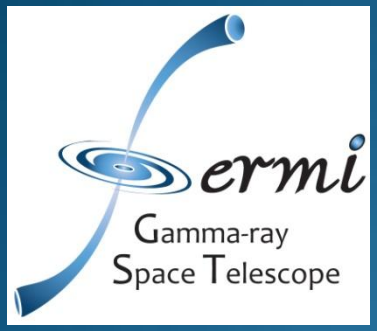
# Fermi Offline Infrastructure



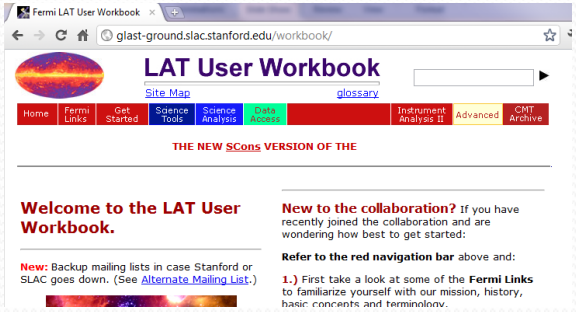

Heather Kelly
heather@slac.stanford.edu
PPA Scientific Computing Apps

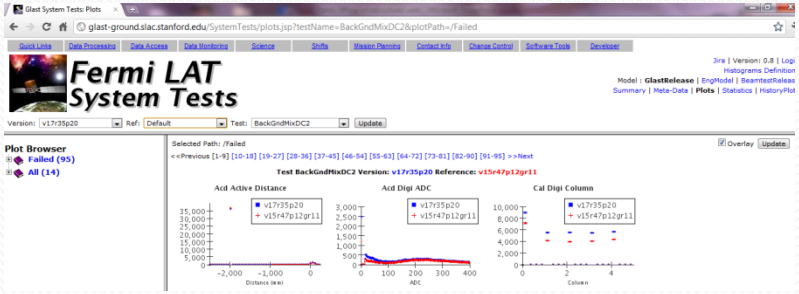
LAT was launched as part of the Fermi Gamma-ray Space Telescope on June 11th 2008.

HEP Meets Astrophysics

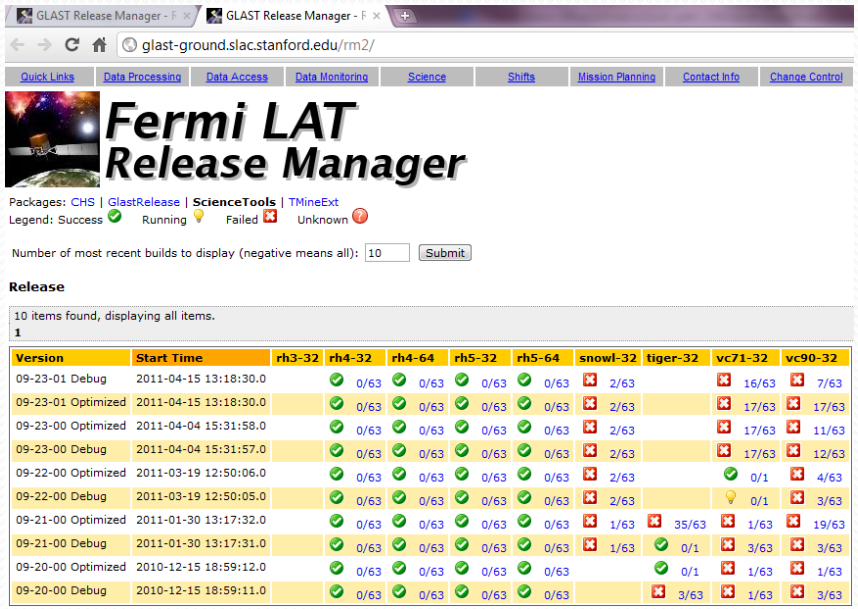
# Philosophy

- Reuse code from the community where possible.
  CLHEP, Geant4, ROOT, FITS, Gaudi, Xerces, MySQL
- Custom code is made non-Fermi specific when feasible.
- One central (SLAC) CVS repository
- Code Documentation via Doxygen
- User cookbook became our online workbook
- Flexibility - runtime customization
- ONE code system for simulation, test data analysis and flight
- Test, Test, Test, before flight
  Reviews (PDR, CDR), balloon flight, multiple test beam instruments, data challenges
- Validation via system tests
- Support both Linux and Windows

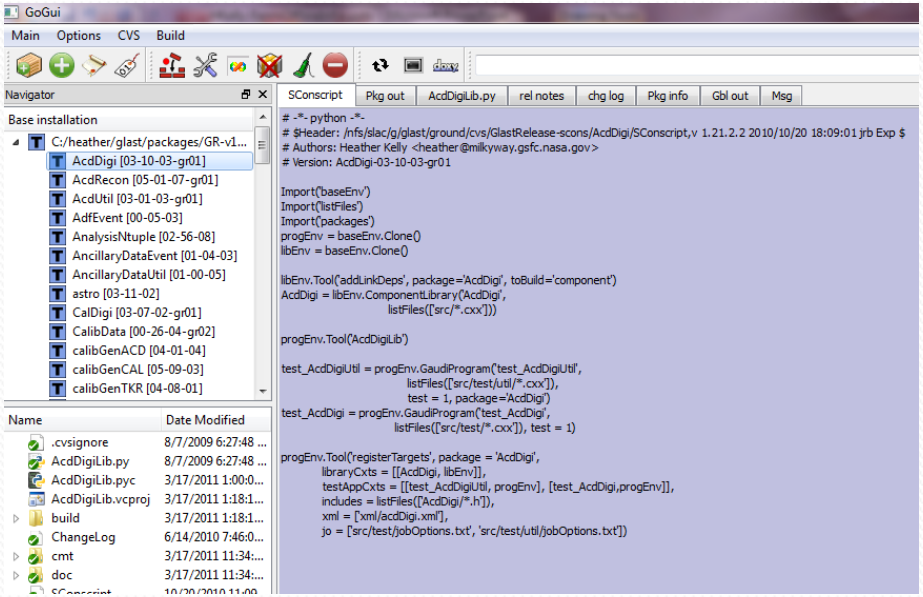


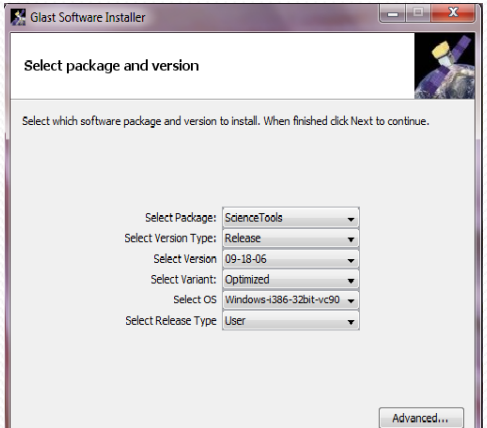



# Automated Builds via RM




# Developer GUI: GoGui



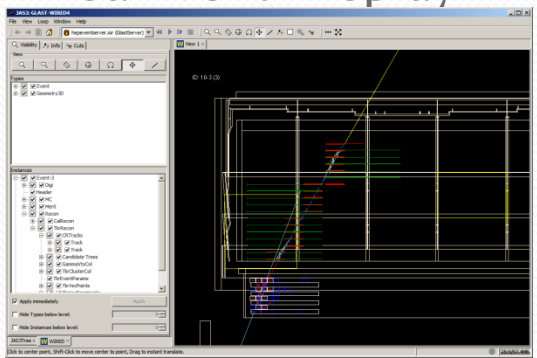




GUI Installer

# Offline Software

## GlastRelease

- C++ Monte Carlo (Geant4) simulation with data-driven geometry and reconstruction software
- Gaudi framework
- Output file format: ROOT
- Utilized as part of the data processing pipeline.
- Builds via CMT
  RHEL4-32/gcc 3.4
  Windows/VC++ 7.1
- Wired Event Display



## ScienceTools

- All software related to scientific analysis of Fermi LAT data
- Written in C++ with python interfaces via swig
- Most is publicly available through the Fermi Science Support Center (GSFC)
- Builds via SCons
  RHEL4-32/RHEL4-64/gcc3.4
  RHEL5-32/RHEL5-64/gcc4.1
  Snow Leopard gcc 4.2
  Windows/VC++9
- Obeys HEASARC/OGIP standards which implies FITS files and use of parfiles to provide user input.

# Why Gaudi?

- By 1999 (pre-proposal) we had plenty of code:  detailed Monte Carlo & prototype reconstruction with ASCII ntuple output.
- Gaudi provided a framework for our code
  - Clear lines of division
  - Separation of data from algorithms
  - Standard interfaces for algorithms
- Basic Services:
  - Event loop, Logging
  - Transient Data Store (TDS)
  - Job options, where input ASCII file contains job parameters
- We utilize the core functions of Gaudi
- Gaudi and G4 can play nice, despite that both want to control event loop. We usurp G4's RunManager and instead use our custom interface to G4 to request one event at a time.
- Using Gaudi led us to CMT.
- http://proj-gaudi.web.cern.ch/proj-gaudi/

# Why ROOT?

- ROOT provided more robust file format which could accommodate storage of MC data, as well as digitization and full reconstruction, as well as ntuples.
- Object Oriented I/O fit our OO design
- 5 star support ★★★★★
- Python interface (PyROOT)
- ROOT has grown substantially since we adopted it. Fortunately, it has remained modular.
- Certainly a learning curve for those wanting publication quality plots.
- http://root.cern.ch

# Challenges

- Changing build systems (CMT -> SCons)
- Maintaining multiple branches of MC/recon in CVS
  This is also a plus, in that L1 processing is shielded.
- Aging externals and the need to push ahead to modern versions of compilers and operating systems (OBF, Gaudi…)
- Brain Drain

# Successes

- Testing before flight paid off
- Early start provided ample time to write code and refactor
- Technical Writer produced Online Workbook
- Use of HepRep as our interface to Event Display
- Distributed, diverse group works well together
  Instant messaging, EVO, yearly face-to-face meetings
  In 2005, we had 25 developers spread over 9 time zones.