

LArSoft news

R. Snider
Fermilab

LArSoft General Meeting
Aug 21, 2013

Issues to discuss

- Details and update on status of “git transition”
- LArSoft developer position

The “git transition” plan

- More than just “transitioning to git”:
 - SRT to cmake transition for builds
 - “relocatable” ups integration for environment management
 - multi-repository build tool for builds and development workflow
 - Use of git-flow to facilitate development workflow
 - New distribution scheme / system / capabilities
 - New redmine site (actually reclaiming the original...)
 - New (and updated) documentation
- Plan documented and approved end of July, early Aug
 - Official version of tasklist and status:
 - <https://sharepoint.fnal.gov/project/LArSoft/Lists/Categorized%20Tasks/AllItems.aspx>
 - Milestones:
 - <https://sharepoint.fnal.gov/project/LArSoft/Lists/Tasks/AllItems.aspx>

The plan

- Organized into five work areas
 - LArSoft code base and cmake build
 - Multi-repository build system (mrb) development and deployment
 - Redmine changes
 - Documentation
 - git repository transition
- More details available since last general meeting

Note that steps are time-ordered within each work area in the following

- But dependencies between work areas not so easily discerned
(working on this...)

LArSoft code base and cmake build

- Work related to organizing code into git repositories
 - Strategy is to group together code in the same level of functionality
- Several sub-areas of work
 - Break LArSoft into components
 - Identify a few low-level packages for prototyping the transition
 - Identify experiment-specific code throughout LArSoft
 - Migrate to experiment-specific repositories
 - Agree on contents of all components
 - cmake and ups integration
 - Prototype transition using a few low-level packages
 - Add balance of components when issues are understood
 - Set up new LArSoft areas
 - in /grid/fermiapp/larsoft
 - Set up experiment areas and builds
 - Each experiment will have it's own build

mrbs development and deployment

- Builds on work done by g-2 and art
 - Working with g-2 on some aspects of design and implementation
 - Will soon make available the documentation for the g-2 system
 - A preview of some of the system features, capabilities
 - Will also create a new mrbs project in redmine
 - Document the full feature set, status, etc, there
- Task list
 - Design capabilities that allow checking of dependency tree
 - The main extension beyond g-2 version
 - Implement basic functions
 - checkout, setup, build
 - Implement extended functions

Redmine changes

- Want to re-claim the original “LArSoft” redmine project for the new system / infrastructure
- Task list
 - Create larsoft-alpha and larsoft-beta redmine projects
 - Start as sub-projects of LArSoft-svn
 - Perform transition prototyping in larsoft-alpha
 - Test full procedure in larsoft-beta
 - Create a new Redmine projects for legacy content in “LArSoft”
 - Migrate legacy content from current location to this new project
 - Re-configure existing “LArSoft” project to host content for the new system
 - Production repositories
 - Documentation

Documentation

- Prior to going live, document set to include
 - The development model
 - git references
 - Detailed workflow examples
 - Tutorials
 - LArSoft release / administrative procedures

Whenever possible:

- Make available documentation that allows preview of final system elements
- Two major sub-areas here
 - Re-locate existing documentation to the new LArSoft redmine project
 - Will need to update anything that needs to change because of the new system
 - Write new documentation
 - Draft versions needed for testing
 - Final versions needed before going live

git repository transition

- This is the formal execution of the transition
- Task list:
 - Create alpha versions of repositories to prototype the transition
 - Prototyping work extends through full builds of experiment releases from experiment-specific and core LArSoft repositories
 - Completion Implies that basic mrb functionality is available
 - Prototyping complete after first pass validation
 - Create beta versions of repositories
 - Test of the final procedure and validation
 - Perform the migration
 - Freeze svn and create pre release prior to migration
 - Production transition to git
 - Create a “post” release under new system
 - Compare results from both
 - If we get identical results (and documentation ready): Go Live!

Where are we now?

- Task status

- Created new redmine project “LArSoft Alpha Test”
 - <https://cdcvns.fnal.gov/redmine/projects/larsoft-alpha>
 - Prototype repositories live there. Work notes are on the wiki page.
- Identified low-level components for prototyping
 - LArGeo: in larsoft-alpha-largeo with SimpleTypesAndConstants + Geometry
 - LArData: in larsoft-alpha-lardata with RawData + RecoBase + SummaryData + OpticalDetectorData + AnalysisBase + Utilities + Simulation
 - LArReco: in larsoft-alpha-larreco with AnalysisAlg + RecoObjects + RecoAlg

Some re-factoring of content is needed here, both in terms of packages within components, and the classes within packages.

Will initially discuss proposal with experiment representatives

- Should happen soon

Where are we now?

- Task status (cont'd)
 - Working on prototyping the transition
 - Write scripts to transfer repositories and metadata.
 - The procedure is understood
 - Converting to cmake build is next
 - Created new redmine project “LArSoft cvs (legacy site)”
 - <https://cdcvs.fnal.gov/redmine/projects/larsoft-cvs>
 - Will be home to content currently on “LArSoft” page.
 - Next task is to migrate the content
 - Define new components / repositories for all of LArSoft
 - A high priority. Important to understand dependencies early on.
 - Move “experiment-specific” software out of core LArSoft packages
 - Implies we agree on what those are
 - Want to keep other re-factoring to a minimum, but need to be open to allowing this across the rest of the project also

[Will need to consult with everyone on the details here](#)

Where are we now?

- Task status (cont'd)
 - Experiment-specific software areas and builds
 - Expected to create software areas early on
 - Decided to wait on this until the cmake build is understood better
 - Will get to this later, but for now, the set of tasks to create experiment-specific software accounts / areas is deferred.
 - Documentation
 - Need to migrate existing documentation to LArSoft redmine project
 - The new home after migrating away from LArSoft-svn
 - Update the documentation as we go
 - Will need to do this one-by-one. May need to ask experiments for assistance with this, but do not know the extent of changes needed yet.
 - Project will develop new build / release documentation
 - Will work on this once we've defined the component packaging

Status summary

- Transition is now in plan execution phase
 - Expect to see significant progress over the next few weeks
 - Can track progress at the sharepoint site and on redmine pages
- Do not yet have a complete time estimate
 - Hoping for a time scale of 1 - 2 months
- Work to follow immediately after the tasks discussed here:
 - Distribution system
 - Extended mrb functionality
 - Remote builds

LArSoft developer wanted

- Have an opening for a developer to work on LArSoft
 - Code management tasks
 - Development related to LArSoft “infrastructure” issues (eg, architecture)
 - Algorithm development

A consultant position at present.

Funded for two years. Working to make it permanent.

- Inform colleagues or make suggestions. **Contact rs@fnal.gov ASAP.** Plan to move quickly.

“git transition” status

-
- Task list circulated at end of July, approved Aug 5
 - Official version is on the LArSoft project sharepoint site:
 - <https://sharepoint.fnal.gov/project/LArSoft/Lists/Categorized%20Tasks/AllItems.aspx>
 - Milestones:
 - <https://sharepoint.fnal.gov/project/LArSoft/Lists/Tasks/AllItems.aspx>
 - Status of both are being tracked there
- Current state
 - The real work of the transition plan is being executed now
 - Very early in the plan, however
 - Well behind where we expected to be by this time
 - Delayed by Snowmass a bit, but bulk was just allowing more latency than necessary

Detailed “git transition” status

- Reporting milestones:
 - Identify 2-3 low-level components for prototyping transition
 - Working prototype (git + cmake build + ups) on low-level components
 - Final LArSoft component packaging identified
 - Final component packaging with ups + cmake (no mrb)

 - LArSoft redmine page reclaimed for new repositories
 - LArSoft-alpha and LArSoft-beta sub-projects created

 - mrb basic functionality working
 - mrb extended functionality working

 - conversion to git readiness checks completed
 - git conversion completed
 - LArSoft distribution (via tarballs)
 - Demonstrate automated remote install and build

| |
|-------------|
| Completed |
| In progress |
| Not started |

Schedules and milestones

- Receiving very little guidance from stakeholders on schedule requirements for git transition
 - Are there drivers from stakeholders?
 - The sooner is clearly the better here? What should our target be?
 - Review drivers, estimate from code management.
- Milestones
 - The project needs to know what “we” are working on.
 - Will make rounds with each stakeholder to document who is doing what
 - Can then go from there to get to some milestones