

win10+python3.6+vs2015+boost实现c++封装成python模块

1、why for this?

- 1、记录boost_python的编译过程
- 2、最重要的是记录中间遇到的问题

2、Dependency:

windows10 64位

python3.6 (anaconda3)

VS2015 (V140版本)

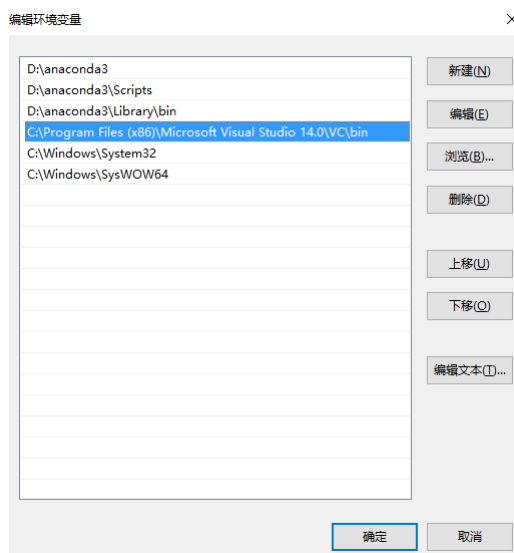
boost 1.69.0

3、preparations

说明：boost官网的第三方lib文件是针对Python2的，没有python3，因此我们需要下载源码用python3进行编译生成。

1. python3的下载安装

tips: 此处如已通过anaconda下载过python，记得把python3添加到系统路径，一般安装时会默认添加。



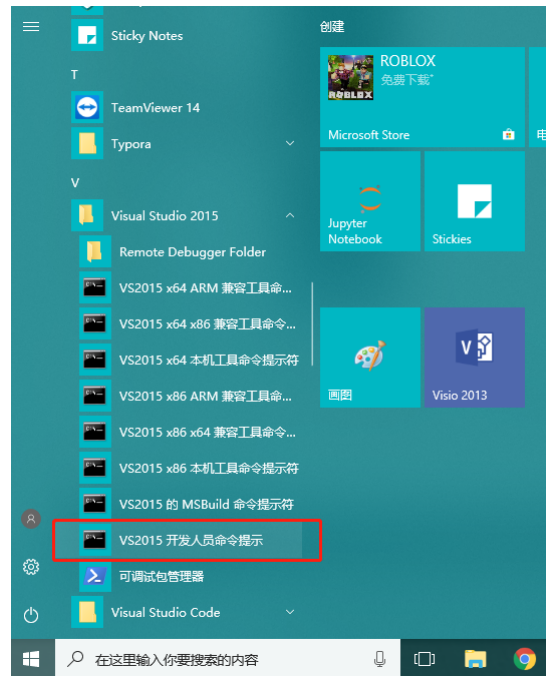
2. boost的下载

进入官网<http://www.boost.org/>，选择最新的版本，这里是1.69.0版本，选择zip文件解压，此处的是D:\boost_1_69_0

3. compile

参考https://blog.csdn.net/qq_30528019/article/details/79278329

- (1) 从开始菜单打开 VS的命令提示符



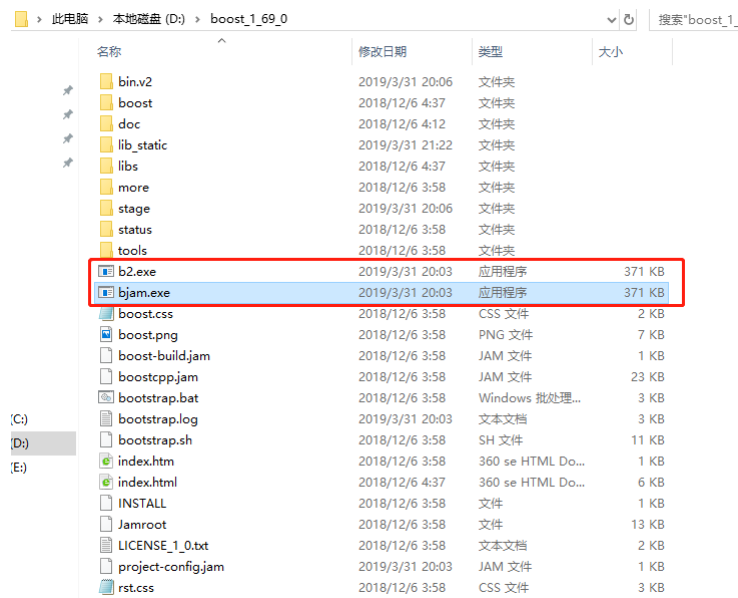
(2) 从终端进入boost库所在目录，如D:\boost_1_69_0，运行bootstrap.bat

windows下切盘符要先直接输入跳转的盘符如： D:

cd 命令切到D:\boost_1_69_0

输入bootstrap，开始Building Boost.Build engine

运行成功在目录下生成exe文件



(3) 进行编译，生成lib文件

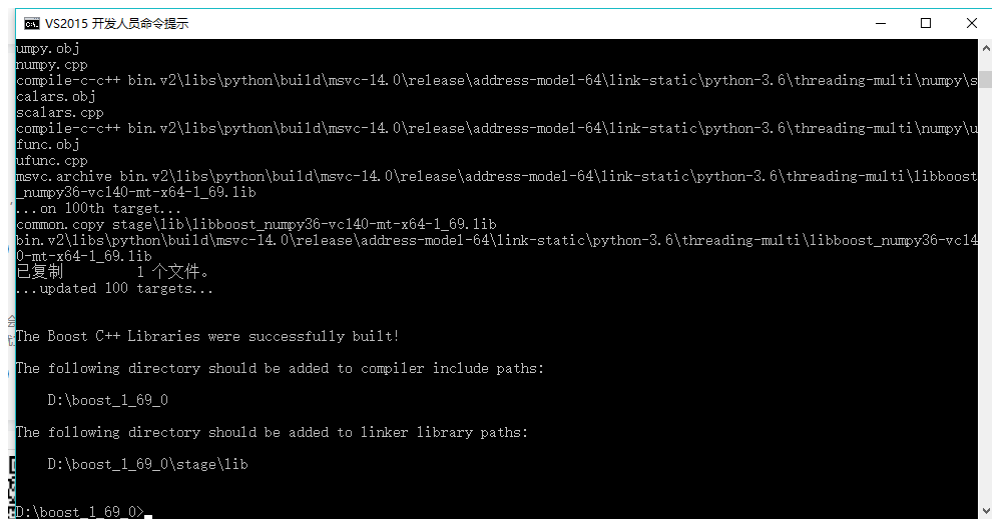
此处方法很多，亲测下面方法有效

静态的库：

直接输入，bjam toolset=msvc-14.0 --with-python threading=multi link=static address-model=64。

(tip: --with-python 里面的python需要是python3版本, 要求系统能找到你的python, 直接在cmd里面输入python能弹出python3说明没有问题, 提示python不是内部或外部命令, 需要将anaconda中的相关路径添加到环境变量, 见3.1节图片; 若提示'cl'不是内部命令, 则是因为vs的cl.exe没有添加到系统变量, 可以将该文件拷贝至系统文件夹下, 或将路径添加至系统环境变量)

成功如下:



```
VS2015 开发人员命令提示
numpy.obj
numpy.cpp
compile-c-c++ bin.v2\libs\python\build\msvc-14.0\release\address-model-64\link-static\python-3.6\threading-multi\numpy\scalars.obj
scalars.cpp
compile-c-c++ bin.v2\libs\python\build\msvc-14.0\release\address-model-64\link-static\python-3.6\threading-multi\numpy\ufunc.obj
ufunc.cpp
msvc.archive bin.v2\libs\python\build\msvc-14.0\release\address-model-64\link-static\python-3.6\threading-multi\libboost_numpy36-vc140-mt-x64-1_69.lib
... on 100th target...
common.copy stage\lib\libboost_numpy36-vc140-mt-x64-1_69.lib
bin.v2\libs\python\build\msvc-14.0\release\address-model-64\link-static\python-3.6\threading-multi\libboost_numpy36-vc140-mt-x64-1_69.lib
已复制 1 个文件。
...updated 100 targets...

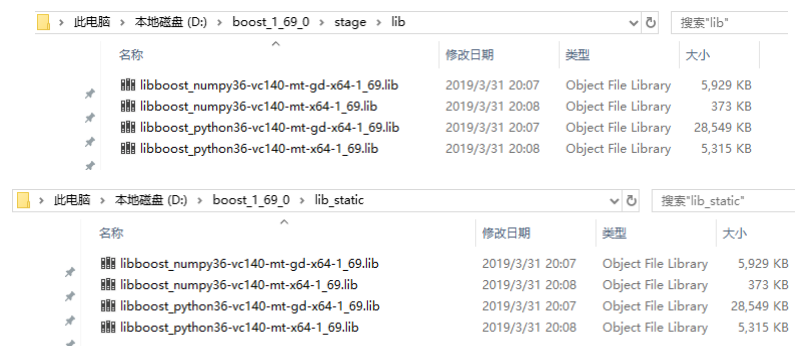
The Boost C++ Libraries were successfully built!

The following directory should be added to compiler include paths:
D:\boost_1_69_0

The following directory should be added to linker library paths:
D:\boost_1_69_0\stage\lib

D:\boost_1_69_0>
```

在boost目录下会生成stage文件夹, 里面的lib就是C++所需的python3的lib文件。可以将Lib复制出来, 放到指定的文件夹, 我这里是把static的库放到新建的lib_static文件夹下, 将用于后续的配置。



名称	修改日期	类型	大小
libboost_numpy36-vc140-mt-gd-x64-1_69.lib	2019/3/31 20:07	Object File Library	5,929 KB
libboost_numpy36-vc140-mt-x64-1_69.lib	2019/3/31 20:08	Object File Library	373 KB
libboost_python36-vc140-mt-gd-x64-1_69.lib	2019/3/31 20:07	Object File Library	28,549 KB
libboost_python36-vc140-mt-x64-1_69.lib	2019/3/31 20:08	Object File Library	5,315 KB

名称	修改日期	类型	大小
libboost_numpy36-vc140-mt-gd-x64-1_69.lib	2019/3/31 20:07	Object File Library	5,929 KB
libboost_numpy36-vc140-mt-x64-1_69.lib	2019/3/31 20:08	Object File Library	373 KB
libboost_python36-vc140-mt-gd-x64-1_69.lib	2019/3/31 20:07	Object File Library	28,549 KB
libboost_python36-vc140-mt-x64-1_69.lib	2019/3/31 20:08	Object File Library	5,315 KB

4、let's do it!

(1) VS 工程配置和编译

1. 分别包含boost 和python3对应的include和lib.

(1) 配置包含目录

D:\boost_1_69_0

D:\anaconda3\include

(2) 配置库目录

D:\boost_1_69_0\lib_static

D:\anaconda3\libs

2. 代码设置和编译成dll

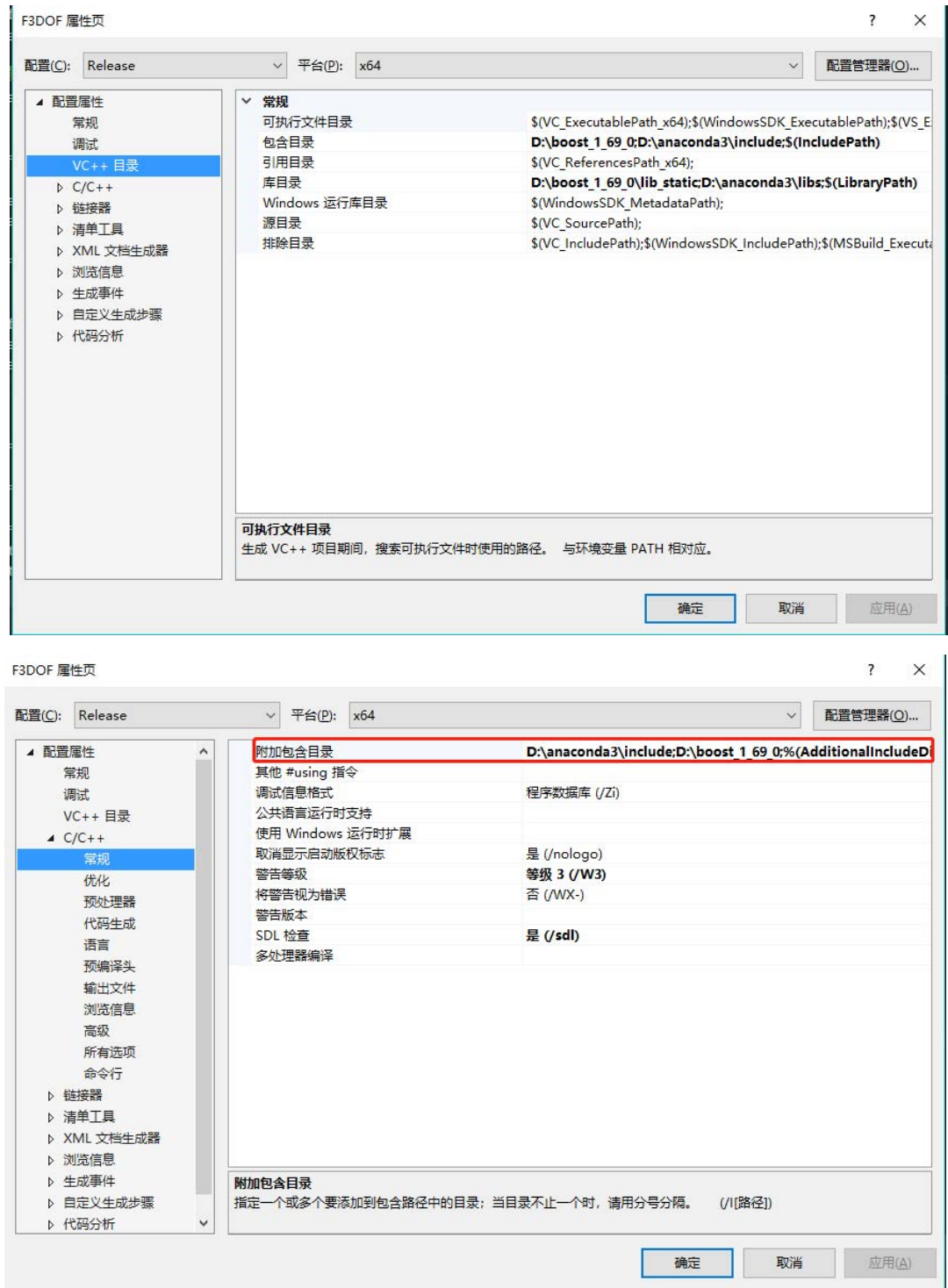
(1) 由于使用的是静态库, 首先需要在你的c++文件前加上下面这句: #define BOOST_PYTHON_STATIC_LIB

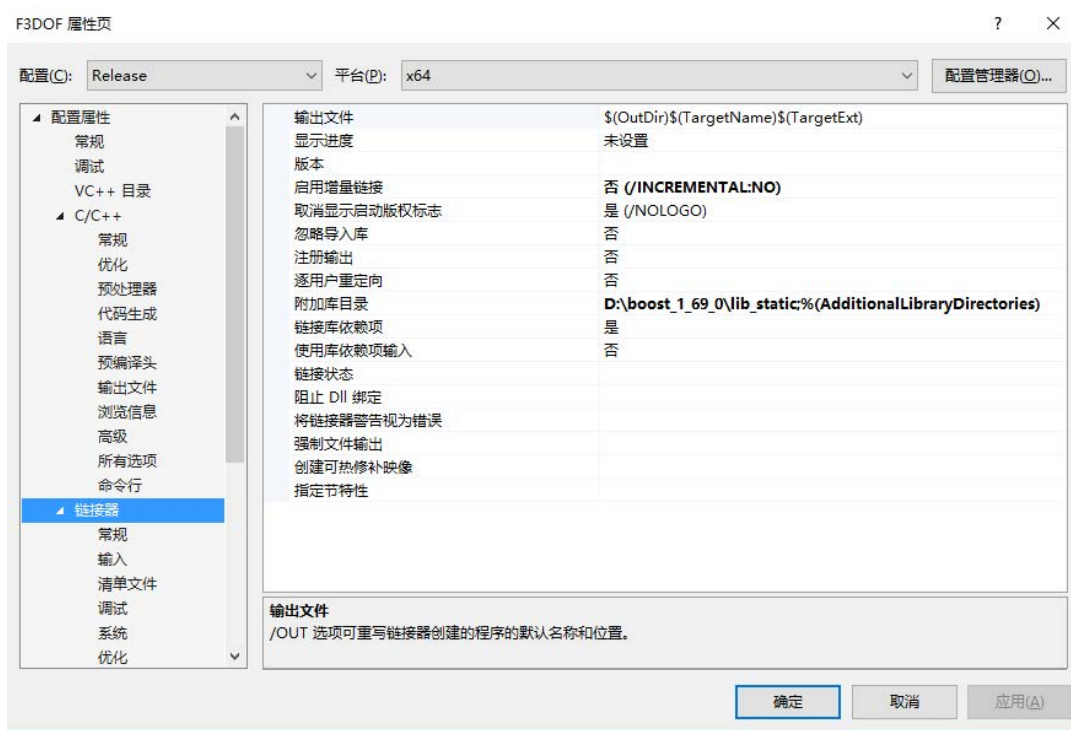
tips: 此处是指定静态库的, 位置放置不正确就会出现LNK error: 找不到lib文件

要将该句添加到预编译头里 (stdafx.h) , 或许可以设置不使用预编译头, 直接添加到cpp文件里

- (2) 并在头文件中并包含和命名空间
- (3) 封装类和函数
- (4) 生成的dll文件改后缀名为.pyd即可放到python项目文件夹内调用

以个人配置为例:





tips:出现有提示命令的错误,可以尝试添加命令到设置中,如下图所示;对于vs出现的格式问题,可以用notepad++将所有文件转换成带BOM的UTF8格式。

