LBNE Software Build Status and Plans

Brett Viren

Physics Department

BROOKHAVEN NATIONAL LABORATORY

JRATM

10 Dec 2014

Outline

Build Issues

Plan

Status

Near Term Work

Build Issues for art-based packages

- **Inverted dependency**: CET's CMake-based build system depends on UPS and a huge body of scripts that implement Fermilab software infrastructure.
- Effects **LArSoft** for 35t and FD simulation, **artDAQ** for 35t readout
- Violates LBNE **requirements**:
  - Effectively impossible to port to all LBNE platforms.
  - No control of versions of packages by LBNE.
  - Build from pristine source + known patches (since mostly addressed by CET)
- Various **inadequacies** in UPS/CET build design:
  - Does not build all LBNE software.
  - Leads to painful development environment.
  - High barrier for use outside Fermilab.
  - Build system remains as run-time dependency.
  - Huge machinery/inconvenience for a small code base ($\sim$1/5 size of just ROOT).

Brief History

- Deficiency in art build system describe to developers on April 2013.
- I pointed out that it hinders adoption of art by others (since proven by the rejection of art by CAPTAIN and SuperNEMO).
- At that time, I personally offered to help work on fixing their problem.
- Offer was rejected, problem not acknowledged.
- Various meetings with CET at Fermilab.
  - Provided input on how to improve build system, some ideas taken but core problem retained.
  - Presented *Worch*, successfully defended it against list of requirements from CET.
  - Presented plans to reimplement package-level CMake build system to be free of UPS/CET entanglements.
- Told: "go do it and then maybe we'll think about accepting it"

Basic Plan

For now, LBNE must rely on art so we fix the problem ourselves:

- Do the work outside of official repositories to assure **no disruption of status-quo** (done).
- Develop **pure-CMake** build following best practices and replacing UPS/CET encumbered system (done).
- Develop **build automation and release management** system: *Worch* (done).
- Port to official repositories as optional, **status-quo kept default**. Port in order:
  1. `lbnecode`
  2. LArSoft
  3. art
- Test pure-CMake build.
- "Throw switch" to make it default.
- Switch can be thrown at high layer, low-layer kept as-is.

Status

- LBNE GitHub org established
  `https://github.com/LBNE`
- art, LArSoft and `lbnecode` packages in tracking forks.
- Pure CMake build complete and partly tested
- *Worch* release management / build automation essentially done.
- Initial production of UPS binary packages from *Worch* build results complete.

Near Term Work

- Polish and test CMake purification.
- Polish and test UPS binary package production.
- Port changes to official repositories as optional.

Each item is estimated to take 1 week FTE.
Only counts actual time doing technical tasks, not any required discussion, reviews, meetings.

Interaction with LArSoft and art teams

- LBNE will port `lbnecode` ourselves.
- Start discussions with LArSoft team on details of how to integrate purification work.
  - So far, good encouragement for our work.
  - I expect a high probability of acceptance.
- When complete, attempt to push same patterns to art.
  - I have no understanding of likelihood of acceptance.
  - We receive no messages of encouragement.

Worse case scenario, LBNE will apply CMake purification via patch and not integrate with Fermilab/CET.
Definitely, we hope to avoid this.