

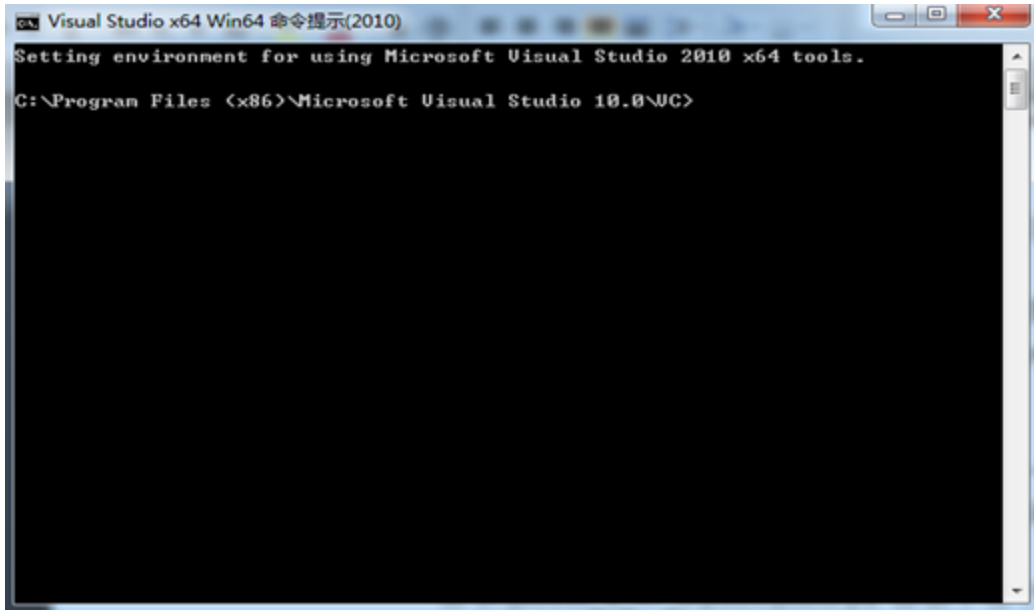
vs2010 x64 下 boost Python 编译

(1)、安装好 Python， 下载 boost 的源码，解压到一个目录下，例如 d:\D:\boost_1_47_0

(2)、按照网上很多人说的，首先编译 boost 的编译器 bjam.exe， bjam 的源码已经被包含在 boost 的源码中，

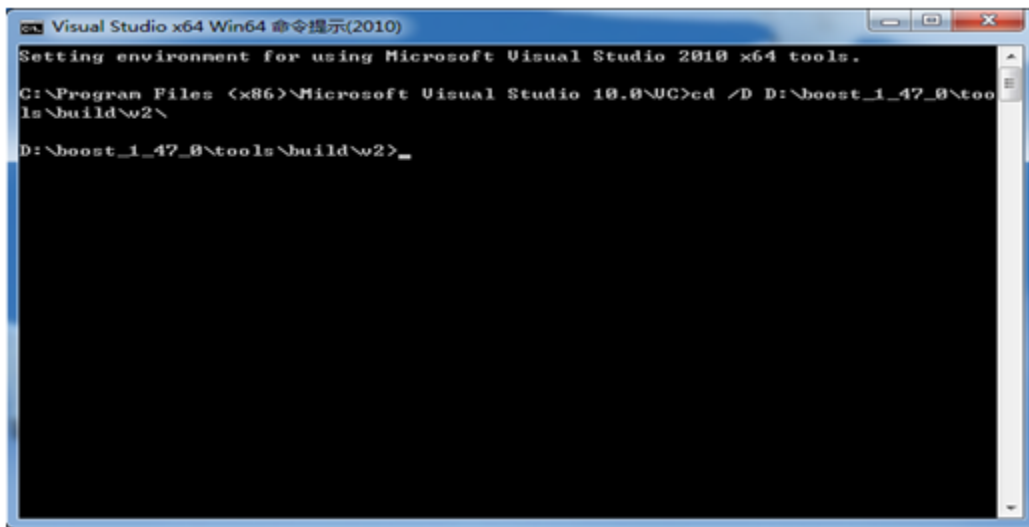
在 D:\boost_1_47_0\tools\build\v2 目录下有个 bootstrap.bat，

运行 vs 2010 x64 的命令提示符工具，如图：



执行命令：cd /D D:\boost_1_47_0\tools\build\v2\

将命令提示符窗口定位到 D:\boost_1_47_0\tools\build\v2 目录下，



执行命令：bootstrap.bat 开始 bjam 的编译，



```
Visual Studio x64 Win64 命令提示(2010)
Setting environment for using Microsoft Visual Studio 2010 x64 tools.
C:\Program Files (x86)\Microsoft Visual Studio 10.0\VC>cd /D D:\boost_1_47_0\tools\build\w2\
D:\boost_1_47_0\tools\build\w2>bootstrap.bat
Bootstrapping the build engine
Bootstrapping is done. To build, run:
    .\b2 --prefix=DIR install
D:\boost_1_47_0\tools\build\w2>
```

编译 bjam 完成后如上图所示

在 D:\boost_1_47_0\tools\build\w2\engine\bin.ntx86_64 目录下找到 bjam.exe ,

(3)、修改 bjam 的配置文件, 在 D:\boost_1_47_0\tools\build\w2 用文本编辑器打开 user-config.jam 这个文件,

修改其中的设置, 将

Configure specific msvc version (searched for in standard locations and PATH).

using msvc : 8.0 ;

改为 (因为这里用的是 vs 2010) :

Configure specific msvc version (searched for in standard locations and PATH).

using msvc : 10.0 ;

将:

Python configuration.

Configure specific Python version.

using python : 3.1 : /usr/bin/python3 : /usr/include/python3.1 : /usr/lib ;

改为 (python2.7 的安装目录和 python 版本设置) 设置 python 的目录和版本等信息:

Python configuration.

Configure specific Python version.

using python : 2.7 : C:\\Python27 : C:\\Python27\\include : C:\\Python27\\libs ;

(4)、将刚才的 bjam.exe 拷贝到 D:\boost_1_47_0\ 下, 准备 boost 库的编译。

将命令提示符定位到 D:\boost_1_47_0\ 下

执行 bjam 编译命令:

能正确生成 动态库版本的命令写法:

x64 debug

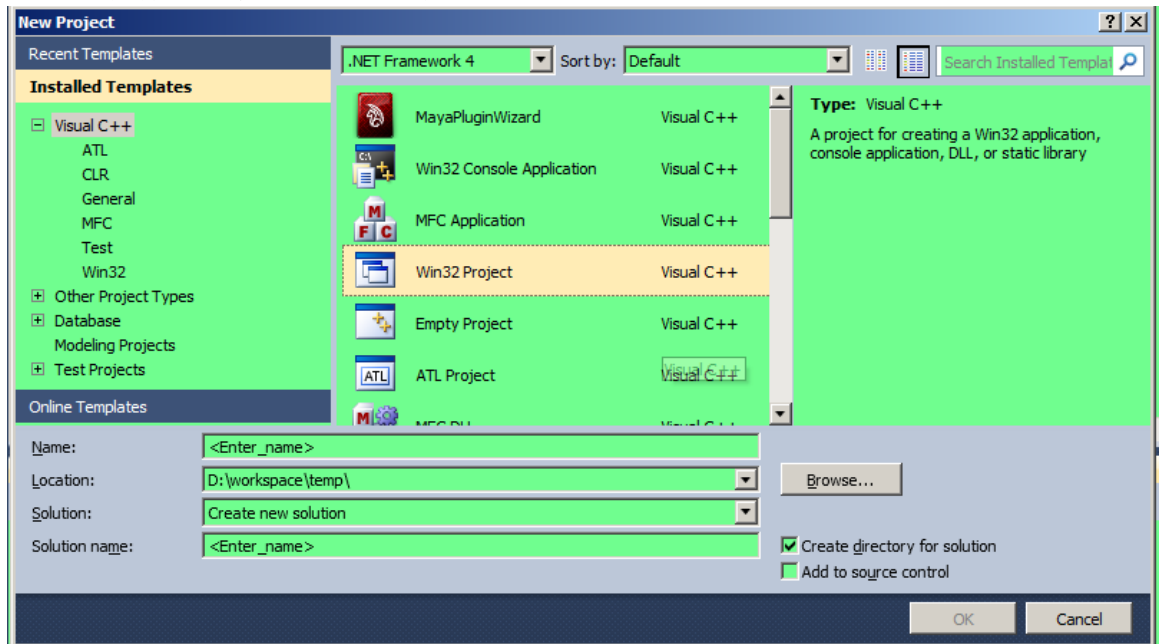
**bjam --with-python --prefix=d:\boost stage toolset=msvc-10.0 variant=debug
link=shared address-model=64 threading=multi runtime-link=shared install**

x64 release

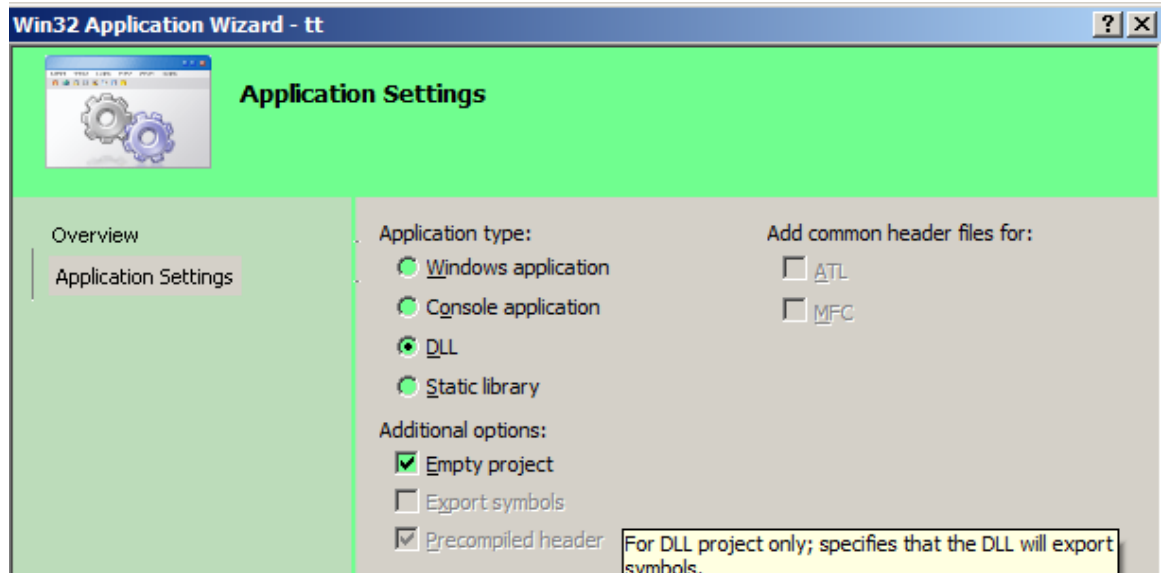
**bjam --with-python --prefix=d:\boost stage toolset=msvc-10.0 variant=release
link=shared address-model=64 threading=multi runtime-link=shared install**

Python 中如何调用 C++写的扩展模块

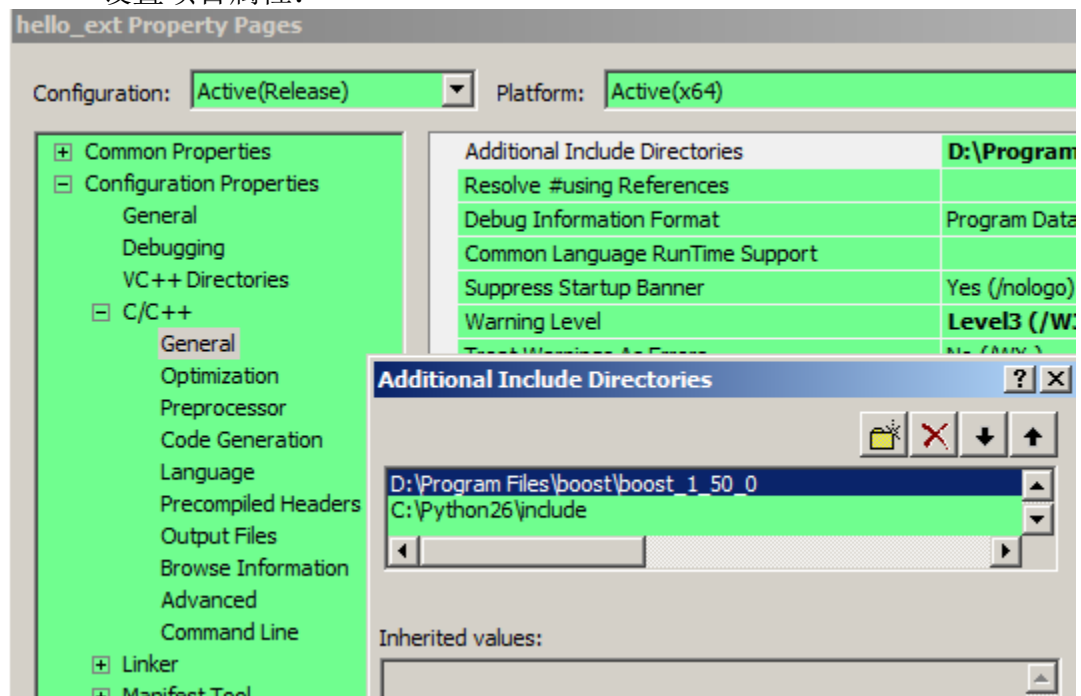
1、 vs2010 中新建一个空的 DLL 工程:

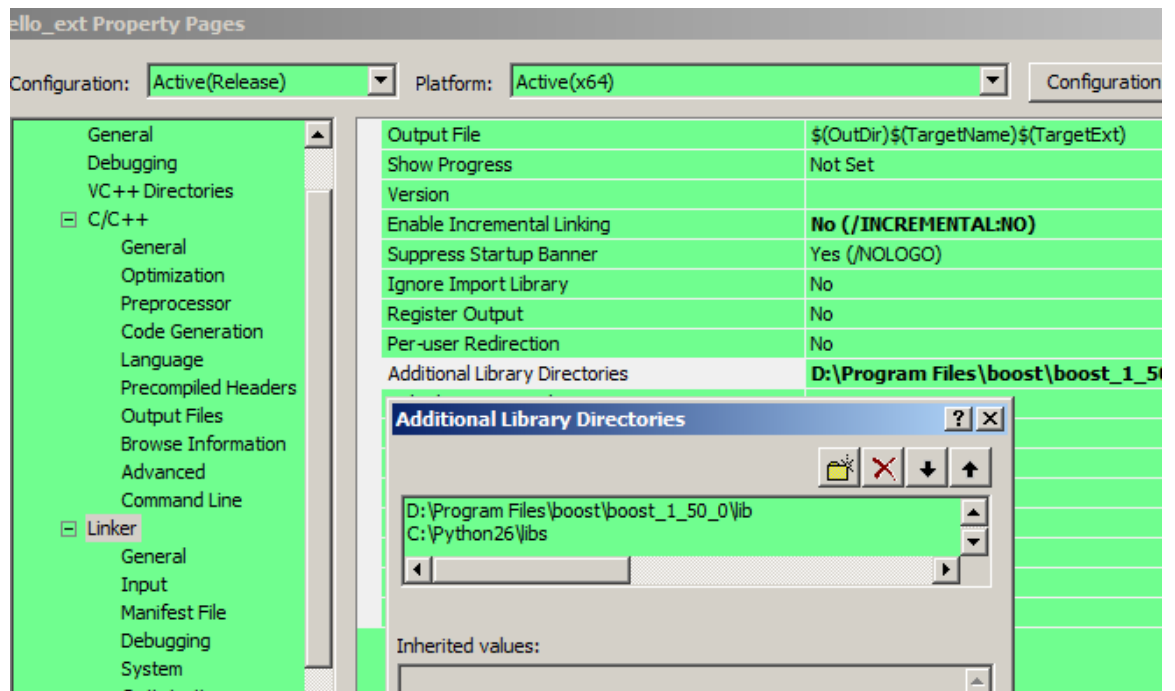


a.



2、
b. 设置项目属性：





新建 hello.cpp 文件,把下面代码拷进去:

```
#include <boost/python/module.hpp>
#include <boost/python/def.hpp>

char const* greet()
{
    return "hello, world";
}

BOOST_PYTHON_MODULE(hello_ext)
{
    using namespace boost::python;
    def("greet", greet);
}
```

8. 编译,生成,把输出的 dll 改名为 hello_ext.pyd,

把 hello_ext.pyd 和 boost_python-vc90-mt-gd-1_48.dll
拷贝到 python 的工作目录下

在 python 工作目录下新建 hello.py 编写如下代码:

```
import hello_ext
```

```
hello_ext.greet()
```

