Larsoft/art Build Experiences

Brett Viren
(with input from others)

Physics Department

BROOKHAVEN
NATIONAL LABORATORY

November 13, 2013

Contents

Experience/Suggestions from Building Outside FNAL

Topics For tomorrow morning

Reports of three people:

- B. Viren, BNL, SL5 (built through larsoft)
- J. Marshal, Cambridge, SL6 (built through larsoft)
- B. Bhuyan, IITG, CentOS6 (not yet successful)

Cambridge Report

- Many thanks to John Marshal for a detailed report.
- This presentation is mostly from cribbed that.
- I happen to share all his observations and suggestions.

Source Packaging

- Bundling of multiple source packages into one is unsatisfactory.
- Build scripts should be bundled together but separate from source.
- Scripts should download the source from external repository as part of the build.
- Mismatch between build scripts and source bundle versions, had to resort to guessing what to download based on time stamps.
- Both art_externals and nu_extras contain ROOT and these versions can differ.

Build Configuration

- Build scripts should be configured by a central configuration file.
- It should be possible for the user/installer to modify this configuration such as to change which Geant4 is built.
- This build config should be versioned to make overall releases.
- An overall build script should be used to assure correct build ordering and consistent qualifier usage.
- Unnecessary dependencies, real or misunderstood, eg `cstxsd` not needed, not clear why GCC/Python was needed.

Building Externals

- Building externals not particularly difficult, but time consuming.
- Must take care to build in correct order.
- Build qualifiers inconsistent, must read each build script to understand what should be applied.
- A "preview" method to check that all settings are correct/consistent is wanted. Some way to run through the build scripts without actually downloading/installing anything.

Larsoft Itself

- Difficulty in creating the setup script.
- Should be produced by the build system.

Additional Comment from me

- The focus here is just on larsoft. LBNE has other end-user applications which do not depend on art. Some goals:
  - All LBNE software built using the same system.
  - Two approaches, want both:
    1. One big suite - same GCC-from-source even if an application doesn't need it
    2. Separate islands - don't build GCC/art if don't need it
- Automation is important
  - Non-experts must build this
  - Collaboration can't afford to spend days/weeks babying an installation
  - We will have many releases as time goes on
  - We want a continuous integration system which must be automated.
- Autonomy is important
  - LBNE must be free to choose the package versions it needs.
  - End user/installers must be likewise be free to choose.

Summary

Desired but currently lacking features:

automation one thing to download and run, unattended running

transparency no bundling of source - system should pull from upstream archives/repositories, factor out meta data (versions and qualifiers) from scripts

consistency assure no version/qualifier mismatches, do not allow regressions for old releases, no duplicate source in bundles (but see above)

coverage only include packages that are actually needed, support multiple end-goal applications with possibly conflicting dependencies

usability last-step environment management

(eg, https://github.com/brettviren/worch)

art-related build meeting

Tomorrow, 9am, WH1E: technical build-related details pertaining to art will be discussed. Nominal agenda:

- LBNE S&C Requirements pertaining to the use of art (Brett)
- Overview of cet/art CMake-based build system (Chris)
- The MRB tool (Lynn)
- The worch installation system (Brett)

(Note: a longer version of the requirements and a shorter version of the worch presentations will be given tomorrow afternoon in the LBNE session.)