

Status of the Fast and Full Simulation build system based on CMake

Marco Corvo

CNRS and INFN

II SuperB Collaboration Meeting

Frascati December 13-16, 2011

Project status

- ① This project has been presented almost one year ago
- ② We can claim that is mature enough to be widely exploited by users
 - Improvements are clearly not only possible but also needed
- ③ I will summarize the major goals of the project

Major improvements

- One of our goal was to get definitely rid of SRT as a base for building FastSim
- Not completely accomplished but now we can avoid using old *.mk* files to declare dependencies among packages
 - dependencies are now handled completely by CMake via the usual CMakeLists.txt file
- If a package needs a new dependency, it must be declared in the list **LIB_LINK_LIST**

Specific targets

- ① The user can specify, with the bash script `cmakeRun.sh`, the target he's willing to build
 - For example you could only be interested in building the library of a specific package:
`cmakeRun.sh buildlibs -p Package1`
 - or the binaries of a set of packages:
`cmakeRun.sh buildbins -p Package1,Package2`
- ② The build of test binaries has been decoupled in order to avoid failures in the main build process due to errors in some 'secondary' executables

Builds based on a given installed release

- 1 With SRT the user is able to build a single package relying on a release which is already installed (using `.mk` files and `sbsrtpath`)
- 2 With CMake the same feature is available
- 3 The dependencies are managed via specific `.cmake` files (a bit like old `.mk` ones, but smarter)

I just discovered a problem (bug?) while building a single package.
I have to investigate on it ...

CMake for FullSim

The CMake build system is now available also for FullSim

- Same principles as for FastSim
- CMakeLists.txt has been specialized for both Full and FastSim
 - **sbnewrel** selects the right one depending on the Release you're asking for
- A new CMake macro has been added to manage the external dependency on **Geant4**
- Some minor adjustments were needed to make CMake files deal with both Full and FastSim

Plans

- Try to investigate deeper the possibility to use CPack as a packaging tool to create RPMs
 - Told many times already but with limited time slots it's quite hard
- The SRT scripts, which interface with Subversion, show a lot of errors/warnings as old BaBar SRT now copes with new SuperB infrastructures
 - They should be cleaned, at least, but more likely rewritten (and here there's room for discussions on the script language to adopt)

Conclusions

- I think that CMake proved to be a nice tool to manage software configuration in place of SRT, that is hand-managed Makefiles
- Once the system is in place it's quite straightforward both to add new externals and to add packages
- To me the most useful feature is that we have a single point to look at in case of problems
 - ... and the scripts are much as simple as bash ones, provided you digest a little bit the syntax