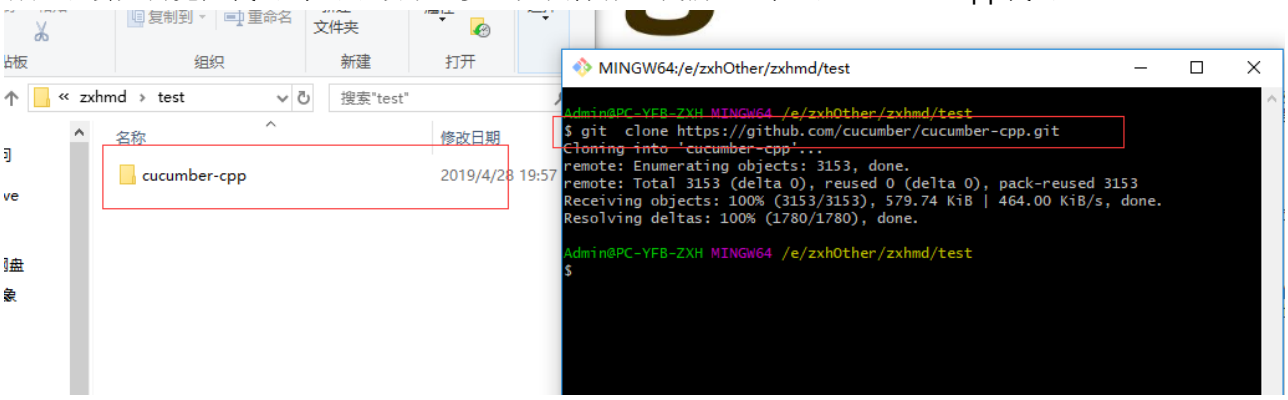
cucumber-cpp 安装部署文档

准备文件

文件信息	**version**	下载地址	说明
git	2.21.0	https://git-scm.com/downloads	必须
cucumber-cpp	477090e	https://github.com/cucumber/cucumber-cpp.git	branch=master 必须
cmake	3.9.1-win32-x86	https://cmake.org/files/	必须
ruby	2.6.3.1 x86	http://www.ruby-lang.org/en/downloads/	必须
cucumber	2.4.0	http://www.ruby-lang.org/en/downloads/	必须,并且低于3.0版本
Boost	1.63.0	https://www.boost.org/users/history/	必须
vs	2015	https://visualstudio.microsoft.com/zh-hans/downloads/	必须
Qt	5.9.1	http://download.qt.io/archive/qt/	可选,example 需要

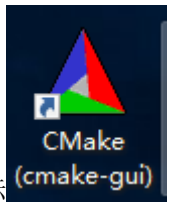
Setp1 安装GIT 检出源码

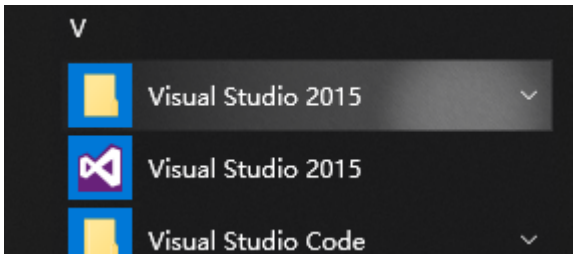
1. 功能：检出cucumber-cpp代码， 未配置getst 与 gmock 的话cucumber cpp会自己去检出对应的代码
2. 安装流程：只需要官网上下载对应的版本即可，傻瓜式安装只需要点击下一步，不在赘述，不会自行百度。
3. 安装好之后右键鼠标应该能看见一个git bash 与 git Gui，git bash 就是我们需要操作的地方。ps git 是非常强大的版本管理工具。小白教程
4. 你应该会如何克隆代码 才可继续下一步，如图操作，我们已经检出cucumber-cpp代码



5.

Setp2 安装基础软件

1. 安装cmake ruby vs 傻瓜教程 点击 下一步 下一步 不会自行百度
2. 安装完成以后测试一下 我们呼出cmd.exe 输入ruby -v 命令应该能看到如图的结果 cmake 与 vs 安装完成会生产对应图标 



Setp3 安装需要更改一下的软件

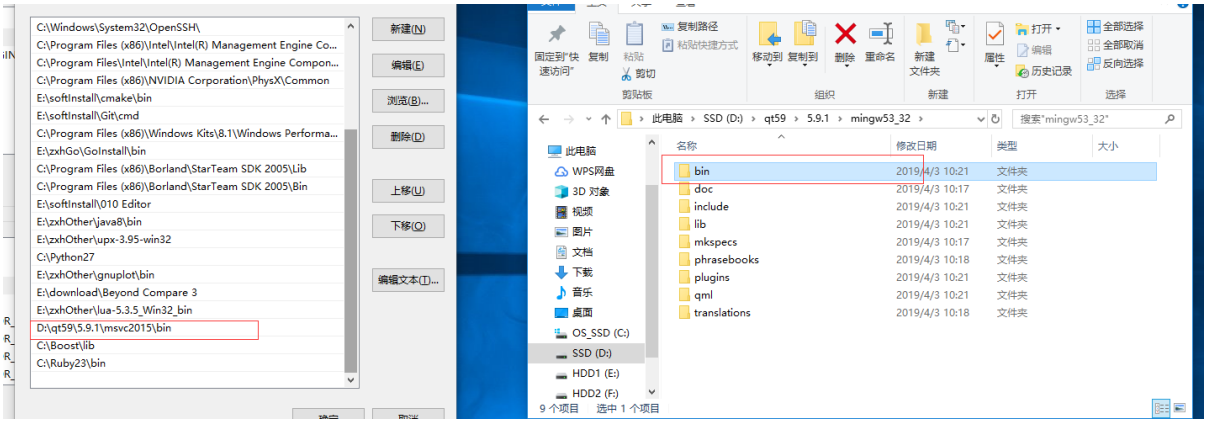
1. 安装cucumber
    1. 我们右键呼出git bash 输入一下命令 gem install cucumber -v **2.4.0**

```
Admin@PC-YFB-ZXH MINGW64 ~/Desktop
$ gem install cucumber -v 2.4.0
```

    2. 稍等片刻（ruby 默认镜像在国外，如果下载失败 可以开一个VPN，或者镜像连接其他源，不会自行百度）
    3. 输入cucumber --version 命令应该能看到对应版本

```
Admin@PC-YFB-ZXH MINGW64 ~/Desktop
$ cucumber --version
*** WARNING: You must use ANSICON 1.31 or higher (https://github.co
on/) to get coloured output on Windows
2.4.0
```

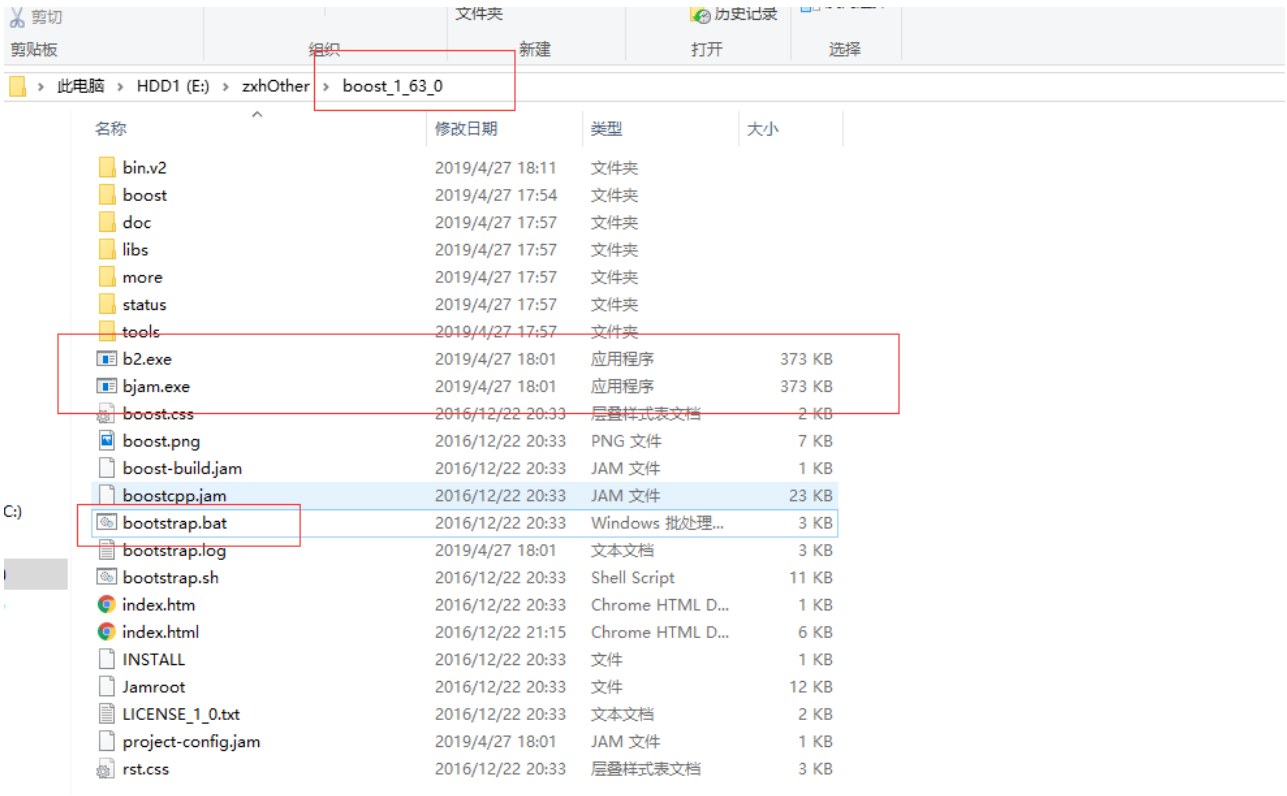
2. 安装Qt
    1. 去qt官方下载对应安装包，双击点开在选项里面勾选 **msvc** ，因为cucumber-cppexample里面有用到qt,如果不需要example 则可以跨过此步
    2. 安装好后我们添加qt 的bin目录到电脑path



ps.这里其实可以不安装qt 只需要include 相应的.h 文件与 lib 但是笔者在编译cmakelist的时候遇到了问题，在教程就采用百分百不会错的办法，对了配置了path 记得**重启**电脑哟。

Setp4 编译Boost

1. 我们去官网下载boost源码 解压开 然后 运行 bootstrap.bat（这里会卡一下，其实是程序在配置），完成以后会生产b2.exe 与 bjam.exe 如图

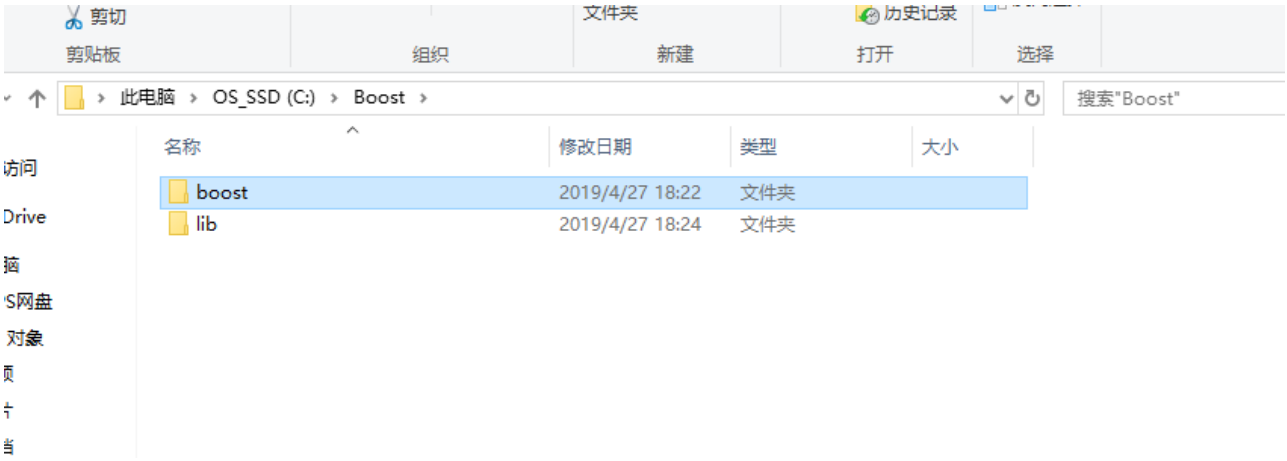


,ps b2 与bjam 都是编译代码的，区别是b2 较新，这里我们采用b2

2. 在当前目录打开git bash 输入一下命令,我这里是生成在**c:/Boost** 下，有需要的同学自己修改对应参数，我们这里使用--with-xx 的参数只生成我们想要的库，boost还是非常大的全部编译的话笔者i5-8500 需要三个多小时，指定库的话十几分钟就搞定,不知道参数自行百度。

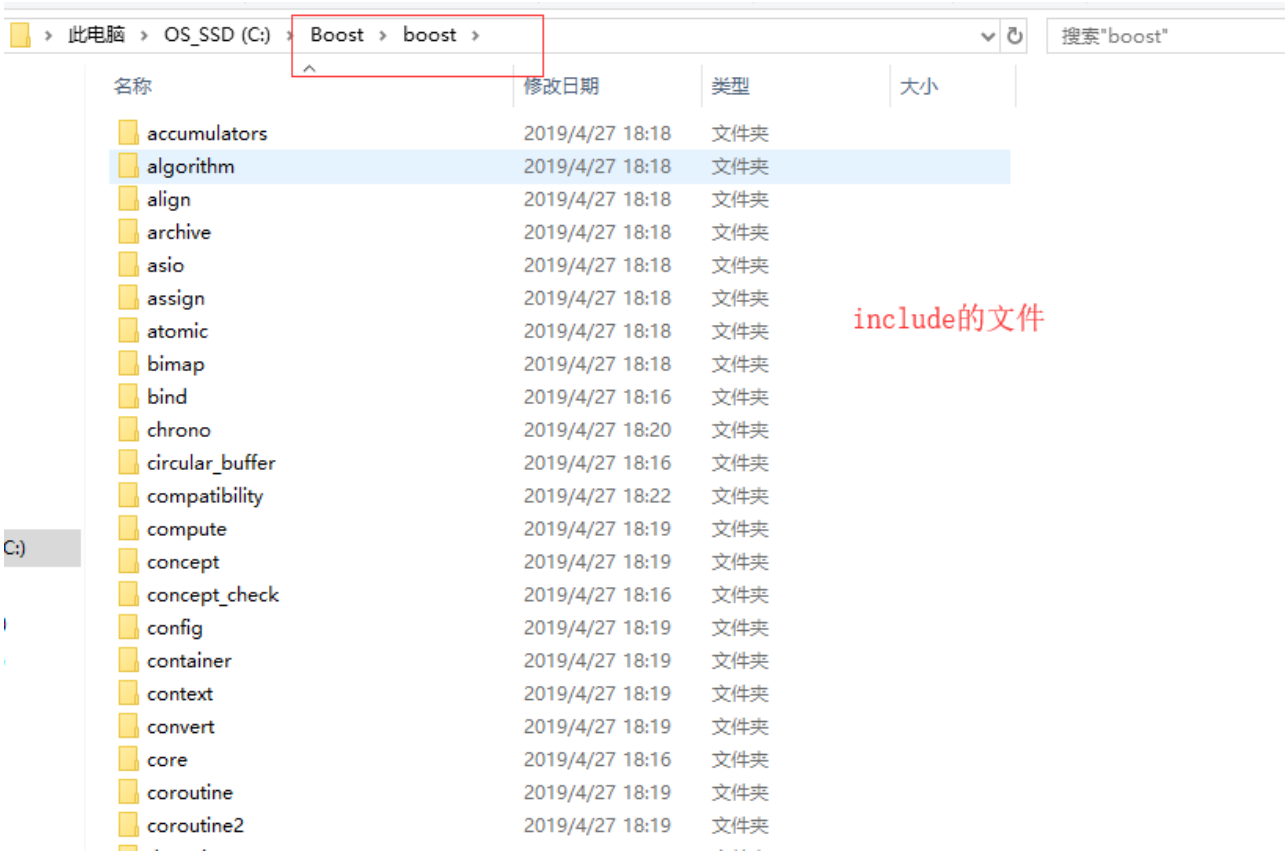
```
 ./b2 --prefix=c:/Boost --build-type=complete --with-thread --with-atomic --with-
regex --with-system --with-date_time --with-test --with-filesystem --with-chrono -
-with-program_options address-model=32 install
```

3. 在等待一会儿之后应该能在对应的生成目录看到一些lib 文件 这里我们调整一下目录结构，原因是用cmakelist构建项目需要调整好目录结构不然会报一些 找不到boost，找不到 thread.lib 等。目录结构如下

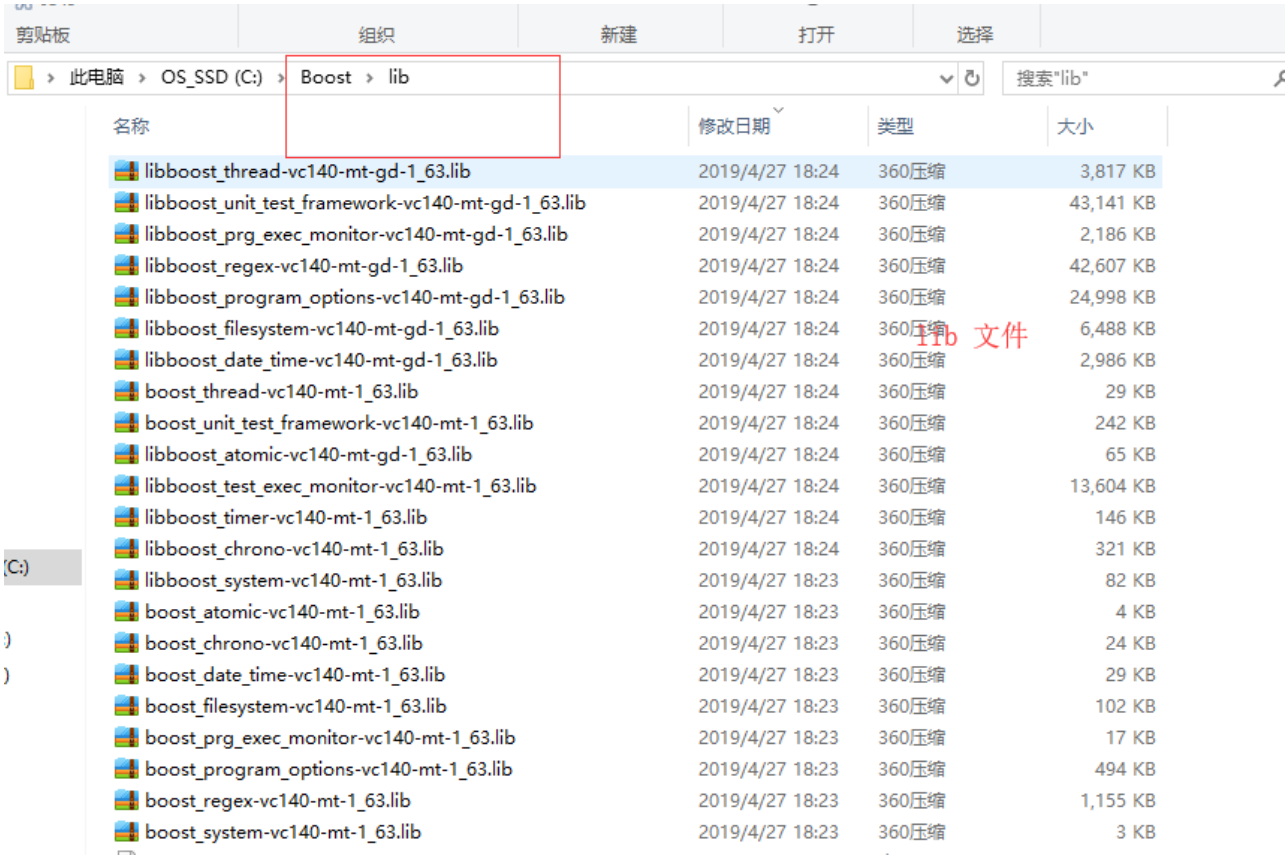


4.

5.

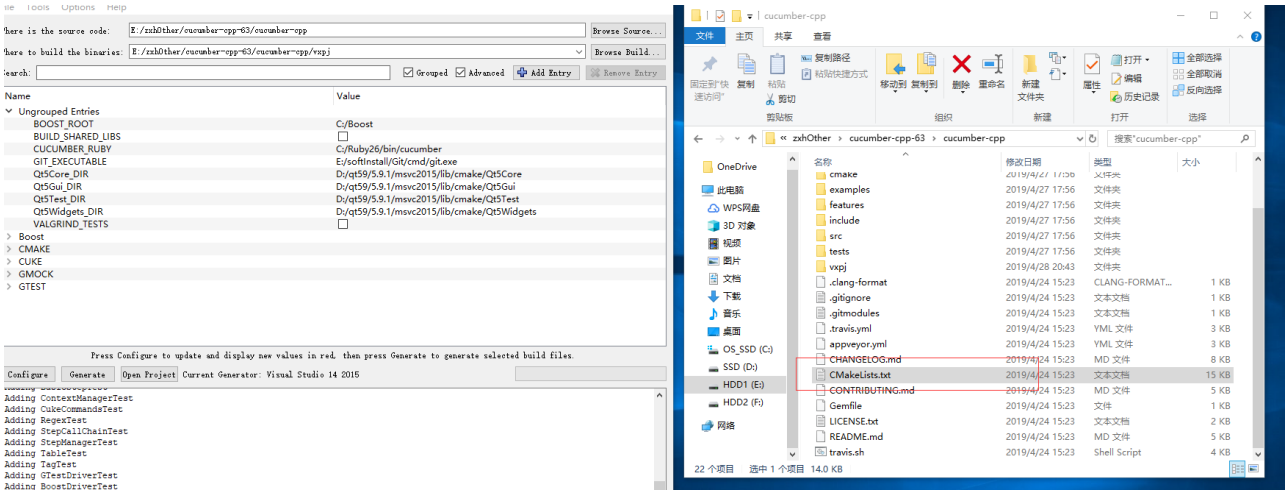


6.

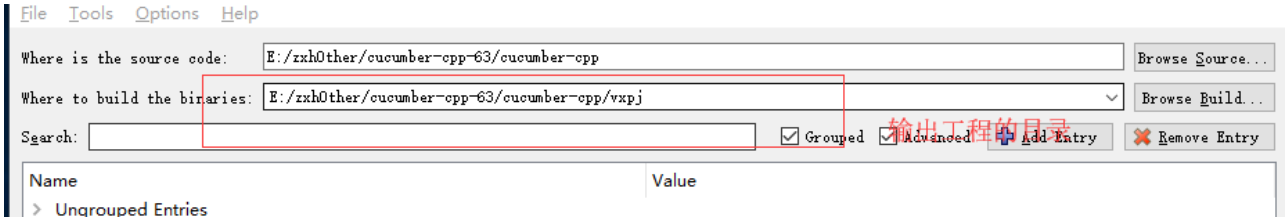


Setp5 构建cucumber-cpp

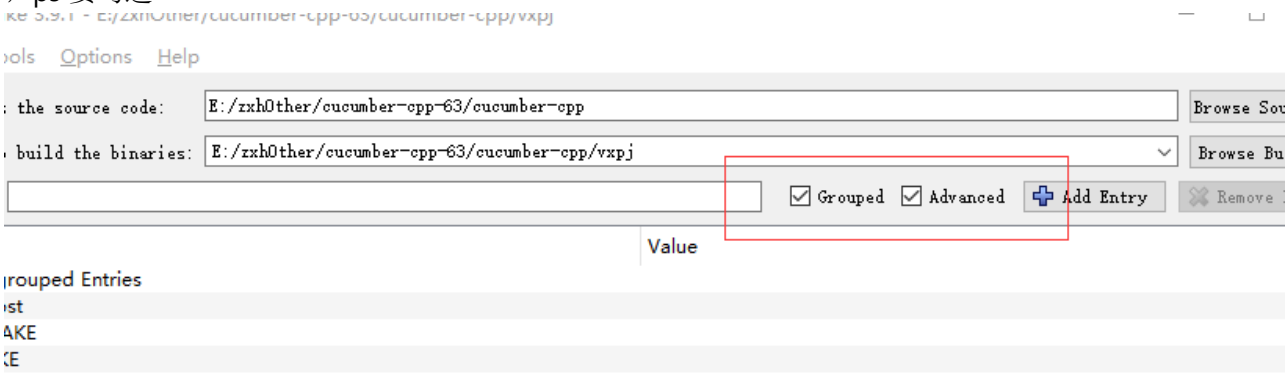
1. 到这里的话你已经成功一半了，我们允许下载好的cmake.exe，把cucumber-cpp下的cmakelist.txt 文件拖入cmake.exe 内，



，指定一下你自己的输出目录



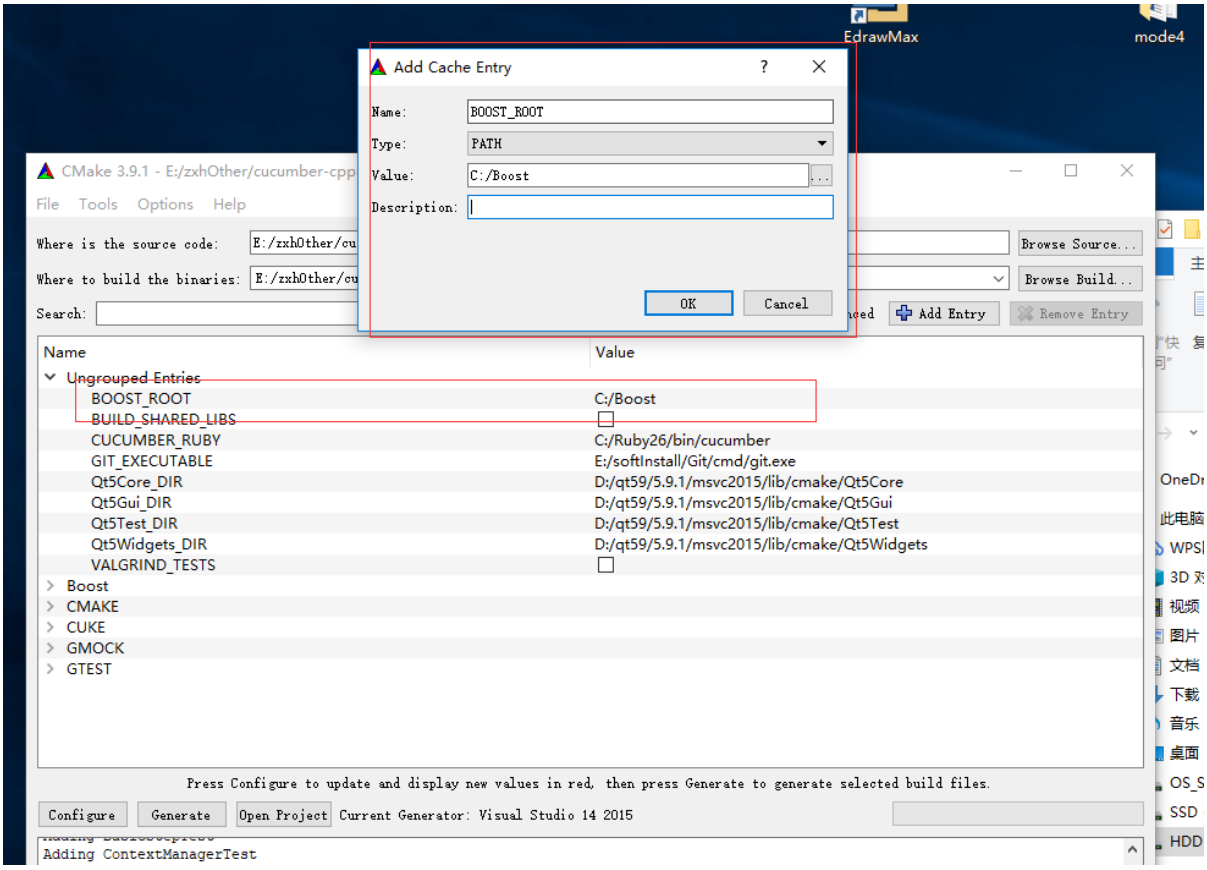
，ps 要勾选



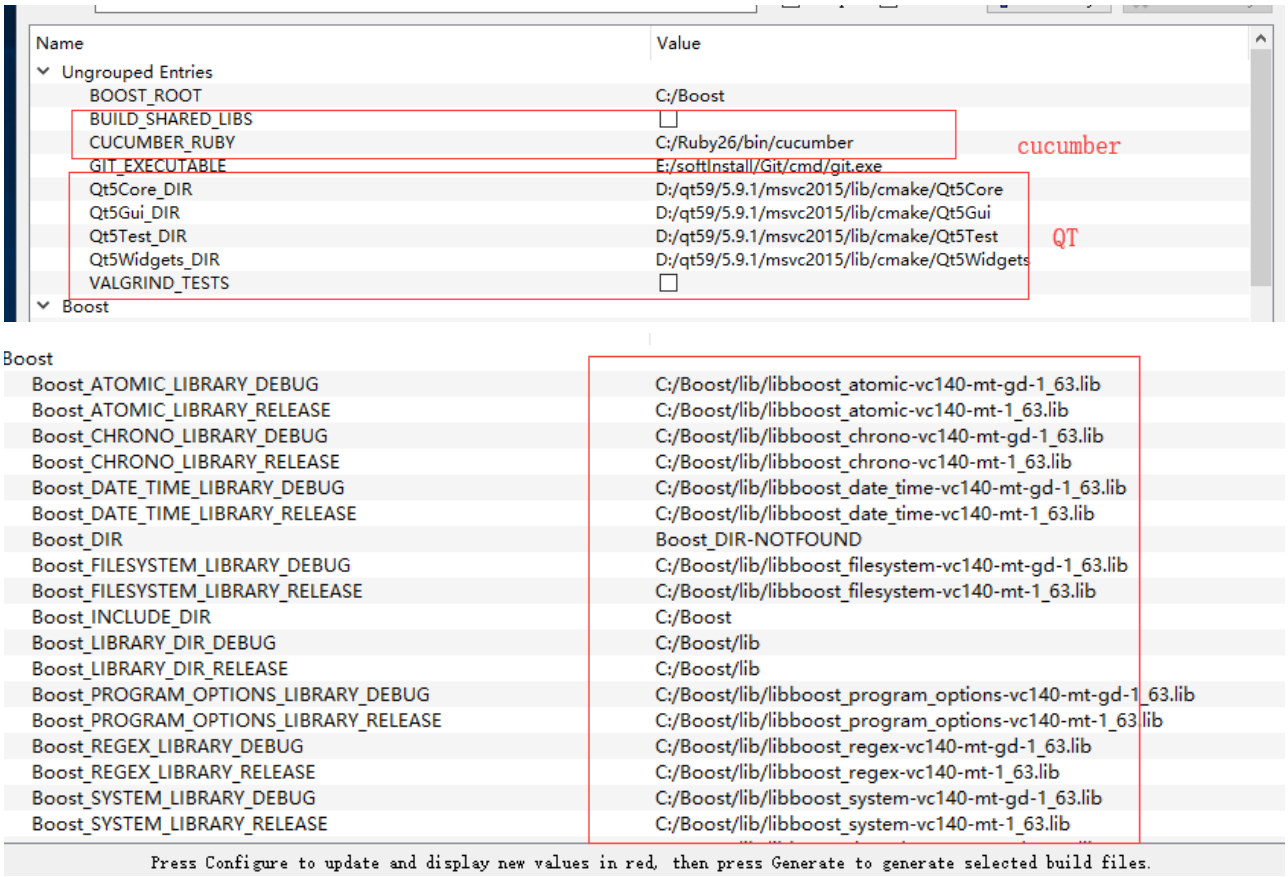
这个哟！

2. 接下来**非常重要，非常重要，非常重要**，很多人在这里都会浪费很多时间。

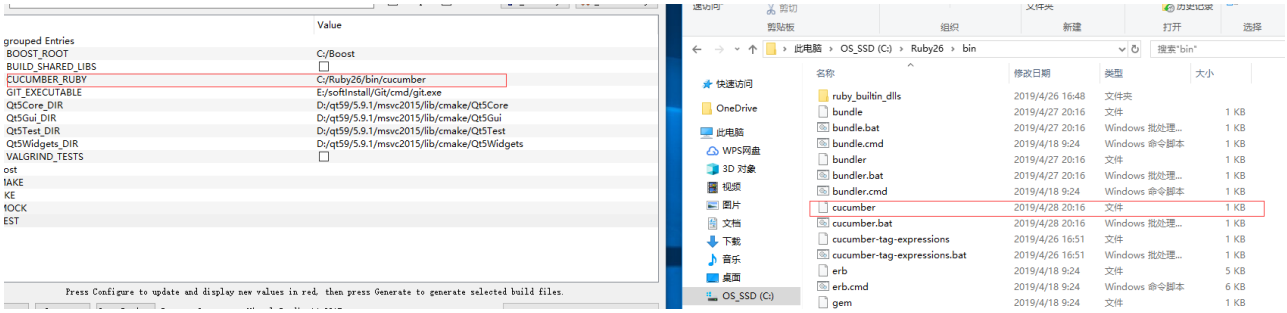
1. 配置BOOST_ROOT 点击 add Entry 添加编译的关键字。



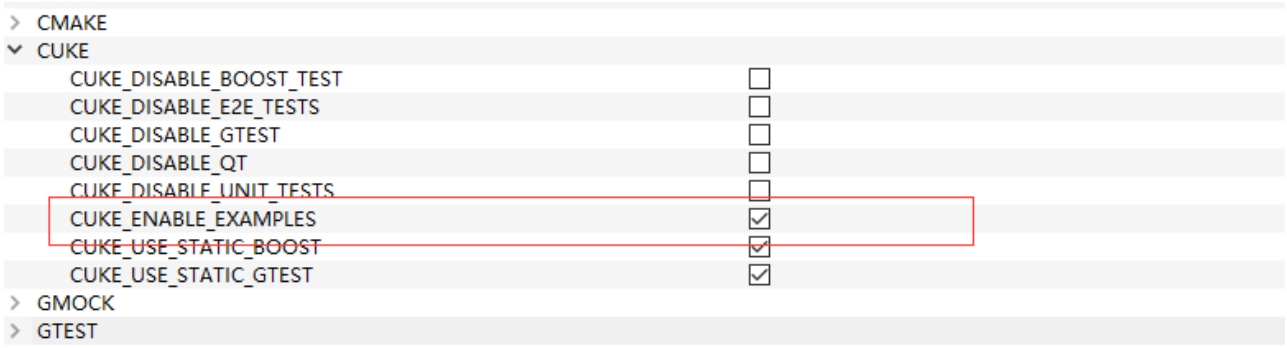
3. 点击Configure，此刻cmake就会去连接刚刚配置的Boost库，如果你正确的安装了qt 以及cucumber 配置结束后你应该能看到如下配置



4. ps.如果cucumber_ruby没找到 但是你有正确安装了程序的话可以手动选择目录如图



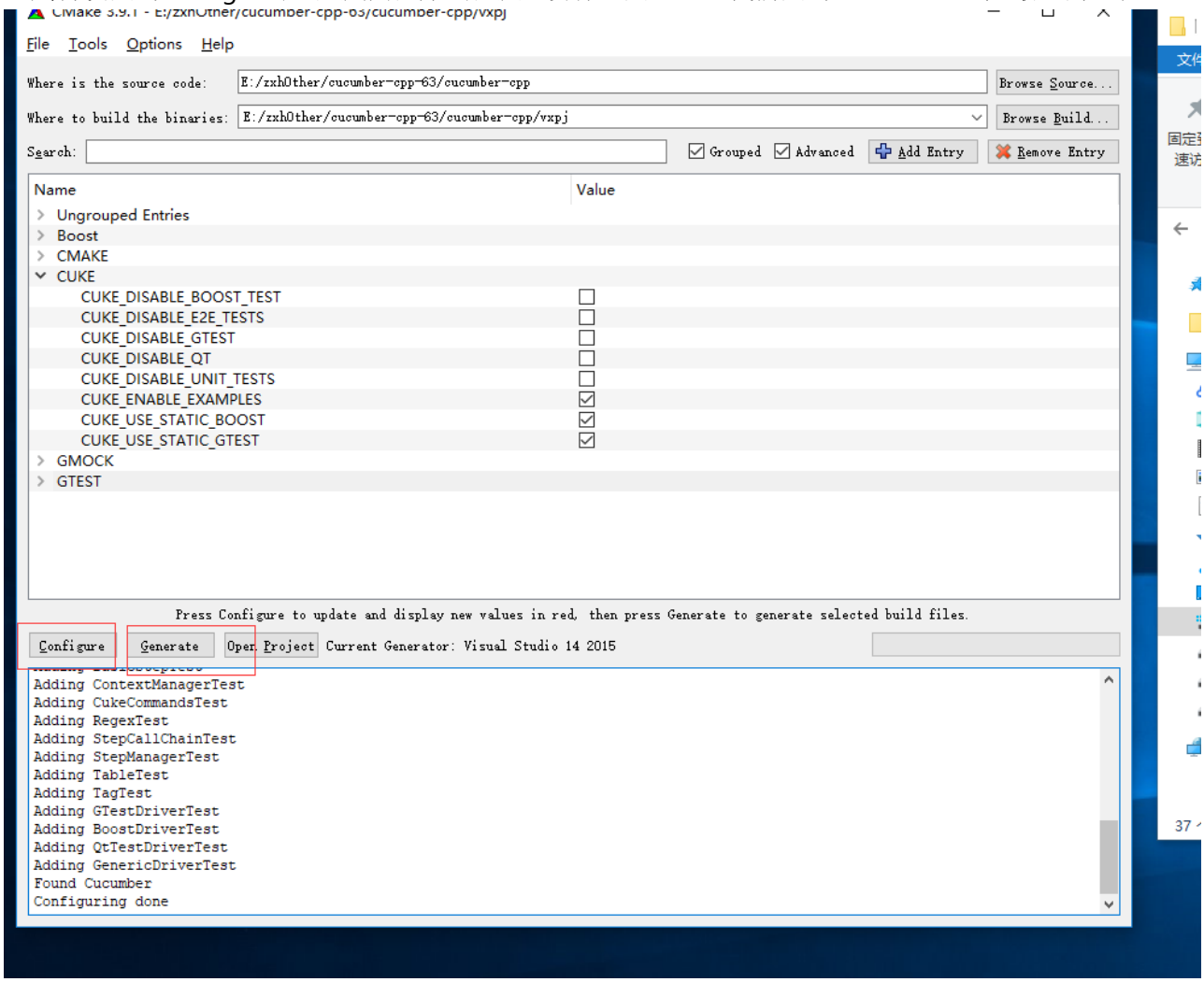
5. 默认是不构建example的 这里要手动打开如图



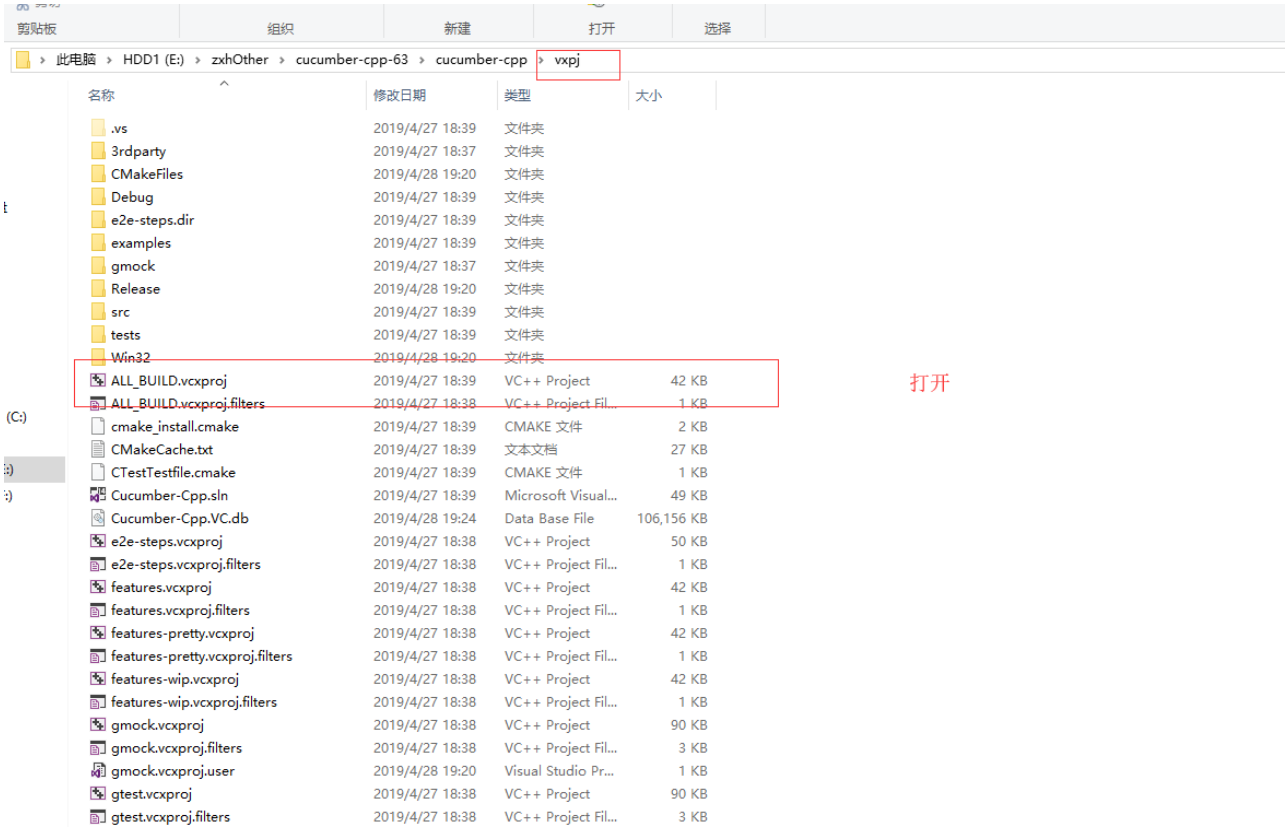
Setp6 构建GTEST GMOCK

1. 这里笔者在连接外部库的时候遇见和qt一样的问题，所以就不自己构建，我们构建cucumber-cpp工程的时候如果发现没有gtest 与 gmock 就会自己去检出对应版本构建，我们采用这种办法。

2. 此刻再次点击Configure 应该就能成功，成就是没有一点红色，我们点击Genernate 生产最终的工程



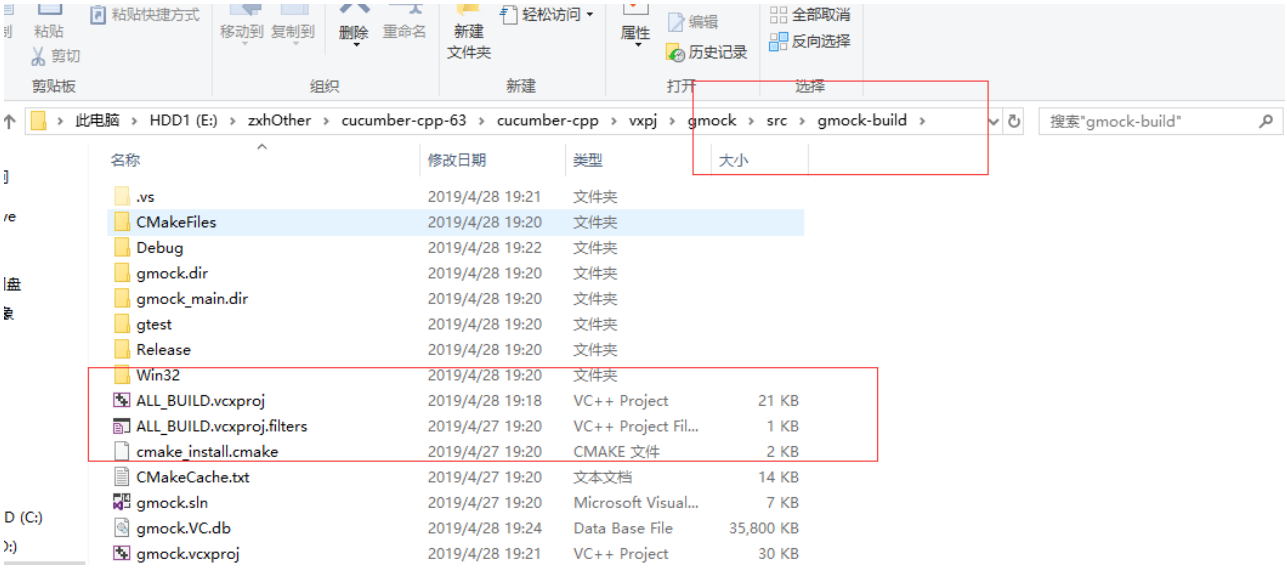
，此刻就生产了最终的工程，如图



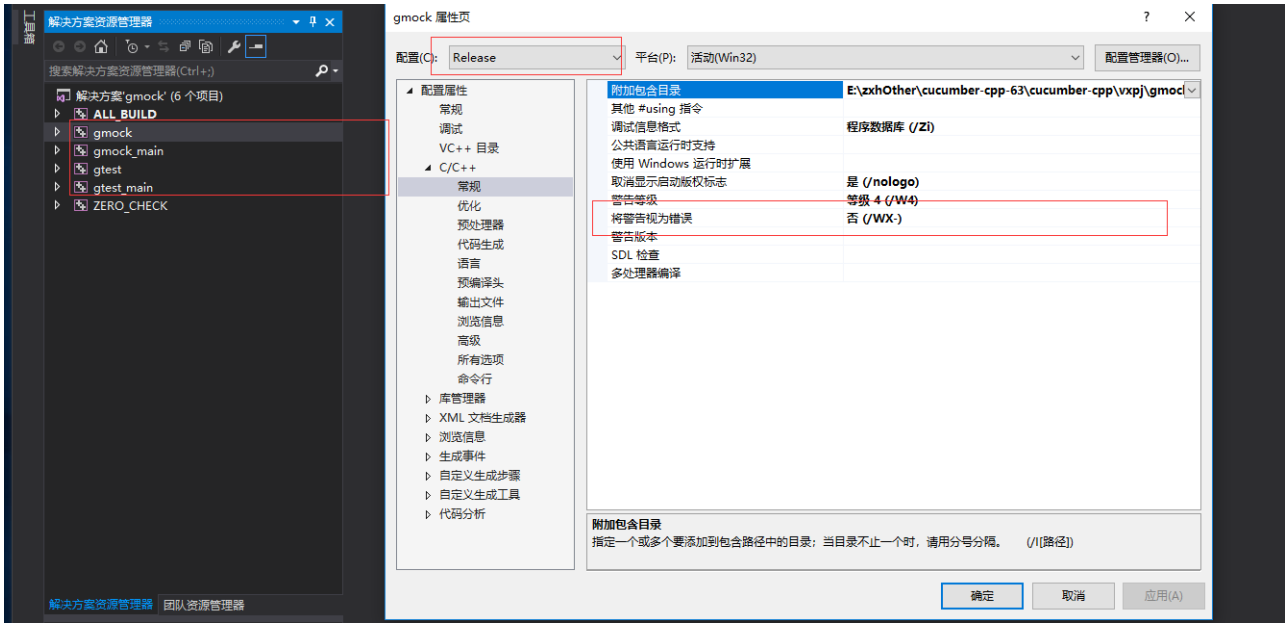
，打开工程

Setp7 编译cucu-cpp源代码

1. 和所有vs工程一样我们只需要选中工程文件，右键 重新生成即可。
2. 这里某些朋友可能会报一些编译错误，类似于 cmd.exe终止，GMock，GTest 等的错误，很有可能是Gtest 于Gmock没有编译过，我找到一种解决办法，我们来到GMOCK的工程



分别右键GMOCK GMOCK_Main 工程

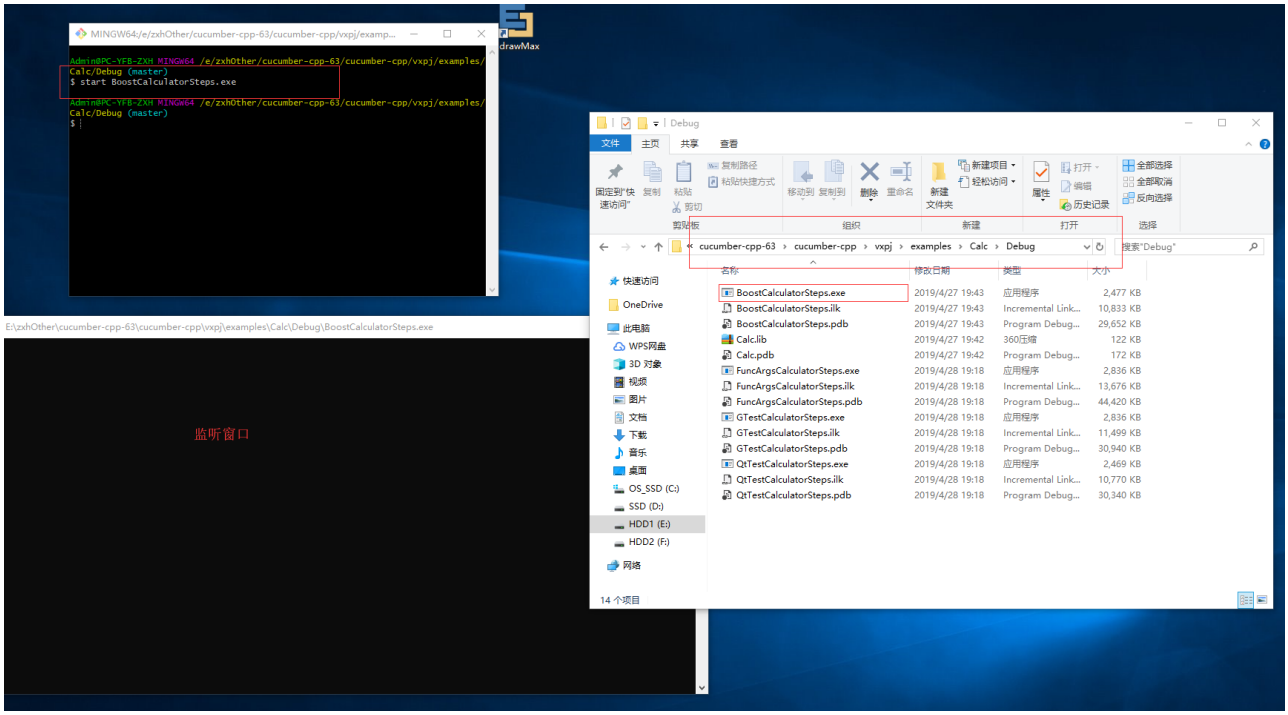


，吧警告视为错误改为否。手动构建这几个工程应该就可以通过，生产对应的lib文件，这个时候在回过头来构建cucumber-cpp的工程就应该可以通过了。到此为只所有源码就都编译完毕了。

Setp8 如何运行example

1. 在笔者实际运行例子的过程中 遇到了很多问题，网上也搜不到结果，笔者经过惨无人道的实测过后终于知道如何运行例子，这里说明一下，避免广大码友在同一条路上翻车。
2. 这里我们来到编译好例子的目录 我这里是cucumber-cpp-63\cucumber-cpp\vxpj\examples\Calc\Debug

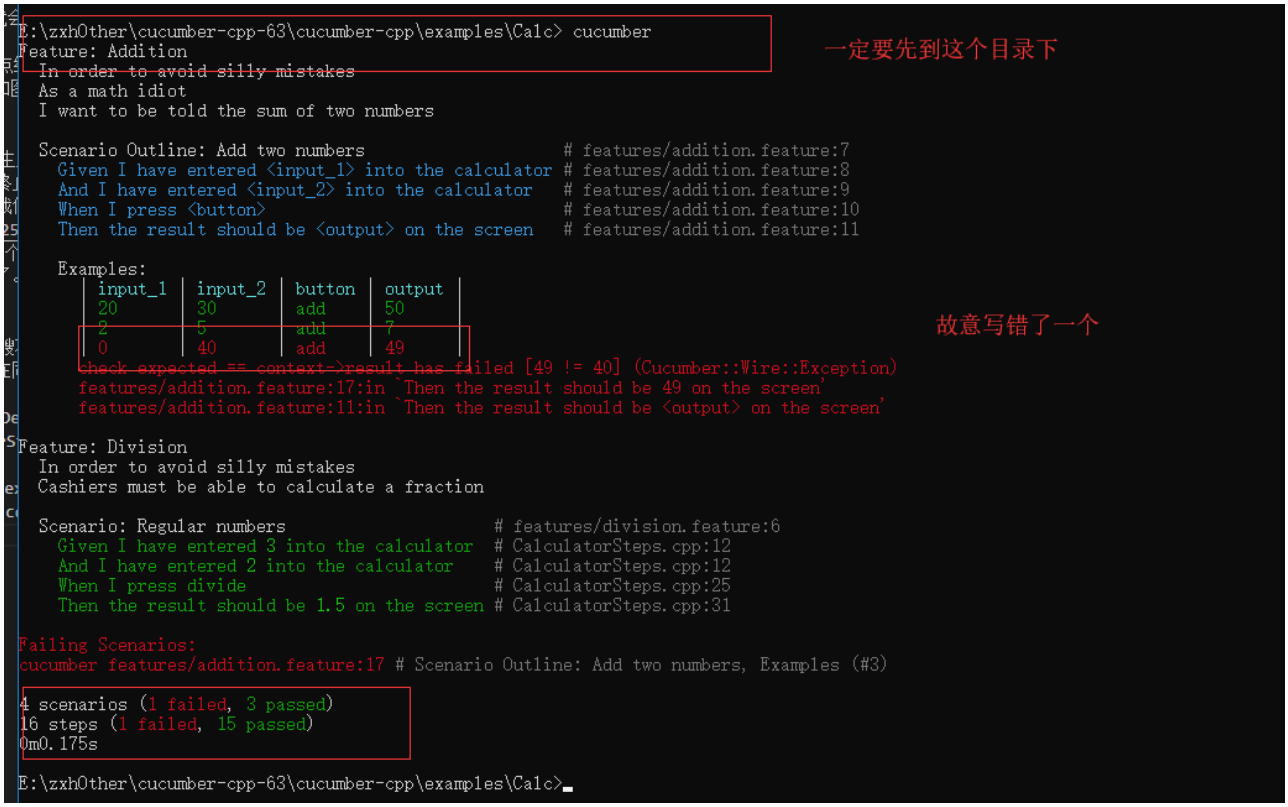
3. 右键 打开git bash输入命令 start BoostCalculatorSteps.exe



,会看到监听端口已经启动起来了。

4. 我们在启动一个cmd.exe 来到BoostCalculatorSteps.exe 对应的源码目录的 我这里是cucumber-cpp-63\cucumber-cpp\examples\Calc 一定是**cd** 到这个目录下 直接跑目录有问题

5. 运行参数 cucumber 即可如图



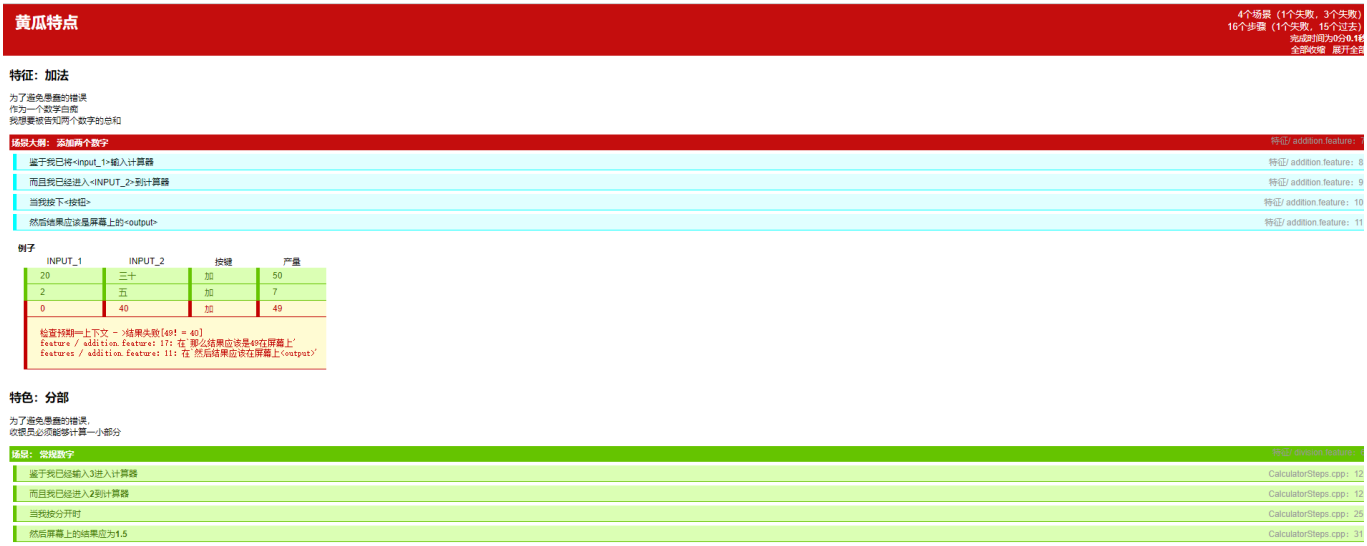
6. ps 说明一下 正常的cmd 是黑白的，笔者这里使用了ansicon.exe 可以支持彩色输出。

Setp9 拓展

1. 这个东西的原理其实是利用cucumber 程序去连接 启动的监听端口传递数据，在控制台输出，cucumber 有很多参数 比如可以输出html 命令如下

```
cucumber --format html --out test.html
```

效果如下



ps.

1. 遗憾:在这次教程里面 为了确保百分百正确，笔者并没有采用 cmakelist 直接连接完成全部的库的这种方式，其实这种方式更好，更佳适合团队合作。
2. cucumber 命令教程
3. cucumber 基本简介