** To Start start “pmg\_agent” in the remote machine, either rsh or ssh has to be invoked. One can define “TDAQ\_RSHELL\_CMD” in local host (“maia”) to simply “ssh” or “/usr/bin/ssh” to use ssh instead of rsh.
- i.** To use rsh, one has to uncomment the following line /etc/inetd.conf to allow rsh:  

```
shell stream tcp nowait root /usr/sbin/tcpd in.rshd -hL
```

where -h is to allow “root” to use this mode, too.
  - ii.** To use ssh, one has to do the usual steps for each shell, ie. :
    - a) eval `ssh-agent`
    - b) ssh-add your\_key

where your\_key may be, for example, ~kinyip/.ssh/id\_dsa. Here, I have assumed that the keys have been properly generated by ssh-keygen and put in the appropriate locations.

- C** “pmg\_agent” (among a couple other process) is not killed even when we exit “DAQ-1”. It seems that when “DAQ-1” restarts, it would check whether “pmg\_agent” is running or not. If it has already been running, it would not try to start it again but just use the existing pmg\_agent. This is true even for the remote board. So, even when ssh-agent is not working, if pmg\_agent is running in the remote machine, “DAQ-1” would still work.
- D** /tmp/pmg\_agent\_XXXX.out in both local and remote hosts are useful log files to look at.
- E** Sufficient ENVIRONMENT variables have to be defined in .bashrc or other shell initialization script in the remote host. In particular, one has to define \$TDAQ\_LIB\_PATH so that \$LD\_LIBRARY\_PATH would be defined accordingly for the remote pmg\_agent process. Otherwise, \$LD\_LIBRARY\_PATH for the remote pmg\_agent would be pointed to certain /afs/cern.ch/atlas/project/tdaq area, and then the user controller process (eg. rc\_larg\_rod\_ctrl) cannot be started because the necessary share libraries such as librc1.so cannot be found.
- F** **Caution:** pmg\_agent should be stopped and started again if you want the ENVIRONMENT variable changes to be effective!

#### Section 4 Using Java to change IGUI interface

- A** Unfortunately, one has to write JAVA codes to change the IGUI panel. The ~kinyip/daq17/comjava is a script which will compile java and put them in the right format.
- B** In particular, when you want to have different JAVA components under different trees. You need to have the appropriate .class files (which appeared after compiling) in the appropriate directory before using “jar uf” to update the respective components. Components placed in the wrong tree would not work.
- C** The ENVIRONMENT variable **CLASSPATH** needs to be defined. Especially, when you want to use your own java codes in the current directory, you need to include “.” (current directory) in the path. And you have to include \$IGUI\_CLASS\_PATH to CLASSPATH to make use most of the existing java objects and functionalities.

#### Section 5 Online Bookkeeping customization

- A** In /usr/local/apache/htdocs/obk\_dir/configdb.php, it assumes “localhost” for the database server (ie. atlassw1.phy.bnl.gov ). But atlassw1.phy.bnl.gov, since 18-August-2002, has become a backup database server only and one cannot write to it directly. The new/current database server is db1.usatlas.bnl.gov. I have to put in the following line in configdb.php :

```
if ($host == ") $host = 'db1.usatlas.bnl.gov';
```

so that the web server would go the above database to get access to the information.

- B** Due to my complaints and requests, Mário Monteiro has added a lot of features since the first edition sent to us. Now, the OBK browser can work no matter whether register\_globals (set, for example, in /etc/php.ini) is on or off. That is, the OBK browser can be run in both servers of atlassw1.phy.bnl.gov or larg-fest.phy.bnl.gov. The privilege control over the users is also improved or corrected. I have set the user “admin” to have all the dangerous privileges (eg. to delete a run entirely or to delete files uploaded by any users) whereas the other users do not. Any users can add comment and upload files with ease but they cannot delete (unless the “admin” user change the privileges for the user in question).

## Section 6 Online monitoring

- A** In principle, there is an event sampler function in “DAQ-1”. But it seems that it is useful for more complicated cases such that there are different sub-detectors and one can pick up different event fragment from each detector etc. For us, we do not really need it and it is probably simpler if we read the data file directly.
- B** Up to this version, the name of the Application should contain **only letters and numbers** can be used for the name of the “process”, otherwise the system would not be able to recognize it (a bug which may be fixed in the next version!). It seems also safe to set the “Name” and “ObjectID” to be the same (it is not necessary in all the places but it is safer to do it this way).
- C** In the monitoring process, histograms are published for our partition “PartitionRod1” in the Online Histogram server “Histogramming”.
- D** “Root” is **not** really integrated into “DAQ-1” yet and one cannot click a button to show “Root” histograms. One has to call an external root program to do so. For that, we have the root\_display.cxx program which “receive” all the histograms from the appropriate partition, server etc.

The problem here is that, in this version of DAQ-1, one **cannot erase previously published histograms**; and unless you restart the server, you have to deal with all the histograms. To get around it, I iterate the OHHistogramIterator to the end (they do not provide the list/vector for the histogram either) and go back to “receive” the only last n ones (at the moment n=1). One has to change the number n in the future to receive the right number of histograms at the end. It is said (again) that in the next version, they would allow the users to erase/clean up previously published histograms.

Now, to show the histograms that have been published to partition “PartitionRod1” in the server “Histogramming”, do

```
/public/root_display/root_display -p $TDAQ_PARTITION -s Histogramming -a 1 -g
```

in which \$TDAQ\_PARTITION is hopefully defined already as the ENV variable for “PartitionRod1”; “-a 1” indicates to look at the last 1 histogram; and “-g” is to show is in graphic mode.

**E** The “EventDump” can only be used to dump events with a standard data format as described in the old note <http://atddoc.cern.ch/Atlas/Notes/050/Note050-1.html> or <http://atddoc.cern.ch/Atlas/postscript/Note050.ps>, which (I think) is superseded by David Francis’ Atlas note published in mid 2002. It is said that they may add the capability to dump events with user-defined data format in a later version.

## Section 7 Miscellaneous

**A** The command “is\_monitor” can be used to look into the IS information. This has helped me find out one particular detail.

**i.** In order to update the “Event number” in the IGUI, one has to define a “Parameter” in the database. The “Object ID” of this parameter is “EventInfoSource” and so is the “Name”. The problem is the appropriate value.

**ii.** Some ISInfo parameters have already been published and it’s good that we make use of those parameters instead of creating another variable which is the same.

**iii.** I started “is\_monitor”, chose “PartitionRod1” partition and then clicked on DF and clicked “show info list”. I saw “DF.RODC1.ROD1” appeared under the “Name” item.

iv. This is the right “value” for the parameter “EventInfoSource”. The exact “value” should be something like RODC1.ROD1:2. “2” means the 3<sup>rd</sup> variable defined for the RODC1.ROD1 module. Here the 3<sup>rd</sup> variable is the number of events read in.

**B** To start a 2<sup>nd</sup> IGUI, you have to do “igui\_start” and choose a partition. Or, simply “igui\_start -p PartitionRod1”, where “PartitionRod1” is the partition.

**C** In the database, the “CreationType” for a certain application controls WHEN a process linked to a particular partition would be started.

i. **Supervised:** the process would be started at “Boot” time and stopped at “Shutdown”. If such an application dies, the DAQ-1 system enters a “Fault” state.

ii. **SOR/EOR:** the process would be started at the “Start Of Run”/ “End Of Run”.

The problem here is that there is no automatic stop mechanism (in the present version of the DAQ-1). One solution is that one may subscribe to the MRS (Message Reporting System) for the “RC\_END” message with a “callback” function to stop the process upon receiving “RC\_END” message.

iii. **Default:** the process would be started when you start the process (ie. clicking the “start” button at the appropriate IGUI page such as “Monitoring”) and stopped when you stop the monitoring process (ie. clicking the “stop” button).

In all these options, one can always stop the process under certain condition such as a maximum number of events is surpassed.

## Section 8 Helpful experts

**A** General:

i. Mihai Caprini ([Mihai.Caprini@cern.ch](mailto:Mihai.Caprini@cern.ch))

ii. Beniamino Di Girolamo ([Beniamino.Di.Girolamo@cern.ch](mailto:Beniamino.Di.Girolamo@cern.ch))

**B** Monitoring and online histogramming:

i. Kolos Serguei ([Serguei.Kolos@cern.ch](mailto:Serguei.Kolos@cern.ch))

**C** Online Bookkeeping:

i. Levi Lucio ([Levi.Lucio@cern.ch](mailto:Levi.Lucio@cern.ch))

ii. Mário Monteiro ([mariomonteiro@softcontrolweb.com](mailto:mariomonteiro@softcontrolweb.com))

**D**