



Marco Corvo
CNRS and INFN

BUILDING TOOLS STATUS



Outline

- Status
- Plans
- Conclusions



Status

- CMake added as alternative build system (Makefile generator) to FastSim V0.2.7
 - Based on CMake 2.6 and available for build on SL4, SL5 (32 and 64 bit) and MacOS
 - **It doesn't substitute but works in parallel with SRT**
- CMake build managed by a shell script
 - Copes with initial build configuration
 - Replicates some SRT behaviours (as regards make installdirs)
 - Still lacks other features (script management, initial setup)



Status

- Provides Release and Debug builds
- Still coupled to some SRT features
 - Build dependencies calculated starting from `link_X.mk` and `bin_X.mk` files
 - Management of placeholders for external libraries (ROOT, CLHEP) to be improved
 - Previously made with `arch_spec_*` files



Plans

- Decouple Cmake build system from SRT
 - Make the new system substitute SRT, avoiding to replicate obsolete and useless features
- Provide more granular builds
 - Now “**make all | libs**” to build binaries and libraries or only libraries
 - Add binaries, test binaries



Plans (cont'd)

- Extend build platforms as required
- Improve the management of external packages
 - In view of extending the build system to the FullSim software
 - By now FastSim depends only on Root and CLHEP and the management is much simpler



More plans

- First prototype of build reports on CDash
 - Just for developers
 - Needs a lot of improvements
 - Complete V0.2.7 build generates almost 8k warnings
 - Must clean massively the code in order to make CDash usable and useful
- Possibly integration with CPack, an automatic packaging system



More plans (cont'd)

- Investigate automatic build frameworks
 - Needed to manage nightly or trigger builds based on SVN repos commits
 - Useful to test whether committed code breaks a build
 - Possible candidate is **Buildbot** but a survey is undergoing to identify other ones
- Extend the usage of CMake to FullSim
 - Needs to go on with the work on “packetization” of code



- Need to think to a strategy for unittest into packages
 - Thinking of an automatic system to build SuperB software it would be useful to have small test units which check code and trigger build failures



Conclusions

- FastSim build based on CMake is available, though in “beta” version
- Efforts to improve and make it more reliable and efficient are needed
- There are other issues related
 - Decoupling from SRT
 - Automatic builds
 - Further integration with web based report application (CDash) and packaging systems (CPack)



Conclusions (cont'd)

- We strongly encourage people involved in FastSim development to start using the new system in order to provide feedback and to help finding bugs