READ-QRF

The Baikal Ntuple Reader for Monitor Quality Files
User's guide, Draft 1, December, 2000

Ralf Wischnewski, Serge Klimushin

1 Description of the Ntuple Variables

Her we give a brief description of the meaning of the ntuple variables that are generated by the program read˙qrf.

QRF-NTUPLE (identifier = 100) contain the following variables for each run that enters the ntuple as an "entry".

```
Type   Name          Meaning

 R*4  * run          run-number
 R*4  * year         year of run start (i.e. 1999)
 R*4  * day          day of run start  (days since beginning of current year:
                       i.e. 10-Feb == 41 days)
 R*4  * sec          seconds of run start ( since 0:0:0 of current day)
 R*4  * dt           run duration in secs ("Laser" runs have dt < 5000)

 Muon Statistics Block.....
 I*4  * nevtot       total number (statistics) of events in run
 I*4  * nevgood      total number of "good" events (completely error free)
 I*4  * nevbad       total number of "bad" events
                     (with any error from error capture (see below))
                     nevtot = nevgood + nevbad
 I*4  * nchdead      number of out-of-order (dead) channels (with nhits == 0)
 I*4  * nchqsdead    number of channels with abnormally small nhits (% < cutqsdead)
 R*4  * cutqsdead    cut-off percentage value for nchqsdead (usually set to 1%)
 R*4  * arrmul(97)   array multiplicity
                     (arrmul(i) is the number of events with number of
                     hit channels >= i)

 LTS Block.....
 I*4  * iclts(96)    LTS-channel index (LTS interval is set to 0.8 sec)
 I*4  * nevlts(96)   number of LTS-events the channel # i took part in
 R*4  * avlts(96)    channel # i average LTS-rate (Hz)
 R*4  * sglts(96)    channel # i LTS-rate sigma (Hz)

 Muon Block.....
   ... Hit rates
 I*4  * icmu(96)     channel index
 I*4  * nhmu(96)     channel abundance (nhits for channel # i)
 R*4  * avamp(96)    channel # i average amplitude (A-code with pedestal)
 R*4  * sgamp(96)    channel # i amplitude sigma (A-code)

   ... Time Differences
 I*4  * ixdt(96)     channel time difference combination index (see remark.1)
 I*4  * nhdt(96)     statistics for given combination ixdt
 R*4  * avdt(96)     average time difference for given combination ixdt in nsec
                     (time=code*betta, where bettas values are for ethalon run)
 R*4  * sgdt(96)     sigma of time difference for given combination ixdt (nsec)
```

```
 Muon Error Statistics for events and channels ...
 I*4  * ixneverr(32)  index of error-event type   (trivial: ixneverr(i)=i)
 I*4  * ixncherr(32)  index of error-channel type (trivial: ixncherr(i)=i)
 I*4  * neverr(32)    number of error-events of type ixneverr (1..32)
 I*4  * ncherr(32)    number of error-channels of type ixncherr (1..32)

... Information related to ordering of raw channel-wise information
    before clusterization.
    Considered is the event number difference between successive
    arriving at shore channels: D_ev=N_ev(k+1) - N_ev(k),
    where N_ev(i) (=0..63) is the event number associated with k_th channel
    arriving at shore.
    Big differences D_ev point to "Misordering" of the raw data.
    During a given run there's nevdiffall=n_channels_per_event*N_events
    channels arriving.
 I*4  * nevdiffall      number of channels that were considered (i.e. arrived)
 I*4  * nc1diffall      number of chans between D=c1diffall and 63-c1diffall.
 I*4  * nc2diffall      number of chans between D=c2diffall and 63-c2diffall.
 I*4  * c1diffall       the value of cut1 used above
 I*4  * c2diffall       the value of cut2 used above
 I*4  * ixevdiffall(64) the bins for D_ev distribution (trivial: ixevdiffall(i)=i)
 I*4  * evdiffall(64)   the full distribution of D_ev
                        Good runs have only entries near bins 1 and 64 !!!
                        If the middle is populated, be careful.

 Error capture ...
  ... for Events.
 ixneverr    meaning
   2         lost World time
   4         naked World time
   5         wrong BPI number
   8         wrong channel number
   9         double channel event                   !!!
   10        lost event number                      !!
   11        stop sorting (corrupted clusterization) !!!!!!!!
   13        out-of-trigger (hit number < 4)        !!!!!!!
   All others are unused and should be =0.

  ... for Channels.
 ixncherr    meaning
   - to be added -
```

```
Remarks:
--------

1. Only 6 combination of channel time difference ixdt are available for each
   half_string # i (i=1..16):
       i12 --> t1-t2, i34 --> t3-t4, i56, i24, i26, i46;
   where tj is time of channel # j (j=1..6) at half_string # i.
   Example for ixdt:
             1426 - means (t2-t6) for half_string # 14,
              256 - means (t5-t6) for half_string # 2.

   So: nt/pl 100.avdt%run dt>5000.and.ixdt=1534 means:
   "plot distribution of the average time difference between channels 3 and 4
    at half_string # 15 vs run # for all non-laser runs".


2. To avoid year changement jump use variable (year-CURRENT_YEAR)*365+day
   instead of "day".
   Example:
   Plot average LTS-rate of channels # 38 and 39 for all non-zero LTS runs
   along two years (1998 and 1999 since 0:0 1-Jan-1998) at the same picture
   null 0 800 0 2000               * prepare coordinate box ( Hz <= 2000 only)
   cdir //lun1                     * go to ntuple of 1998
   nt/pl 100.avlts(38:39)%(year-1998)*365+day  dt>5000.and.avlts(38:39)>=5 -s
   cdir //lun2                     * go to ntuple of 1999
   nt/pl 100.avlts(38:39)%(year-1998)*365+day  dt>5000.and.avlts(38:39)>=5 -s


3. Example for array multiplicity plot:
   nt/pl 100.arrmul(6)/dt%run dt>5000 -
       plots counting rate of master (>=6 hits) vs non-laser run #
   nt/pl 100.(arrmul(int((96-nchdead)*0.5))*100./nevgood)%run dt>5000
       plots percent of events with huge hit multiplicity
       (50 % of operating channels) vs run #

4. Example for error-events:
   nt/pl 1000.(neverr(13)/dt)%(neverr(9)/dt) dt>10. -
       plots rate of out-of-trigger events vs rate of double channel events
       (for all runs with non-zero run-duration dt)

5. Let point out that COLOR marker helps to investigate many-plot picture.
   To change color of marker use simply PAW-command
       set pmci icol     (where icol is color index : 1 - black, 2 - red ...)
   before plot via  nt/pl 100...
```