Geant 4.10 Notes

Richard Longland

2014-08-20 Wed

1 Install

- Follow install instructions at `http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/InstallationGuide/`

2 CMake notes

- Fast compile to check for coding errors add a `/fast`
  `make G4run/fast`
- Can make several build directories for each src directory for testing different things
- See `http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/InstallationGui` `html/ch04s03.html`
- Good technique: have (i) release build and (ii) a dev build
  1. Once you have something working:
     `cd LENAGe/build-release,`
     `cmake -DCMAKE_BUILD_TYPE=Release ../src`
     `make -j6`
  2. To develop something,
     `cd LENAGe/build-dev,`
     `cmake -DCMAKE_BUILD_TYPE=Debug ../src`
     `make -j6`, edit code, `make -j6`, etc.
- Remember that git tracking of source code also needs to be kept track of
- `http://www.cmake.org/cmake/help/cmake_tutorial.html`
- Building a Geant4 application with CMake needs:
  - A 'src' directory with everything in it (like the old LENAGe directory, for example)
  - Note that there is another directory within this called 'src' with all the .cc files (again, like with the old code)
  - CMakeLists.txt is the CMake script. Should be in the base 'src' (e.g., for example B1 it's in `~/geant4/B1/src`)
  - A separate 'build' directory (`~/geant4/B1/build`)
- Example B1 is a good place to start
  - `mkdir ~/geant4/B1/src`
  - `cp -r /opt/geant4/geant4.10.00.p01-src/examples/basic/B1/* ~/geant4/B1/src`
  - `mkdir ~/geant4/B1/build`
  - `cd ~/geant4/B1/build`

    – `cmake ../src`
    – `make`

3 THM Notes

- Copy CMakeLists.txt over from example B1
- Comment out the `add_custom_target` command near the bottom. This screws up if the executable is named the same as the project
- Copying scripts directory is now much easier than the `configure_file` stuff in the B1 example. Now we can use `file(COPY`
  - See line beginning with "set(SCRIPTS" in THM for an example
- Added check for datasets, line beginning with "foreach(dsname"
- Added check for multithreading, line begins with "`find_package(Geant4 COMPONENTS multithreaded`"
- To use root, we need the FindROOT.cmake file (copied from the Root FAQ)
- Add root include directories and lib directories (see section starting with "include(FindROOT.cmake)")
- Needed to add G4SystemOfUnits.hh to a bunch of files since it was removed from the standard linking
- Same for G4PhysicalConstants (for example, for 'pi', 'proton$_{massc2}$', etc.)
- Make sure that `target_link_libraries` also points to the `${ROOT_LIBRARIES}` (see line beginning "`target_link_libraries`")
- To make the dictionary needed to output root files
  - Add an `EXECUTE_PROCESS` to compile the dictionary in `CMakeList.txt`
  - See `http://hypernews.slac.stanford.edu/HyperNews/geant4/get/analysis/410/3/1.html`
  - Make sure it's referred to in `add_executable`
    `add_executable(THM THM.cc RootOutputDataDICT.cxx ${sources} ${headers})`