How to run JTriggerEfficiency (1/2)

login on interactive machine in CCLyon:

1. ssh –Y <user name>@cca.in2p3.fr
2. source $KM3NET_THRONG_DIR/src/Jpp/trunk/setenv.csh \
   $KM3NET_THRONG_DIR/src/Jpp/trunk/
3. mkdir –p /sps/km3net/users/<user name>/tmp/
4. cd /sps/km3net/users/<user name>/tmp/
5. JSirene.sh (if not already done)
6. JTriggerEfficiency.sh
   wait 5 minutes to process more than 10,000 events

How to run JTriggerEfficiency (2/2)

next steps:

6. JPrintMeta –f trigger_efficiency+background.root
   will print meta data
7. JTriggerMonitor –f trigger_efficiency+background.root
   will print trigger statistics

final step:

8. rm –rf /sps/km3net/users/<user name>/tmp/

Documentation

- need immediate help?
  - JTriggerEfficiency –h! will print all command line options with their default values
- trouble with syntax?
  - auxiliary script JTriggerEfficiency.sh simplifies syntax
- want to know more?
  - browse to www.km3net.org

    INTERNAL jenkins.km3net.de Jpp Jpp_trunk DogyGen HTML

  - Tip: bookmark this page / add this page to your favourites

JTriggerEfficiency options (1/3)

- JTriggerEfficiency –h

  Auxiliary program to trigger Monte Carlo events.

  usage: JTriggerEfficiency

  -h <help>
  -@ <parameters>
  -B <rates_Hz>
  -O <triggeredEventsOnly>
  -P <pmtSimulator>
  -R <run>
  -S <seed>
  -a <detectorFileA>
  -b <detectorFileB>
  -d <debug>
  -f <inputFile>
  -n <numberOfEvents>
  -o <outputFile>
  -r <runbyrun>

JTriggerEfficiency options (2/3)

- detectorFileA
  - detector geometry and calibration file (e.g. .det or .detx)
    - to convert Monte Carlo truth to raw data
- detectorFileB
  - detector geometry and calibration file (e.g. .det or .detx)
    - to convert raw data to calibrated data
    - by default, but not necessarily, same as detectorFileA
- inputFile
  - file produced by e.g. km3, JSirene, etc. (either ROOT or ASCII formatted)
- outputFile
  - ROOT formatted DAQ file

JTriggerEfficiency options (3/3)

- parameters
  - “(<key> = <value>;)+” list of trigger parameters
  - file name with list of trigger parameters
    - e.g. $JPP_DATA/ trigger_parameters_arca.txt
- triggeredEventsOnly
  - write only triggered events (recommended), else write all events
- pmtSimulator
  - “(<key> = <value>;)+” list of PMT parameters
  - file name with list of PMT parameters
    - e.g. $JPP_DATA/ PMT_parameters.txt
- seed
  - seed for random number generation (TRandom)

JTriggerEfficiency.sh options (1/1)

- JTriggerEfficiency.sh –h

  Setting environment variables for Jpp software.

  JTriggerEfficiency.sh [detector file [(input file)+ [output file [trigger file [PMT parameters file]]]]]

  Note that if more than one input file is specified, all other arguments must be provided.

- example

  JTriggerEfficiency.sh <detector file> <input file> <output file> <trigger file>

What you should know (1/3)

- JTriggerEfficiency is an application that applies the same trigger software to simulated events as JDataFilter to real data
  - Monte Carlo true information is converted to a time slice with DAQ hits
- JTriggerEfficiency is a place holder for all detector related effects
  - PMT inefficiencies, time slewing, discriminator thresholds, etc.
- JTriggerEfficiency is designed to be flexible
  - accepts many input parameters (e.g. several parameters per PMT)
  - can operate in run-by-run mode (i.e. uses summary data taken with real detector)

What you should know (2/3)

- There are many trigger parameters
  - see e.g. JPrint –f <file name>
  - parameters can be set on the command line using option -@
    - -@ <file name>
    - -@ “<key>[.<key>] = value; …”
  - main parameters for each trigger
    - trigger3DShower.enabled =0(1 to enable)
      trigger3DShower.numberOfHits =5
    - triggerMXShower.enabled =0(1 to enable)
      triggerMXShower.numberOfHits =8
      triggerMXShower.numberOfModules =2
    - trigger3DMuon.enabled =0(1 to enable)
      trigger3DMuon.numberOfHits =5

What you should know (3/3)

- Other parameters
  - TMaxLocal_ns = 10// local time window for L1 hits
  - combineL1 = 1// combine multiple L0 hits within time window

- Other outputs
  - writeSummary = 1// pre-scaler: 0 = off; 1 = on; 2 => ½; etc.
  - writeL0 = 0// L0 hits
  - writeL1 = 0// L1 hits (e.g. for K40 analysis)
  - writeL2 = 0// L2 hits (primarily for testing)
  - writeSN = 0// same as L2 but separate stream for Supernova