GLAST Large Area Telescope: I&T Integration Readiness Review

Online Peer Review
July 21, 2004

User Scripts

Lester Miller
I&T Online
SLAC

Shift Perspective to User

- What the user script sees
- HippoDraw: a visualization and analysis package
- Summary of user scripts
- How the user implements a script
- Running the user scripts:
  - Experiences with and lessons from EM2

What the User Sees

- **LATTE provides:**
  - Framework of state machine
  - Libraries for data parsing, report production, configuring hardware registers, etc.
- **HippoDraw provides:**
  - Visualization (histograms)
  - Analysis tools (fitting, ntuple projection, plotting)
- **User provides a script:**
  - Setup()/StartRun(): hardware set-up (beyond configuration input file)
  - Running(): The Algorithm. How many events with which hardware state (Event collection loop)
  - StopRun(): Analysis of collected data and completion status (Pass/Fail)

LATTE:
Framework
Data parsing
Hardware registers
I/O support
GUIs

userApplication

HippoDraw:
Histogramming
Analysis Tools

HippoDraw: Visualization Package

- HippoDraw is used by the scripts for data analysis
- Capabilities:
  - Real-time plotting of histograms
  - In-memory ntupling with dynamic histogram projection or statically defined histograms
  - "Inspector" GUI for real-time user interaction (rebin histograms, display format, creation of new ntuple projections)
  - Histogram fitting
  - Ntuple and Histogram storage
  - Storing plots as images for output
- Can also be used standalone for data analysis outside of LATTE (e.g. in a multicast consumer for plotting data)
- Supported locally: can add features as we request them

Summary of User Scripts

- Users are:
  - Subsystem deliverers: writing acceptance tests for flight hardware
  - I&T: Scripts for testing interaction of LAT components, inter-subsystem timing requirements
  - Any other online DAQ need which arises
- Online group's role is to support the delivered scripts and write the I&T scripts
- Framework allows quick setup of new event collection algorithms as need arises
- Philosophy: Scripts are to perform both data collection and analysis (no 'offline' component)
- Subsystem scripts: See LAT-TD-02834
  - Subsytems scripts from CAL, TKR, ACD, and Electronics (ELX)
    - Register exercise
    - Pedestals (Noise/Dead/Hot)
    - Gain and linearity
    - Threshold/Charge Inject DAC characterization
  - Most are already functional and preliminary versions are being exercised
  - Will become separate packaged releases

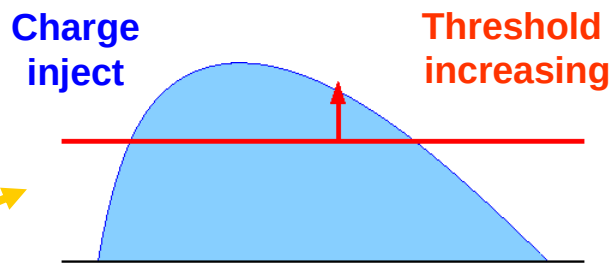
User Script Example: TkrNoiseAndGain

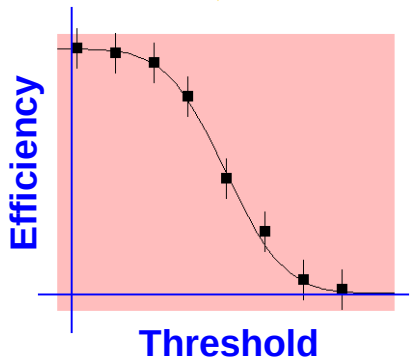
Setup:
Configure tracker
Configure trigger: charge injection

Algorithm:
Charge inject constant amount
Step through increases in
Accept threshold

Analysis:
Fit each efficiency vs. threshold
to error function
Extract a noise and gain from
Fit mean and width

Cleanup:
Produce reports
Decide status





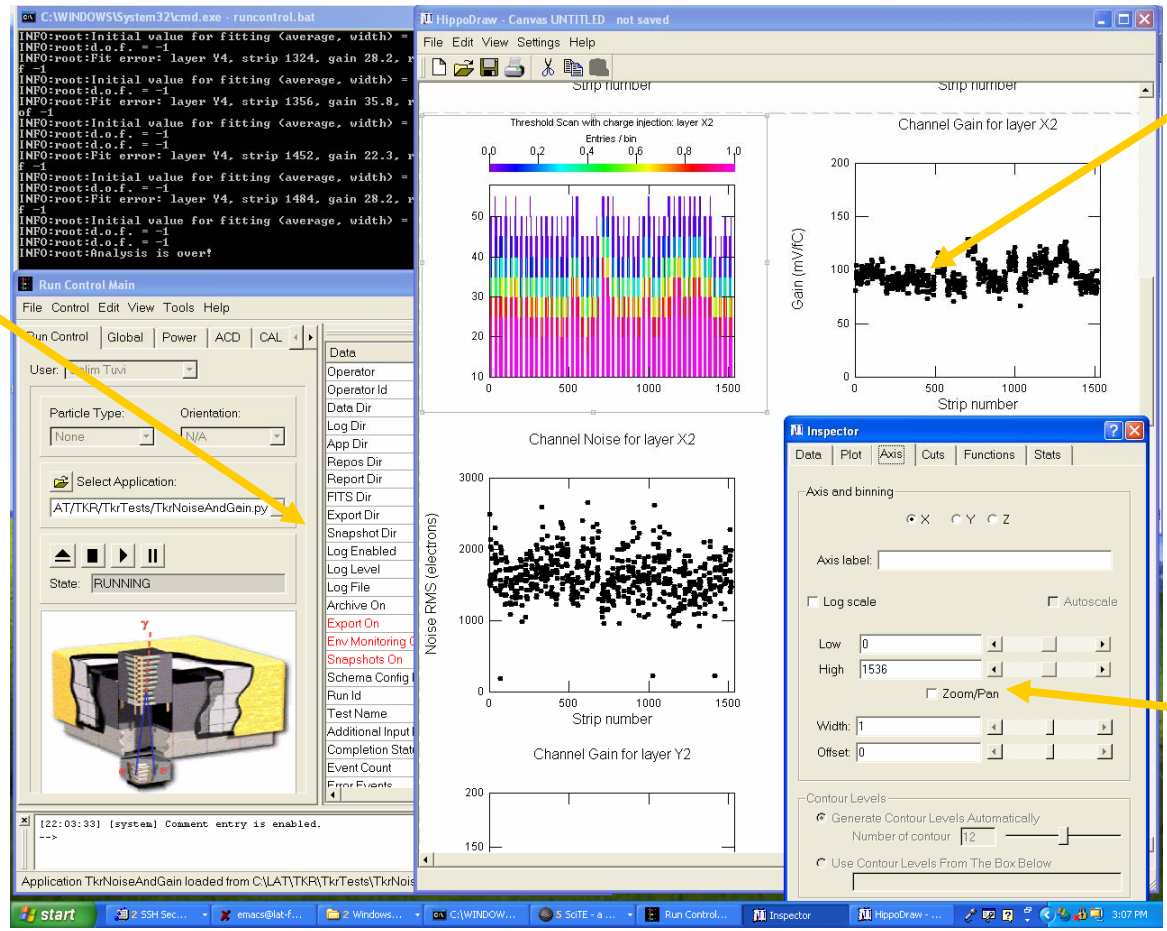
PASS!

User Script Example: Screenshot

This is what it looks like while running in LATTE:

LATTE RunControl GUI

Script Plots



HippoDraw Inspector

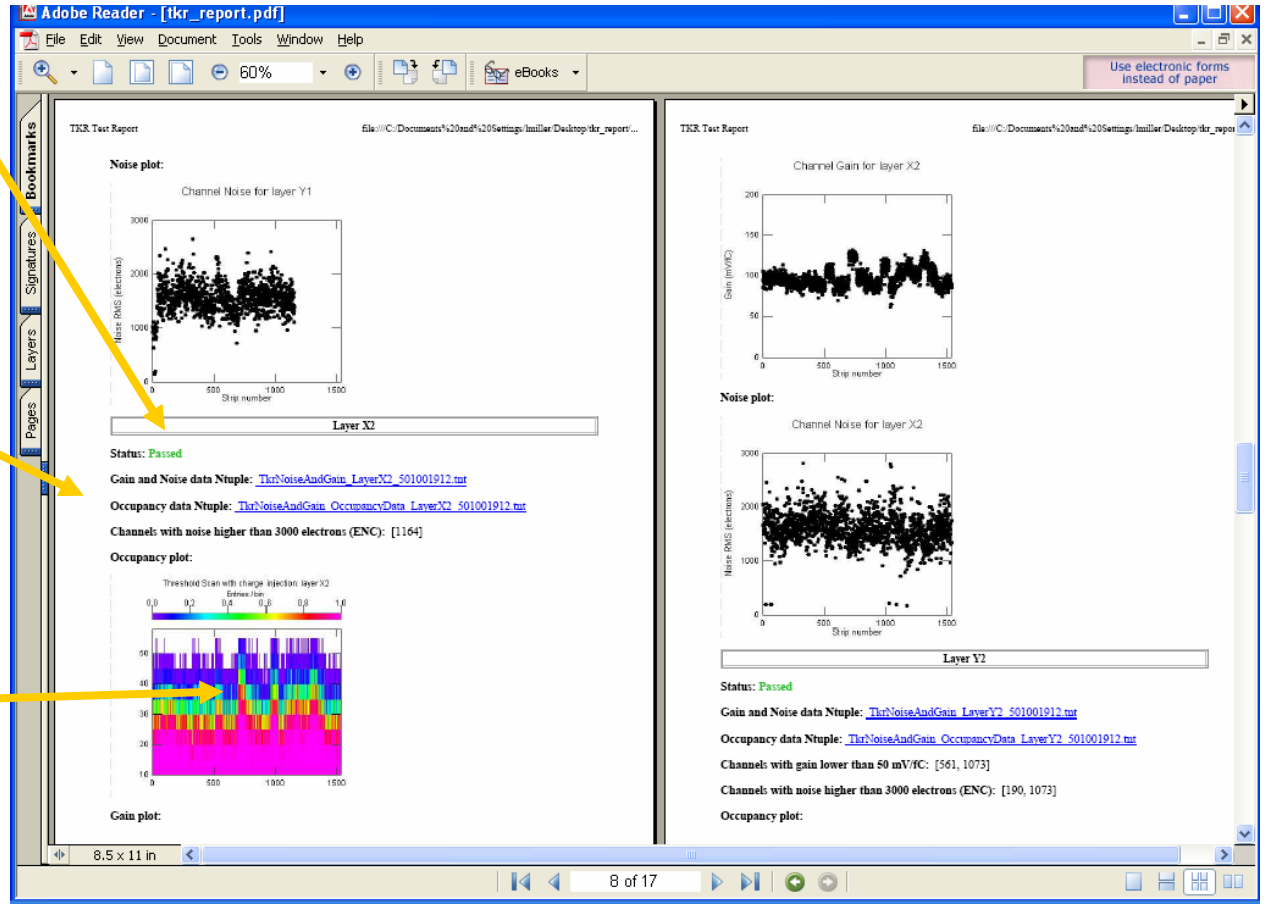
User Script example, cont.

Produces a Report output (html):

Status return

Links to associated files

Embedded plot Picture from HippoDraw



Current Status

- We have run preliminary subsystem scripts with the EM2 mini-Tower:
  - 4 x-y tracker layers
  - 2 calorimeter layers
- We have run with a variety of hardware configurations:
  - mini-Tower in test stand
  - mini-Tower in Test stand with a GASU
    - Different triggering hardware
    - Event data has other contributors: event builder
    - Different data flow (through event builder)
  - mini-Tower attached to test bed
    - Test addressing different tower
    - Different power supply
- We ran with read-only LATTE code (centrally served), will migrate to more ‘realistic’ testing conditions as it is possible

Current Status

- Lessons learned (and being addressed):
  - In running with EM2 we found several assumptions valid only for local installations and/or test stands
    - TEM address hardcoded
    - Assuming raw data has only one contribution (TEM)
  - Scripts are not completely insulated from hardware attached
    - Trigger hardware in test stand and GASU have different behaviour
    - Some extra registers exist in GASU/test-bed which may need configuring
  - Scripts make assumptions about user writeable areas
- Exercising user scripts with more complex hardware has had no show-stoppers. We are ready for multiple tower readout.
- I&T scripts still need to be written (in progress)

And for the future

- **Further possibilities:**
  - **Users can bundle several scripts in a "testSuite"**
    - **Chain together scripts, only proceed if script returns Pass status**
    - **Can simplify acceptance testing for operators**
    - **Framework in place, not tested by any users (yet)**
  - **Scripts can 'replay' data taken to reproduce analyses using a local event server serving from a file**
    - **Generally we will archive all raw data taken**
    - **This has not been fully tested but may prove useful**