

DEAP-3600 – monitoring, DQ etc

Ben Smith
TRIUMF

Midas Workshop – 26 July 2017

Outline

- DEAP-3600
- Remote shifting
- End-of-run data quality checks
- Other things

* Generally showing things we've built on top of Midas, rather than Midas things specifically

DEAP-3600

- Dark matter, SNOLAB, liquid argon, 255 PMTs
- Runs ~ continuously (1 year done, 2 to go)
 - 23 hours of physics data
 - 1 hour of calibrations
- Operated remotely unless hardware breaks
 - Shifts are 24 hours long

24 hour shifts!?

- Generally < 1 hour of actually looking at the DAQ
- We send the operator a notification if something goes wrong
 - A run stopped (if the operator didn't ask for it)
 - A custom start/stop webpage sets an ODB value when the operator clicks "Stop run"
 - A custom program (/Programs/Execute on stop run) checks if that ODB value was recently set
 - A Midas alarm is raised
 - Alarms classes are set to call a script (/Alarms/Classes/<class>/Execute command)

Notification channels

- SMS
 - We use messagebird (<https://www.messagebird.com/en-ca>)
 - We pay \$1/month so our messages come from a unique number
 - Cell operators often block the free/shared numbers as they are also used for spam
 - Messages themselves are a few cents
- Slack (collaboration tool, <https://slack.com>)
 - Has a JSON API
 - Sent to #daq channel and current operator
- Email
 - Use Python's smtplib

Automated monitoring

- We raise Midas alarms if we see abnormalities in the data we're taking
 - Sometimes stop the run
 - Sometimes stop the run and turn off the PMTs
- We use a rootana-based analyzer that reads events from SYSTEM buffer
 - Sets values in ODB
 - Midas alarms are set to check those ODB values

End-of-run data quality

Stopping a run (video)

Start/stop run

[Main DAQ page](#)

Current run

Run 19117: Running [Stop run](#) [TEST - show DQ plots](#)

High voltage status: ON

Next run

Run number

19118

Run type [Help](#)

456 - Physics trigger 1000ADC in 8bins, betas 6k-70k prescale 100 - DNF - SQT F

[Show historical settings too](#)

Write Data

Yes No

Comment

Physics trigger at 1000ADC in 8 bin, beta prescale factor 100, SQT filtering, VETO se

Operator name

Ashlea

Is a calibration source deployed?

Yes No

Stopping a run (text)

- Go to DEAP's custom "Start/stop run" page
- Click "Stop run"
- Get presented with 1D or 2D plot of data from this run
 - Read the guidance
 - Select whether the data looks good or not
 - Move on to next plot
- Click "Submit" and DQ responses are stored in a CouchDB database

Usage in data quality checks

10

- There are 4 levels of data quality
 - Automated DAQ checks (did any PMTs turn off? etc)
 - End-of-run checks (what you just saw)
 - Post-processing checks
 - Post-analysis checks
- End-of-run checks helps flag issues sooner
- We don't rely on shifters remembering to check plots themselves – they have no option to skip them!

Dataflow

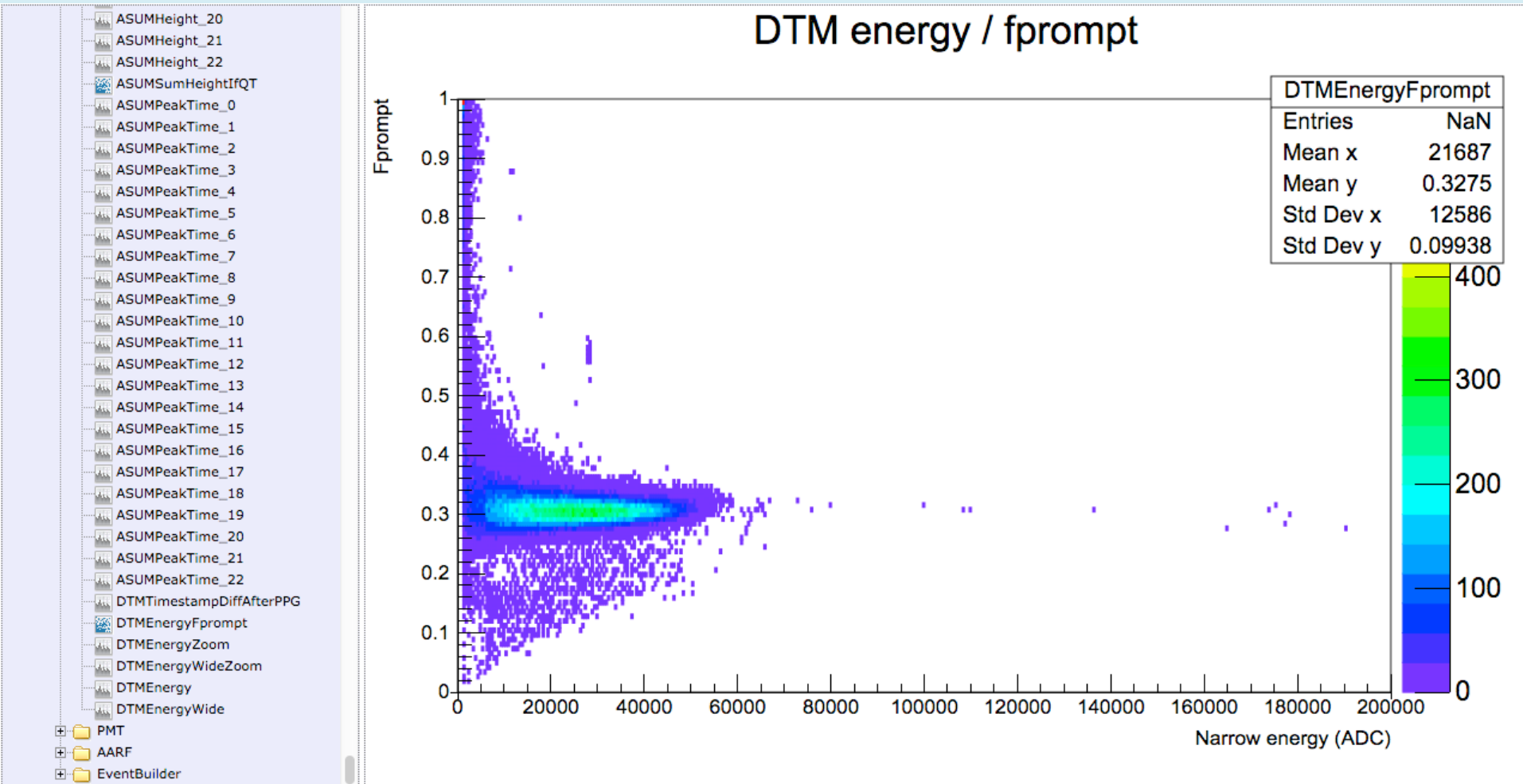
- Events read from Midas SYSTEM buffer
- Analyzed with rootana-based analyser
- ROOT plots written to ROOT file on disk (constantly updated)
- JSROOT exposes ROOT file to web (<https://root.cern.ch/js/>)
- Custom webpage reads plots from JSROOT and displays them

Aside: JSROOT

- JSROOT comes with it's own TBrowser-like display
 - See next slide
- This may be enough for some purposes
- We wanted tighter integration
 - Don't use JSROOT for displaying plots
 - Do query it for list of plots / plot data

JSROOT standard interface 13

(this is a webpage)



Web-based event display

- We have a few plots available through webpage
- It is not very responsive — not great for waveforms, but okay for plots that change less frequently

DEAP DAQ web display

Runtype

Plot

Update (s)

MIDAS Status

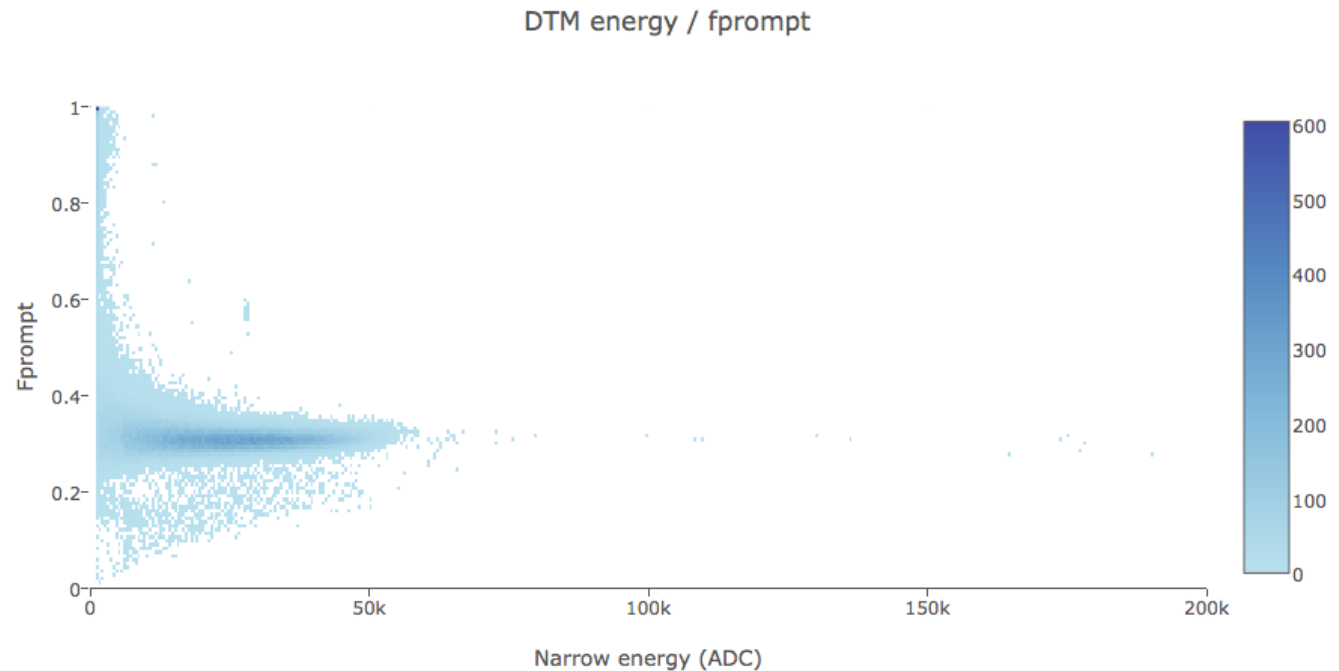
- DAQ is running
- Current run is 20270

Other plots

- [Disk space](#)
- [Physics trigger rates](#)
- [Periodic trigger rates](#)
- [External trigger rates](#)

Links

- [MIDAS Status Page](#)
- [Root Online Browser](#)



Web tools we use

- JSROOT gives data in JSON format
- Plots rendered using
 - dygraphs for 1D graphs (<http://dygraphs.com>)
 - plotly for 2D graphs (<https://plot.ly>)
- Webpages also use
 - jQuery for javascript niceties (<https://jquery.com>)
 - bootstrap for layout/CSS (<https://getbootstrap.com>)
 - CouchDB for database access (run settings, DQ reports etc. <http://couchdb.apache.org>)

Notes and comments

- It is really slow to read JSROOT plots with a lot of bins
 - Web plots have fewer bins than "real" event display plots
- Standard JSROOT display can eat a lot of CPU
- I do not recommend this approach for "live" monitoring of an experiment
- It does work well for occasionally looking at a few plots
- Tying DQ checks to the end-of-run process ensures that checks actually get done

Other midas things you might want to talk about

- A multi-threaded event builder
- Waveform filtering in readout front-end and event builder
- Storing equipment settings in CouchDB for easy setup of different "run types"