

Geant4 CMake: 10.2 to 10.3

Ben Morgan

THE UNIVERSITY OF
WARWICK

Core fixes/features since 10.1

- CMake version bump -> 2.8.12 (-> 3.3 *for C++11*)
- Windows builds of DLL/Archives improved
- Default use of @rpath in OS X install names
- “Granular” mode of system CLHEP can be used if requested
- G4ENSDFSTATEDATA exported to environment by default

Improved Build Product Layout

- When building Geant4, libraries etc are output to same directory structure as they will be installed in:

+ - YourBuildDir/

+ - <buildscripts>

+ - BuildProducts/

+ - <MODE>/

+ - bin/

+ - lib/

+ - libG4global

Present for Xcode,
VS etc



Install Guide Updates

- cmd/PowerShell guide for Windows to be added
 - *Addresses Bugzilla #1729 and makes install process of the toolkit a bit easier and close to identical to UNIX.*
- Will retain Visual Studio GUI guide for user applications
- Will not provide guides for MinGW or Cygwin

lambdas

```
[] {foo();}
```

constexpr

initializer lists

regex

C

++

11

nullptr

```
shared_ptr<T>,  
unique_ptr<T>,  
weak_ptr<T>
```

```
auto i = v.begin();  
for(auto x : collection)
```

See Parallel 6b and Plenary 7!

Modularization: G4processes

- Reached symbol table limit on Windows VS2015
 - See, e.g., <http://cdash.cern.ch/viewBuildError.php?buildid=163810>
- **Plan, for discussion, to break into three libs for 10.2:**
 - “G4processes-{hadronic,electromagnetic,general}”
- *No changes needed on your or users part, CMake will transform any compile/link paths/dependencies as needed*

Modularization: After 10.2

- Geant4 == >3500 .hh files over 145 modules
- “Granular” too small, “Global” both too big/small
- How then to organise code into libraries?
 - Let’s start discussion this week on how to approach this!
 - There are CMake, Code organization and C++ API design issues here

Modularization: Sketch of CMake API

```
# sources.cmake
include(Geant4CMakeAPI)

geant4_add_module(G4MyModule
  PUBLIC_HEADERS G4MyModule.hh
  SOURCES        G4MyModule.cc
)

geant4_module_link_libraries(G4MyModule
  PUBLIC  G4globman G4csg
  PRIVATE G4intercoms ZLIB::ZLIB
)

geant4_module_compile_definitions(G4MyModule
  PUBLIC $<$<CONFIG:Debug>:DEBUGMYMODULE>
)
```


Managing Data Libraries

- We need to look at how data libraries are read/located
- Environment variable based setup awkward with too many potential points of failure
- Access via lots of small files has/is causing issues for MT performance?
- Let's use this week to discuss ideas
 - Slide notes include some of mine

“Geant4GMake.gmk”

- A new pure-Make implementation of configuration for applications still using old GMake build system
 - See branch `cmake-feature-gmakesupport`
- Behaviour is a “superset” of environment configuration
 - Now fails if option requested which is not available
 - Options enabled by environment/command line/make variables, with reporting of option source

Geant4GMake.gmk

For users, a very simple one line replacement, with usage:

```
$ make
```

```
...
```

```
$ make VERBOSE=1
```

```
...
```

```
$ make geant4_help
```

```
name := exampleB1
G4TARGET := $(name)
G4EXLIB := true
```

```
# geant4-config gives make fragment path
include $(shell geant4-config --gmake-file)
```

```
.PHONY: all
all: lib bin
```

```
include $(G4INSTALL)/config/binmake.gmk
```

```
visclean:
```

```
    rm -f g4*.prim g4*.eps g4*.wrl
```

```
    rm -f .DAWN_*
```

```
# Optional geant4_help target(s)
```

```
include $(G4GMAKE_TARGETS)
```

Summary

- CMake system/documentation in good shape for 10.2
- Please come to Parallel 6b and Plenary 7 for C++11!
- Two major Software Management tasks for 10.3:
 - *Library modularisation*
 - *Data library location and format*
- Let's start the discussion on these during this week!