LAT Software Quality Assurance Strategy

- Combination of Pre- and Post-Release tools

- push-button builds for building releases and for developer tests of performance and cross-platform compatibility

- build released checkout pkg
- run unit, system tests
- record in Database
- compare to standards; report

- build HEAD
- run unit test
- report problems to package owners

- easy controlled installs for users on all platforms

- Nightly Builds

- Document Builder
  - build Doxygen doc
  - available on web

- Release Mananger
- Compile Server
- Release Builds
- Binary Distributions
- Nightly Builds
- Build Doxygen doc
- Available on web
LAT Software Quality Assurance Strategy

• Document Builder

  – Uses Doxygen software combined with SAS documentation policy to automatically generate web documentation for software packages

  – Automatically updates web pages that provide access to package documentation

  – Checks repository nightly (via cron) for new packages and new tags, and performs above operations if found
LAT Software Quality Assurance Strategy

- Compile Server
  - Uses CMT package from LAL to track version dependencies between packages (works on all supported platforms) and implement package definition strategy (i.e. define checkout package, i.e. a package for release)
  - Performs nightly, multi-platform build of head versions (i.e. development versions) of checkout packages and logs results to web pages
  - Performs multi-platform build for individual package at request of developer
  - Performs multi-platform build of checkout package at request of maintainer and logs result to a database
  - Performs multi-platform system and/or unit tests if above build steps succeed
  - Builds binary releases for easy remote installation
LAT Software Quality Assurance Strategy

• Span of LAT ground software tests
  – Unit tests (run by Compile server for nightly and release builds)
  – System Tests (run by Compile server for release builds)
  – Instrument Performance Tests
  – End-to-end tests – “Mock Data Challenges”
LAT Software Quality Assurance

• Unit tests
  – Test individual software packages via test designed by package maintainer
    • Tests have expected outcomes
    • Tests are run by release management software when maintainer “tags” package
    • Tests reside in conventional location (../test subdirectory) and have conventional names (test_PackageName)
    • Failures reported automatically to package maintainer (e.g. via email)
    • Examples:
      – Regression tests, histogram comparison tests
LAT Software Quality Assurance

• System tests
  – Test application (checkout) packages
    • Tests are run by release management software when a release is declared
    • Tests generate diagnostics
    • Diagnostics tracked between releases and compared against standards
    • Failures reported automatically to designated list of management team members
  • Examples:
    – Regression tests, histogram comparison tests, performance tests
LAT Software Quality Assurance

• Instrument Performance tests
  – Test basic instrument performance parameters
    • Show that parameters meet LAT Performance Specification
    • Regular testing and tracking of results will allow for study of code evolution and possible large deviations from understood performance
    • In particular, examine (after background rejection and resolution cuts)
      – TKR front and back section PSF, as a function of energy and angle
      – Energy resolution on-axis and at > 60 deg. incidence, as a function of energy
      – Effective area as a function of energy and angle (and hence FOV)
      – Residual background as fraction of accepted high-latitude diffuse flux as a function of energy

K. Young
Sample TKR reconstruction plots showing reconstructed track multiplicity, origin point of tracks, and PSF (68 % and 95 % containment) for gammas

K. Young
LAT Software Quality Assurance

• End to end tests – “Mock Data Challenges”
  – Large scale test of entire LAT ground software system
    • Bulk processing of simulated source raw data through Level 1 processing followed by Level 2 analysis
    • “Single blind” – those doing analysis don’t know the underlying physics – their job is to discover it
    • Large scale effort involving large fraction of collaboration and certainly Science Working Groups
    • Anticipate 2-3 Mock Data Challenges prior to launch
      – In mid 2003
      – Towards the end of 2005
Manpower & Schedule

• Manpower Estimates
  – Compile Server – 1 FTE for 6 months (KY)
  – Document Builder – 1 FTE for 2 months (KY)

• Schedule
  – Should have “version 1” of Release Management system ready by March (many components have already been built) for serious evaluation and testing