

Join GitHub today

GitHub is home to over 20 million developers working together to host and review code, manage projects, and build software together.

Sign up

Dismiss

# Using gtest command line arguments with test target's command line arguments #765

New issue

Closed bofinbabu opened this issue on 29 Apr 2016 · 5 comments



bofinbabu commented on 29 Apr 2016

I'm feeding my application with cmd line args. I've been feeding gtest arguments to select a particular test-case (using --gtest-filter=TestCaseName.\*) . I've read from a stack overflow answer that gtest will automatically parse it's arguments and leave the rest for the program (test target).  
Is it true? (I've observes that it's not working in my case)  
If yes, do I need to pass gtest args before any other arguments? (order)

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

3 participants



Bu11etmagnet commented on 29 Apr 2016

Yes, InitGoogle{Test,Mock} will remove the arguments it recognizes. You don't have to put the gtest arguments before others. You should, however, call InitGoogleTest early, before you start processing arguments yourself.

```
#include "gtest/gtest.h"

int main(int argc, char **argv)
{
    ::testing::InitGoogleTest(&argc, argv);

    for (int i = 1; i < argc; ++i) {
        printf("arg %2d = %s\n", i, argv[i]);
    }

    return RUN_ALL_TESTS();
}

$ ./argrm_test foo bar baz
arg 1 = foo
arg 2 = bar
arg 3 = baz

$ ./argrm_test foo bar baz --gtest_filter=foo
arg 1 = foo
arg 2 = bar
arg 3 = baz
Note: Google Test filter = foo
[=====] Running 0 tests from 0 test cases.
[=====] 0 tests from 0 test cases ran. (0 ms total)
[ PASSED ] 0 tests.
```

3



bofinbabu commented on 29 Apr 2016 • edited

In my case I need to access program's command line arguments inside a test case.  
  
For that I had to do a *dirty* trick of making `**argc_and_argv` global variables.

```
int argc; // Making arg and arv global to access within TESTs
_TCHAR** argv;
```

```
TEST(SampleTest, Test0){
    // Here I'm accessing argc and argv
}
```

```
// my main
GTEST_API_ int _tmain(int t_argc, _TCHAR** t_argv){

    argc = t_argc;
    argv = t_argv;

    printf("Initializing gtest \n");
    ::testing::InitGoogleTest(&argc, argv);

    return RUN_ALL_TESTS();
}
```

In this case when I do `TheExe.exe --gtest_filter=SampleTest.Test0 arg1 arg2`

I'm getting the following error message from Google Test.

This program contains tests written using Google Test. You can use the following command line flags to control its behavior:

```
Test Selection:
--gtest_list_tests
    List the names of all tests instead of running them. The name of
    TEST(Foo, Bar) is "Foo.Bar".
--gtest_filter=POSITIVE_PATTERNS[-NEGATIVE_PATTERNS]
    Run only the tests whose name matches one of the positive patterns but
    none of the negative patterns. '?' matches any single character; '*'
    matches any substring; ':' separates two patterns.
--gtest_also_run_disabled_tests
    Run all disabled tests too.

Test Execution:
--gtest_repeat=[COUNT]
    Run the tests repeatedly; use a negative count to repeat forever.
--gtest_shuffle
```



bofinbabu commented on 29 Apr 2016

Sorry the issue got solved.  
I used - instead of \_



bofinbabu closed this on 29 Apr 2016



ZacharyIG commented on 8 Sep 2016

Hi,

Can you elaborate on how you resolved? I'm having the same issue. Where did you need to use - instead of \_?

Thanks!  
-zack



Bu11etmagnet commented on 12 Sep 2016

See the first comment. He used `--gtest-filter=` originally. This is not recognized by gtest. The correct option is `--gtest_filter` .