

Front-End plans

1. Current system is at: <http://glast-ground.slac.stanford.edu/SystemTests/>
 - Histograms read from Root files via xrootd
 - Summary page displays status of series of "tests", with count of "failed" plots for each test
 - Success/Failure calculated at fill time by KS comparison with reference plot
 - Results stored in oracle "meta-data" table
 - Other statistical comparisons can be computed on fly ("statistics tab")
 - Can plot "time history" (actually version history) of meta-data from database
 - Plots displayed with overlayed reference plots (user can change default reference plots)
 - Tree for browsing individual plots or groups of plots.
 - Zoom in to any plot from page of plots by clicking on it
 - Admin functions allow
 - Plots to be "grouped"
 - Limited control of plotting "style"
 - Hide old code versions
2. Strategy
 - Should keep current system tests as "Software Release Tests"
 - Should keep good features of current system, but be flexible in evaluating different requirements of data quality monitoring
 - New system should bring together as much as possible of monitoring data in a single "web application"
 - Should link in to e-log, housekeeping, activities, quicklook etc.
3. Planned Features
 - Flexible method of choosing time period over which plots are displayed
 - Primary method of selecting plots will be run based, but should also support time period, orbit #, etc
 - Should be possible to integrate plots from multiple runs (may be required for some plots)
 - Probably similar mechanism to current trending application
 - Plots can come from histograms in root files, dynamically created from root tuples or from fits tables
 - Need to test performance for generating plots "on the fly" from tuples
 - Should be able to display tabular data as well as plots
 - Overlay of reference plots, or plots from different run/time ranges
 - Need way to display summary plots, plus drill down to see more details
 - Should support summary pages (html with embedded plots, tables) created by experts (c.f. <http://www.slac.stanford.edu/exp/glast/ground/LATSoft/nfsLinks/u21/Integration/rootData/135002042/v5r0608p6/calib-v1r0/reconReport/v3r2p7/html/index.html>)
 - Flexible method of testing plots for "success/failure"
 - Selection of algorithms for comparing with (expert selected) reference plots
 - Algorithms for testing individual plots (e.g. smoothness)
 - Interactive features
4. Plan for January
 - Ability to display plots from root files on run by run basis
 - No reference plots initially
 - Need
 - histograms (and/or tuples) created by pipeline for each run
 - some database giving location of files (pipeline database may be sufficient)