

公告

DigitalOcean优惠码  
这里的博客将不再更新，最新博客  
请移步至：  
我的独立博客：  
<http://blog.coderzh.com/>



微信公众号：  
hacker-thinking  
昵称：CoderZh  
园龄：11年3个月  
粉丝：801  
关注：10  
[+加关注](#)

搜索

找找看

随笔分类

- [Agile\(2\)](#)
- [Android\(3\)](#)
- [ASP.NET\(3\)](#)
- [C#\(20\)](#)
- [C/C++\(24\)](#)
- [Cocos2d-x\(1\)](#)
- [Emacs\(2\)](#)
- [Google App Engine\(7\)](#)
- [JAVA\(3\)](#)
- [Linux\(1\)](#)
- [Lua\(2\)](#)
- [Python\(66\)](#)
- [Ubuntu\(9\)](#)
- [VBS\(4\)](#)
- [安全性测试\(9\)](#)

随笔-234 文章-10 评论-2052

玩转Google开源C++单元测试框架Google Test系列(gtest)之六 - 运行参数

一、前言

使用gtest编写的测试案例通常本身就是一个可执行文件，因此运行起来非常方便。同时，gtest也为我们提供了一系列的运行参数（环境变量、命令行参数或代码里指定），使得我们可以对案例的执行进行一些有效的控制。

二、基本介绍

前面提到，对于运行参数，gtest提供了三种设置的途径：

- 1. 系统环境变量
- 2. 命令行参数
- 3. 代码中指定FLAG

因为提供了三种途径，就会有优先级的问题，有一个原则是，最后设置的那个会生效。不过总结一下，通常情况下，比较理想的优先级为：

命令行参数 > 代码中指定FLAG > 系统环境变量

为什么我们编写的测试案例能够处理这些命令行参数呢？是因为我们在main函数中，将命令行参数交给了gtest，由gtest来搞定命令行参数的问题。

```
int _tmain(int argc, _TCHAR* argv[])
{
    testing::InitGoogleTest(&argc, argv);
    return RUN_ALL_TESTS();
}
```

这样，我们就拥有了接收和响应gtest命令行参数的能力。如果需要在代码中指定FLAG，可以使用testing::GTEST\_FLAG这个宏来设置。比如相对于命令行参数--gtest\_output，可以使用testing::GTEST\_FLAG(output) = "xml:"来设置。注意到了，不需要加--gtest前缀了。同时，推荐将这句放置InitGoogleTest之前，这样就可以使得对于同样的参数，命令行参数优先级高于代码中指定。

```
int _tmain(int argc, _TCHAR* argv[])
{
    testing::GTEST_FLAG(output) = "xml:";
    testing::InitGoogleTest(&argc, argv);
    return RUN_ALL_TESTS();
}
```

测试生活感悟(7)  
程序人生(15)  
代码安全(3)  
单元测试(19)  
公告(13)  
每周总结(4)  
软件测试(30)  
设计模式  
性能测试(7)  
学习笔记(27)

随笔档案

2015年9月 (1)  
2015年8月 (2)  
2015年6月 (4)  
2015年5月 (2)  
2015年4月 (5)  
2015年3月 (1)  
2014年5月 (2)  
2014年4月 (2)  
2011年5月 (1)  
2011年3月 (1)  
2011年1月 (1)  
2010年12月 (3)  
2010年11月 (3)  
2010年10月 (2)  
2010年9月 (6)  
2010年8月 (2)  
2010年7月 (4)  
2010年6月 (3)  
2010年5月 (4)  
2010年4月 (9)  
2010年3月 (6)  
2010年2月 (3)  
2010年1月 (16)  
2009年12月 (6)  
2009年11月 (3)  
2009年10月 (4)  
2009年9月 (3)  
2009年8月 (2)  
2009年7月 (7)  
2009年6月 (2)

最后再来说下第一种设置方式-系统环境变量。如果需要gtest的设置系统环境变量，必须注意的是：

- 1. 系统环境变量全大写，比如对于--gtest\_output，响应的系统环境变量为：GTEST\_OUTPUT
- 2. 有一个命令行参数例外，那就是--gtest\_list\_tests，它是不接受系统环境变量的。（只是用来罗列测试案例名称）

三、参数列表

了解了上面的内容，我这里就直接将所有命令行参数总结和罗列一下。如果想要获得详细的命令行说明，直接运行你的案例，输入命令行参数：/? 或 --help 或 -help

1. 测试案例集合

命令行参数	说明
--gtest_list_tests	使用这个参数时，将不会执行里面的测试案例，而是输出一个案例的列表。
--gtest_filter	<p>对执行的测试案例进行过滤，支持通配符</p> <p>? 单个字符</p> <p>* 任意字符</p> <p>- 排除，如，-a 表示除了a</p> <p>: 取或，如，a:b 表示a或b</p> <p>比如下面的例子：</p> <p>./foo_test 没有指定过滤条件，运行所有案例</p> <p>./foo_test --gtest_filter=* 使用通配符*，表示运行所有案例</p> <p>./foo_test --gtest_filter=FooTest.* 运行所有 “测试案例名称(testcase_name)” 为FooTest的案例</p> <p>./foo_test --gtest_filter=*Null*:*Constructor* 运行所有 “测试案例名称(testcase_name)” 或 “测试名称(test_name)” 包含Null或Constructor的案例。</p> <p>./foo_test --gtest_filter=--*DeathTest.* 运行所有非死亡测试案例。</p> <p>./foo_test --gtest_filter=FooTest.*-FooTest.Bar 运行所有 “测试案例名称(testcase_name)” 为FooTest的案例，但是除了FooTest.Bar这个案例</p>
--gtest_also_run_disabled_tests	<p>执行案例时，同时也执行被置为无效的测试案例。关于设置测试案例无效的方法为：</p> <p>在测试案例名称或测试名称中添加DISABLED前缀，比如：</p> <div><pre>// Tests that Foo does Abc. TEST(FooTest, DISABLED_DoesAbc) { ... }  class DISABLED_BarTest : public testing::Test { ... };</pre></div>

- 2009年4月 (12)
- 2009年3月 (5)
- 2009年2月 (2)
- 2009年1月 (3)
- 2008年12月 (7)
- 2008年11月 (9)
- 2008年9月 (8)
- 2008年8月 (7)
- 2008年7月 (8)
- 2008年6月 (9)
- 2008年5月 (33)
- 2008年4月 (6)
- 2008年2月 (1)
- 2007年12月 (3)
- 2007年11月 (3)
- 2007年10月 (7)
- 2007年9月 (1)

系列文章

- Python天天美味系列
- 攻击方式学习系列
- 瘦客户端那些事
- 玩转gtest系列

读书笔记

- Python网络编程
- xUnit Test Patterns
- 卓有成效的程序员

友情链接

积分与排名

积分 - 547881  
排名 - 209

最新评论

1. Re:Google Test交流与测试开发  
经验总结  
您好，可以发一份给我吗，  
1034828302@qq.com  
谢谢  
  
--xuxd88

```
// Tests that Bar does Xyz.
TEST_F(DISABLED_BarTest, DoesXyz) { ... }
```

设置案例重复运行次数，非常棒的功能！比如：

--gtest\_repeat=1000     重复执行1000次，即使中途出现错误。

--gtest\_repeat=-1     无限次数执行。。。

--gtest\_repeat=1000 --gtest\_break\_on\_failure     重复执行1000次，并且在第一个错误发生时立即停止。这个功能对调试非常有用。

--gtest\_repeat=1000 --gtest\_filter=FooBar     重复执行1000次测试案例名称为FooBar的案例。

2. 测试案例输出

命令行参数	说明
--gtest_color=(yes no auto)	输出命令行时是否使用一些五颜六色的颜色。默认是auto。
--gtest_print_time	输出命令行时是否打印每个测试案例的执行时间。默认是不打印的。
-- gtest_output=xml[:DIRECTORY_PATH\];:FILE_PATH]	将测试结果输出到一个xml中。  1.--gtest_output=xml:    不指定输出路径时，默认为案例当前路径。  2.--gtest_output=xml:d:\    指定输出到某个目录  3.--gtest_output=xml:d:\foo.xml    指定输出到d:\foo.xml  如果不是指定了特定的文件路径， gtest每次输出的报告不会覆盖，而会以数字后缀的方式创建。xml的输出内容后面介绍吧。

3. 对案例的异常处理

命令行参数	说明
-- gtest_break_on_failure	调试模式下，当案例失败时停止，方便调试
-- gtest_throw_on_failure	当案例失败时以C++异常的方式抛出

2. Re:玩转Google开源C++单元测试框架Google Test系列(gtest)(总楼主好厉害，写的通俗易懂，我等晚辈拜服！

--One\_IT\_One\_World

3. Re:Google Test交流与测试开发经验总结

您好，一直在看您的文章，写的非常不错，我最近也刚开始写Gtest，能否方便给我发一份PPT呢？  
邮箱：1206050824@qq.com  
谢谢您啦！

--One\_IT\_One\_World

4. Re:玩转Google开源C++单元测试框架Google Test系列(gtest)(总楼主好厉害，写的通俗易懂，我等晚辈拜服！

--Big~Newbie

5. Re:PyQt4学习资料汇总  
正在学习， 邮箱：

1464983982@qq.com 感谢分享  
--jobs\_huang

6. Re:gtest参数化测试代码示例  
博客园的链接改了，是这个地址：

--canbeing

7. Re:玩转Google开源C++单元测试框架Google Test系列(gtest)之一 - 初识gtest

@xiao\_1bai编译的目标就是生成lib文件，你已经成功了。现在可以在你的项目引用gtestd.lib...

--cnbloghzc

8. Re:ViEmuVS2013-3.2.1 破解  
安装失败，提示：  
所需要的.NET Framework 没有

--xiake007

9. Re:玩转Google开源C++单元测试框架Google Test系列(gtest)之七 - 深入解析gtest

谢谢，作者哥哥，这么多章，最精彩这章。领教了，谢谢

--\$JackChen

--gtest_catch_exceptions	是否捕捉异常。gtest默认是不捕捉异常的，因此假如你的测试案例抛了一个异常，很可能会弹出一个对话框，这非常的不友好，同时也阻碍了测试案例的运行。如果想不弹这个框，可以通过设置这个参数来实现。如将--gtest_catch_exceptions设置为一个非零的数。  注意：这个参数只在Windows下有效。
--------------------------	--

四、XML报告输出格式



```
<?xml version="1.0" encoding="UTF-8"?>
<testsuites tests="3" failures="1" errors="0" time="35" name="AllTests">
  <testsuite name="MathTest" tests="2" failures="1" errors="0" time="15">
    <testcase name="Addition" status="run" time="7" classname="">
      <failure message="Value of: add(1, 1) Actual: 3 Expected: 2" type=""/>
      <failure message="Value of: add(1, -1) Actual: 1 Expected: 0" type=""/>
    </testcase>
    <testcase name="Subtraction" status="run" time="5" classname="">
    </testcase>
  </testsuite>
  <testsuite name="LogicTest" tests="1" failures="0" errors="0" time="5">
    <testcase name="NonContradiction" status="run" time="5" classname="">
    </testcase>
  </testsuite>
</testsuites>
```



从报告里可以看出，我们之前在TEST等宏中定义的测试案例名称(testcase\_name)在xml测试报告中其实是一个testsuite name，而宏中的测试名称(test\_name)在xml测试报告中是一个testcase name，概念上似乎有点混淆，就看你怎么看吧。

当检查点通过时，不会输出任何检查点的信息。当检查点失败时，会有详细的失败信息输出到failure节点。

在我使用过程中发现一个问题，当我同时设置了--gtest\_filter参数时，输出的xml报告中还是会包含所有测试案例的信息，只不过那些不被执行的测试案例的status值为“notrun”。而我之前认为的输出xml报告应该只包含我需要运行的测试案例的信息。不知是否可提供一个只输出需要执行的测试案例的xml报告。因为当我需要在1000个案例中执行其中1个案例时，在报告中很难找到我运行的那个案例，虽然可以查找，但还是很麻烦。

五、总结

本篇主要介绍了gtest案例执行时提供的一些参数的使用方法，这些参数都非常有用。在实际编写gtest测试案例时肯定会需要用到。至少我现在比较常用的就是：

- 1. --gtest\_filter
- 2. --gtest\_output=xml[:DIRECTORY\_PATH\][:FILE\_PATH]
- 3. --gtest\_catch\_exceptions

10. Re:最常用的Emacs的基本操作  
如果Emacs入手都算有些难度。。。那VIM怎么办?

--震灵

#### 阅读排行榜

1. 玩转Google开源C++单元测试框架Google Test系列(gtest)(总)(224598)
2. 玩转Google开源C++单元测试框架Google Test系列(gtest)之一 - 初识gtest(148792)
3. 玩转Google开源C++单元测试框架Google Test系列(gtest)之二 - 断言(107271)
4. 玩转Google开源C++单元测试框架Google Test系列(gtest)之三 - 事件机制(69736)
5. 玩转Google开源C++单元测试框架Google Test系列(gtest)之六 - 运行参数(55607)
6. 玩转Google开源C++单元测试框架Google Test系列(gtest)之四 - 参数化(55280)
7. C# 中使用JSON -DataContractJsonSerializer(54221)
8. 玩转Google开源C++单元测试框架Google Test系列(gtest)之七 - 深入解析gtest(50702)
9. PyQt4学习资料汇总(49953)
10. 代码覆盖率浅谈(49521)

#### 评论排行榜

1. PyQt4学习资料汇总(153)
2. 开源Granados介绍 - SSH连接远程Linux服务器(C#)(66)
3. (原创)攻击方式学习之(1) - 跨站式脚本(Cross-Site Scripting) (49)
4. 三年之痒(44)
5. NancyBlog - 我的Google App Engine Blog(42)

最后再总结一下我使用过程中遇到的几个问题:

1. 同时使用--gtest\_filter和--gtest\_output=xml时, 在xml测试报告中能否只包含过滤后的测试案例的信息。
2. 有时, 我在代码中设置 testing::GTEST\_FLAG(catch\_exceptions) = 1和我在命令行中使用--gtest\_catch\_exceptions结果稍有不同, 在代码中设置FLAG方式有时候捕捉不了某些异常, 但是通过命令行参数的方式一般都不会有问题。这是我曾经遇到过的一个问题, 最后我的处理办法是既在代码中设置FLAG, 又在命令行参数中传入--gtest\_catch\_exceptions。不知道是gtest在catch\_exceptions方面不够稳定, 还是我自己测试案例的问题。

系列链接:

- 1.玩转Google开源C++单元测试框架Google Test系列(gtest)之一 - 初识gtest
- 2.玩转Google开源C++单元测试框架Google Test系列(gtest)之二 - 断言
- 3.玩转Google开源C++单元测试框架Google Test系列(gtest)之三 - 事件机制
- 4.玩转Google开源C++单元测试框架Google Test系列(gtest)之四 - 参数化
- 5.玩转Google开源C++单元测试框架Google Test系列(gtest)之五 - 死亡测试
- 6.玩转Google开源C++单元测试框架Google Test系列(gtest)之六 - 运行参数
- 7.玩转Google开源C++单元测试框架Google Test系列(gtest)之七 - 深入解析gtest
- 8.玩转Google开源C++单元测试框架Google Test系列(gtest)之八 - 打造自己的单元测试框架



DigitalOcean的VPS主机, 稳定、速度快、价格也实惠。可以在上面部署独立网站或各种实用工具。  
我用了很久了, 确实不错, 极力推荐。  
使用这个链接购买可获得10美元优惠。

优惠链接: [DigitalOcean优惠码](#)



微信扫一扫交流

作者: [CoderZh](#)

公众号: hacker-thinking (一个程序员的思考)

独立博客: <http://blog.coderzh.com>

博客园博客将不再更新, 请关注我的「微信公众号」或「独立博客」。

作为一个程序员, 思考程序的每一行代码, 思考生活的每一个细节, 思考人生的每一种可能。

文章版权归本人所有, 欢迎转载, 但未经作者同意必须保留此段声明, 且在文章页面明显位置给出原文连接, 否则保留追究法律责任的权利。

分类: [单元测试](#), [C/C++](#)

标签: [Google Test](#)

- 6. CCNET+MSBuild+SVN实时构建的优化总结(40)
- 7. 创业三年来的一些感想 - 游戏篇(40)
- 8. CoderZh首款Python联机对战游戏 - NancyTetris1.0倾情发布(一) (37)
- 9. 代码安全系列(1) - Log的注入(35)
- 10. 程序员的信仰(35)

推荐排行榜

- 1. 玩转Google开源C++单元测试框架Google Test系列(gtest)(总)(26)
- 2. 创业三年来的一些感想 - 游戏篇(14)
- 3. 程序员的共鸣 - 读《卓有成效的程序员》(12)
- 4. 代码覆盖率浅谈(12)
- 5. 《xUnit Test Patterns》学习笔记5 - xUnit基础(10)
- 6. 三年之痒(9)
- 7. 玩转Google开源C++单元测试框架Google Test系列(gtest)之一 - 初识gtest(9)
- 8. 优美的测试代码 - 行为驱动开发(BDD)(8)
- 9. Python天天美味(总)(7)
- 10. Google App Engine的14宗罪(6)

好文要顶

关注我

收藏该文



CoderZh  
关注 - 10  
粉丝 - 801

+加关注

3

0

» 上一篇: 玩转Google开源C++单元测试框架Google Test系列(gtest)之五 - 死亡测试  
» 下一篇: 玩转Google开源C++单元测试框架Google Test系列(gtest)之七 - 深入解析gtest

posted @ 2009-04-10 01:10 CoderZh 阅读(55607) 评论(1) 编辑 收藏

评论列表

#1楼 2010-02-01 17:33 pscj's Blog

在xml测试报告中能否只包含过滤后的测试案例的信息 简单改一下源代码就OK了

支持(1) 反对(0)

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

- 【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库
- 【调查】有奖调研即刻参与，你竟然是酱紫程序猿！
- 【推荐】Vue.js 2.x 快速入门，大量高效实战示例
- 【活动】腾讯云 学生专属优惠套餐 多规格选择

最新IT新闻:

- 飞一天不成问题？微软测试无人驾驶AI滑翔机
- 谷歌：黑客攻击是用户凭证泄露最常见方式 但是网络钓鱼威胁最大
- 阿里巴巴本月发行美元债券 最高融资70亿美元
- 苏宁减持阿里巴巴股份：预计不超过550万股
- Mozilla推出速度更快的全新Firefox 57 (Quantum) 浏览器
- » 更多新闻...



**最新知识库文章:**

- 关于编程，你的练习是不是有效的？
  - 改善程序员生活质量的 3+10 习惯
  - NASA的10条代码编写原则
  - 为什么你参加了那么多培训，却依然表现平平？
  - 写给初学前端工程师的一封信
- » 更多知识库文章...

站长统计