

How to run JSirene (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`
wait less than 10 minutes to get more than 10,000 events

How to run JSirene (2/2)

next steps:

6. `JPrintMeta -f sirene.root`
will print meta data
7. `JPrintSirene -f sirene.root`
will print JSirene statistics
8. `JDomino.sh - sirene.root`
will produce a set of standard plots

final step:

9. `rm -rf /sps/km3net/users/<user name>/tmp/`

Documentation

- need immediate help?
 - JSirene -h! will print all command line options with their default values
- trouble with syntax?
 - auxiliary script JSirene.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

JSirene options (1/3)

- JSirene -h!

Main program to simulate detector response to muons and showers.

usage: JSirene

-h <help>

-F <fileDescriptor>

-N <numberOfHits> [1]

-S <seed> [0]

-T <Tmax_ns> [0]

-a <detectorFile> []

-d <debug> [1]

-f <inputFile> []

-n <numberOfEvents> [0 9223372036854775807]

-o <outputFile> [sirene.root]

-s <writeEMShowers> [0]

JSirene options (2/3)

- **detectorFile**
 - detector geometry and calibration file (e.g. .det or .detx)
- **inputFile**
 - file produced by e.g. genhen, MUPAGE, etc. (usually ASCII formatted '.evt')
- **outputFile**
 - ROOT formatted AAnet file
- **fileDescriptor**
 - typical value is <path>/l%p.dat
 - JSirene replaces wild card '%' by the corresponding values of the PDFs; for more information, type JPDFType_t in Doxygen search box

JSirene options (3/3)

- writeEMShowers
 - option to add Bremsstrahlung showers to track list
- numberOfHits
 - minimum number of hits to output event
- seed
 - seed for random number generation (TRandom)

JSirene.sh options (1/1)

- JSirene.sh -h

Setting environment variables for Jpp software.

JSirene.sh [detector file [input file [output file [CDF file descriptor]]]]

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

- example

JSirene.sh <detector file> <input file> <output file>

What you should know (1/2)

- JSirene is an application that can be used to simulate the detector response to muons and showers
- JSirene uses general purpose probability density functions (PDFs) for the arrival time of Cherenkov light from muons and showers
 - represent *de facto* our 'best' knowledge of detection principle
 - facilitates QA/QC
- JSirene is designed for speed
 - development ☐ fast turn-around-time
 - mass productions ☐ can run in real-time with less than 4 CPU cores
 - systematic studies ☐ more results per unit time

What you should know (2/2)

- PDFs can be calculated based on a formalism
 - script JMakePDF.sh can be used to create PDFs
 - JMakePDF.sh -W <working directory> -P launches many jobs to produce PDFs in parallel
 - JMakePDF.sh -W <working directory> -C converts PDFs to CDFs
- PDFs can be created from histograms of Monte Carlo data
 - applications JHistPDF and JHistPDG produce histograms from the output of JSirene, km3, etc.
 - applications JMakePDF and JMakePDG convert these histograms to PDFs
- PDFs can be plotted, compared, etc.
 - applications JDrawPDF and JDrawPDG plot the calculated PDFs
 - applications JPlotPDF and JPlotPDG plot the tabulated PDFs
 - applications JDiffPDF and JDiffPDG compare tabulated PDFs