

LArSoft Coordination Meeting

Release and project report

Erica Snider
Vito di Benedetto
Giuseppe Cerati
Lynn Garren
Katherine Lato
Gianluca Petrillo
Saba Sehrish

Fermilab

April 10, 2018

Today's agenda and speakers

- Release and project report (Erica)
- Requirements for a new LArSoft event display (Erica)
- Update on LArG4 refactoring (Hans Wenzel)
- Request for an update to Geant4 version in LArSoft (Aaron Higuera)
- AOB

Releases

- Last week
 - v06_73_00 released on April 5
 - **First release with clang** builds (but no debug builds in this version)
 - Qualifier **c2**
 - e15 and c2 available on SLF6, SLF7, Sierra (SIP disabled), High Sierra (SIP disabled)
 - Changes related to clang
 - See [Building with clang wiki page](#) (under “LArSoft Internals”) for type of changes needed
 - Some of the errors encountered point to problems in the code
 - Check your code if any changes were made for clang!
 - NOTE: This is **the last release for which El Capitan build will be provided.**
- This week
 - **Add metadata to simb::MCParticle / simb::MCTruth association** with index to the associated particle in MCTruth
 - New release of nutools
 - Accommodate the change above + added move operations to simb:MCParticle
 - Debug c2 versions available

Notes on the use of clang

- Available platforms
 - Last week
 - e15 and c2 builds for SLF6/7, El Capitan and Sierra (SIP disabled)
 - This week
 - e15 for SLF6 and SLF7
 - c2 for SLF7 (proposed), Sierra (SIP disabled), High Sierra (SIP disabled)

Notes on the use of clang

- Simultaneous support for two (or more!) compilers
 - Can we ensure that changes developed under one do not break the other?
 - Best to avoid merging to develop until both are known to work.
 - Turn around time for Mac build in CI system is hours
 - Too slow to be a good tool to prevent problems
 - Threatens to break development environment if not done carefully
 - Feeds into other questions about the code integration model\
 - Some ideas
 - Force checks before merging
 - We don't check now. Do we need to?
 - Move to C++17
 - Current clang supports C++17.
 - Current version of gcc does not. Would need to move to art 2.11 + e17. (There are other good reasons to move to art 2.11!...)
 - Not a complete solution
 - Other ideas?
 - Discussion? Should we survey to find out who is developing where?

The end