LBNE S&C biweekly meeting
Nov 13, 2014

Larsoft Build Process Experience and Update

Patrick Gartung
FNAL
gartung@fnal.gov

Objectives

- Rewrite the cmake files for the larsoft package to only use cmake macros available directly from the cmake package.
- Use the changes that Ben Morgan made to the art package as a guide.
- Record the options used to invoke cmake in a worch config

Getting larsoft into github

- Checkout all of the larsoft repos from FNAL redmine (aka upstream)
- Create an empty repo for each in LBNE account on github
  - https://github.com/LBNE
- Push upstream develop and master branches to github
- Github rejected because of 120MB file that was added in initial svn commit to larsoft and then removed.
- Followed directions that github gave for removing large files
  - https://help.github.com/articles/working-with-large-files/

Setting up build environment

- Started a build from scratch using lbne-build package (lbne specific config for worch)
  - https://github.com/LBNE/lbne-build
- Spent some time debugging why “make test” was failing for FNALCore and fnalart packages
- Updated config for the latest version of FNALCore and fnalart
- Updated sqlite3 version and attempted to use the FindSQLite3.cmake macros which should work without passing extra -D options to cmake.
- Figured out that the cmake macros work for a system install of sqlite3 but not in worch environment. Fell back to passing -D options to cmake.
- Updated clhep and root versions needed by fnalart.

Build first larsoft package – larcore

- Started with larcore because of dependency chain
  Larcore<-Lardata<-Larevt<-Larsim<-Larreco<-Larana
- Quickly added rules to build regular libraries and add header files in install area
- Needed to create cmake modules directory and copy over Find*.cmake from fnalart.
- Found a bug in the way root was configured while generating dictionary library

  -DGCCXML_EXECUTABLE={gccxml_install_dir}/bin/gccxml

Comparing it to upstream

- Library sizes were much smaller than upstream
- Location of lib and bin directories different
  - OK because lib location exported in cmake files
- All header files in same location
- Needed to add .gdml and .pl and .C files in gdml dir, .fcl files in job dir, and test scripts in bin

Making it equivalent to upstream

- After adding cxx flag -g sizes were comparable
- Dug through cetbuildtools macros and ups config for larcore to reverse engineer install_gdml(),install_fhicl() and install_scripts() macros.
- Cmake globbing patterns, ups set variables, etc.
- Didn't implement install_source() to install the library source files following the install pattern of fnalart.

Side effect of resizing larcore repo

- While comparing gdml dir found file was missing larcore/Geometry/gdml/icarus.gdml
- In the repo I found icarus.gdml.REMOVED.git-id
- I had passed the wrong flag to the “BFG” utility recommended by github
  - BFG removed top N files by size not files over github 100MB limit
- Both “BFG” and “git filter-branch” removed the one file over 100MB but all commits after that are rewritten changing their commit id.
- Asked github for an exception to the 100MB file size limit for the larsoft repos. They graciously raised the size limit to 200MB.
- All larsoft repos now cloned to github with the upstream commit ids.
- Cloned the lbnecode repo on github just to be sure there were no size limit problems.

Preparing to build lardata

- Added lines to larcore cmake files to export library interfaces following fnalart example
- Added larcoreConfig.cmake.in to be used by other package to import larcore interfaces
- Added GENIE and LHAPDF to worch config and built them.
- Figured out where the get the sources for nutools package.
- Need to modify nutools cmake files and store changes as a patch to upstream in worch.