



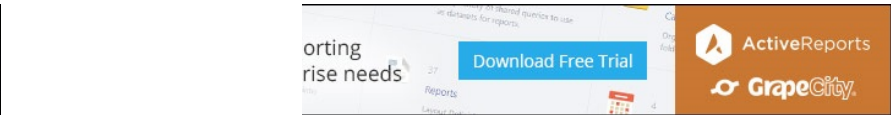
Learn, Share, Build

Each month, over 50 million developers come to Stack Overflow to learn, share their knowledge, and build their careers.

Join the world's largest developer community.

GoogleFacebookOR

Trouble building tests with Google Test, Clang, and libc++



I tried building Google Test with the following CMake configuration:

```
$ CMAKE_CXX_COMPILER="clang++" CMAKE_CXX_FLAGS="-std=c++11 -stdlib=libc++  
-U__STRICT_ANSI__" cmake ../source
```

Building shows CMake picked the right compiler, but my compiler flags aren't passing through:

```
$ VERBOSE=1 make  
...  
/Users/jfreeman/local/bin/clang++ -I/Users/jfreeman/work/googletest/source  
/include -I/Users/jfreeman/work/googletest/source -DGTEST_HAS_PTHREAD=1 -o  
CMakeFiles/gtest.dir/src/gtest-all.cc.o -c /Users/jfreeman/work/googletest/source  
/src/gtest-all.cc  
...  
/Users/jfreeman/local/bin/clang++ -I/Users/jfreeman/work/googletest/source  
/include -I/Users/jfreeman/work/googletest/source -DGTEST_HAS_PTHREAD=1 -o  
CMakeFiles/gtest_main.dir/src/gtest_main.cc.o -c /Users/jfreeman/work/googletest  
/source/src/gtest_main.cc
```

The end goal is that I want my project, which builds with Clang and libc++, to have tests built with Google Test. That means I need Google Test built with libc++ as well.

c++ cmake clang googletest libc++

asked Nov 2 '12 at 15:52

 **John Freeman**
584 9 19

1 Answer

Using variables on the command line with CMake sometimes requires the -D (for Define) flag.

```
$ cmake -DCMAKE_CXX_COMPILER="clang++" -DCMAKE_CXX_FLAGS="-std=c++11  
-stdlib=libc++ -U__STRICT_ANSI__" ../source
```

answered Nov 2 '12 at 17:07

 **katyhuff**
311 1 4

2 I had to add -DCMAKE_CXX_FLAGS="... -DGTEST_USE_OWN_TR1_TUPLE=1" to make it work. – r1uba Mar 29 '13 at 19:05