SciTools Infrastructure

- **Scope of Science Tools Core work**

- **Development Model**

- **Organization**

What are we trying to build?

- **Current model:**
  - **C++ - based atomic tools that can be chained together**

  - **Typically a collection of main routines that perform specific functions**
    - **Parameters obtained from IRAF files**
    - **I/O by FITS files**

- **Required components**
  - **File repository**
  - **Code management system**
  - **Code documentation tool**
  - **Collection of external libraries**
    - **IRAF, FITS, plotting packages**
  - **Machinery to build and test releases**

Software Development Approach

- **Reuse much of LAT's infrastructure**

- **Enable distributed development via cvs repository**

- **CMT tool permits equal development on Windows and Linux**

- **documentation and coding reviews enforce coding rules**

- **"Continuous integration"**
  - **Eliminate surprises for incoming code releases**
  - **Build code every night; alert owners to failures in build or running of <u>unit tests</u>. Results tracked in database.**
  - **Developing comprehensive <u>system tests</u> in multiple source configurations. Track results in database; web viewable.**

Science Tools Toolkit

Package	Description	Provider	Status
PIL, PIL++	IRAF parameter access	HEASARC	In use
cfitsio	FITS file manipulation	HEASARC	In use
XSPEC, Sherpa	For GRB spectral modeling	HEA standards	Under consideration
Root	gui etc	HEP standard	Under consideration
Plplot	Plotting package	HEA standard	Under consideration
python	Scripting	World standard	Under consideration
doxygen	Code doc tool	World standard	In use
AIDA	Analysis abstract interface	Emerging HEP standard	Under consideration
Visual C++/gnu	Development envs	World standards	In use
CMT	Code mgmt tool	HEP standard	In use
cvsweb	cvs web viewer	World standard	In use
cvs	File version mgmt	World standard	In use

Guided Tour

- **View of cvs**   file management

- **Code Layout for CMT**   package organization, builds

- **Doxygen example**   inline doc; "reference manual"

- **Release Manager Builds**   tracks past, in-progress and future releases. Tag early and often!

- **Code reviews**   Likelihood

Organization

- **Core software group meets once per week (nominally) on Tuesdays**

  – **Co-chaired with Bob Schaefer**

  http://www.slac.stanford.edu/exp/glast/ground/ScienceTools/core/meetings/

  – **stcore@glast.stanford.edu**

  – **Discussions on design & tools**

Outlook

- **Machinery is in place for code development**

- **We do not yet have the habit of frequent tagging and declarations of releases**

- **No idea on System Tests yet!**