来源：[CSDN地址](https://blog.csdn.net/bailang_zhizun/article/details/120992244?spm=1001.2101.3001.6650.5&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-5-120992244-blog-88393787.pc_relevant_antiscanv2&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-5-120992244-blog-88393787.pc_relevant_antiscanv2&utm_relevant_index=10)

https://blog.csdn.net/bailang\_zhizun/article/details/120992244?spm=1001.2101.3001.6650.5&utm\_medium=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-5-120992244-blog-88393787.pc\_relevant\_antiscanv2&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-2%7Edefault%7EBlogCommendFromBaidu%7ERate-5-120992244-blog-88393787.pc\_relevant\_antiscanv2&utm\_relevant\_index=10

**OSG3.6.5+vs2017+win10\_x64**

---------------------------------------start------------------------------------

最近开始了OSG的学习，开始搭建OSG开发环境，期间遇到了很多问题，也查阅了很多资料，最终搭建成功。OSG本来就是开源的，不应该敝帚自珍，应该多交流多分享，所以在此记录一下环境搭建的过程，方便以后查阅和交流。

1、下载文件

1.1、OSG源码下载

本仓库

1.2、OSG第三方依赖库下载

下载地址：[Dependencies](http://www.openscenegraph.org/index.php/download-section/dependencies)（zipSource内）

里面有各种版本的第三方依赖库，下载自己需要的就可以了，但是必须和自己的编译器版本保持一致。我下载的是2017版本的 3rdParty\_VS2017\_v141\_x64\_V11\_full.7z。

1.3、数据包下载（zipSource内）

下载地址：[Data Resource](http://www.openscenegraph.org/index.php/download-section/data)

下载的是 OpenSceneGraph-Data-3.4.0.zip

1.4、编译工具下载

编译工具选择的是CMake，官网上推荐2.6.x：

直接下载最新版的CMake，下载地址：[Download | CMake](https://cmake.org/download/)

下载的是 cmake-3.21.3-windows-x86\_64.msi

2、CMake构建

2.1、构建前的准备

为了方便编译，在硬盘下建立一个OSG文件夹，然后分别建立如下3个文件夹：

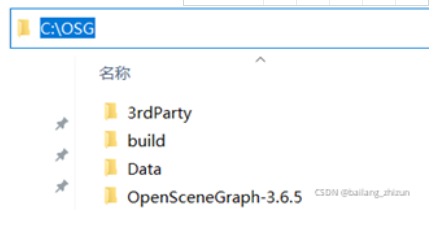
OpenSceneGraph-3.6.5: 用于存放解压后的 OpenSceneGraph-OpenSceneGraph-3.6.5.zip 文件

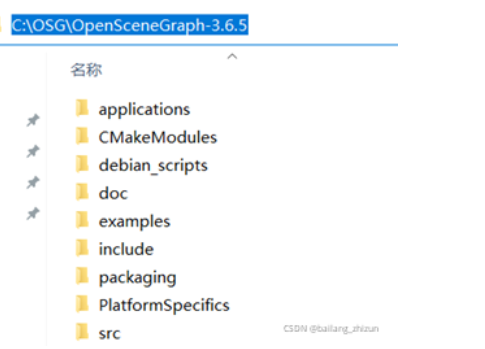
3rdParty: 用于存放解压后的 3rdParty\_VS2017\_v141\_x64\_V11\_full.7z 文件

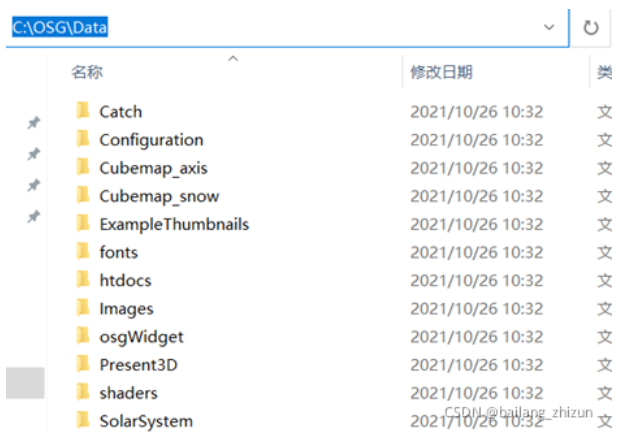
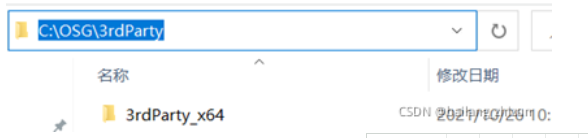
Data: 用于存放解压后的 OpenSceneGraph-Data-3.4.0.zip

build: 用于存放编译后的文件

如下图所示：



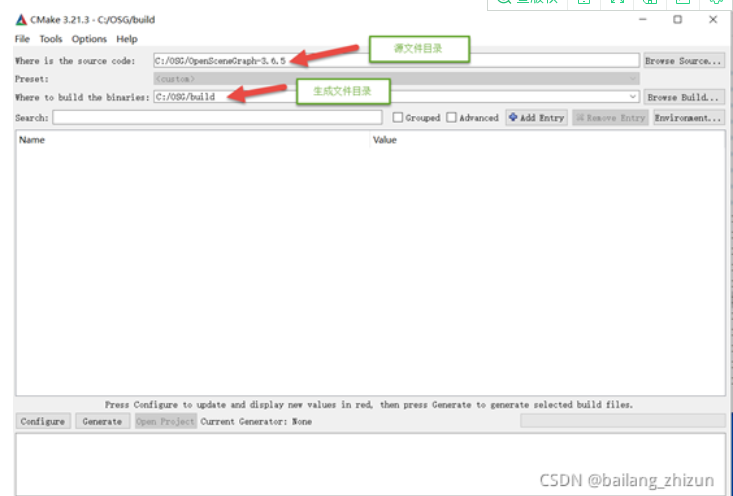




安装CMake。默认安装即可。

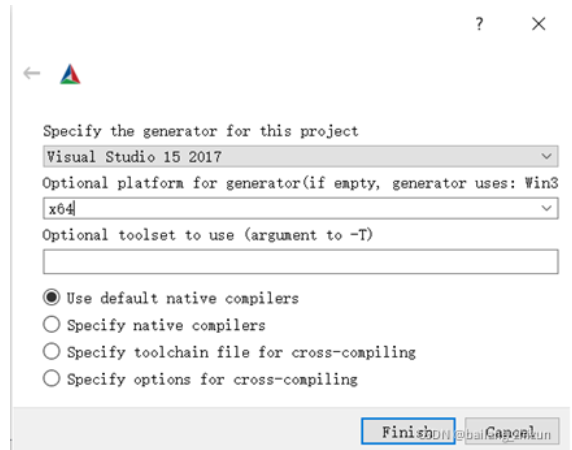
2.2、开始构建

打开CMake，配置源文件目录和编译后的生成目录，如下所示：

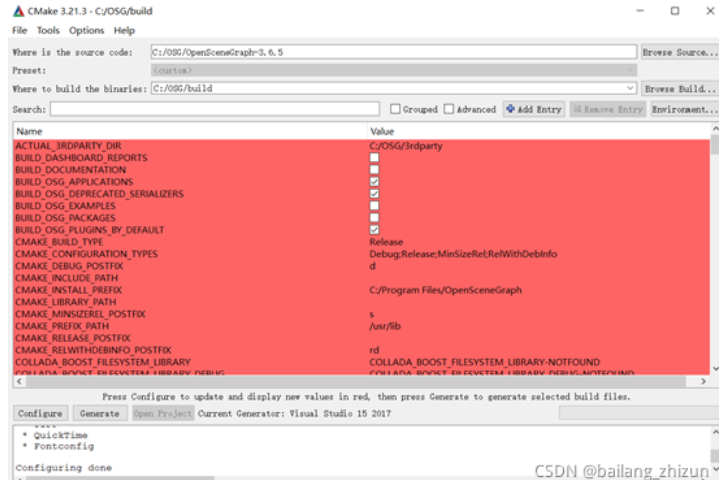


（也可以直接将OpenSceneGraph-3.6.5下的CMakeLists.txt文件直接拖放到CMake界面，然后在重新配置一下build文件目录就可以了）

点击“Configure”按钮，按如下所示进行配置：



点击“Finish”按钮：



修改 ACTUAL\_3RDPARTY\_DIR 的值为 C:\OSG\3rdParty（前面配置好的3rdParty的目录）：



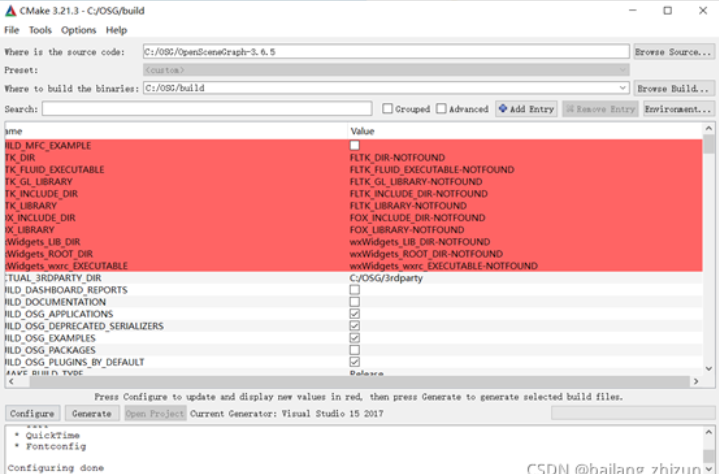
BUILD\_OSG\_EXAMPLES 打钩：



CMAKE\_INSTALL\_PREFIX 的值为上面设置的build的路径：

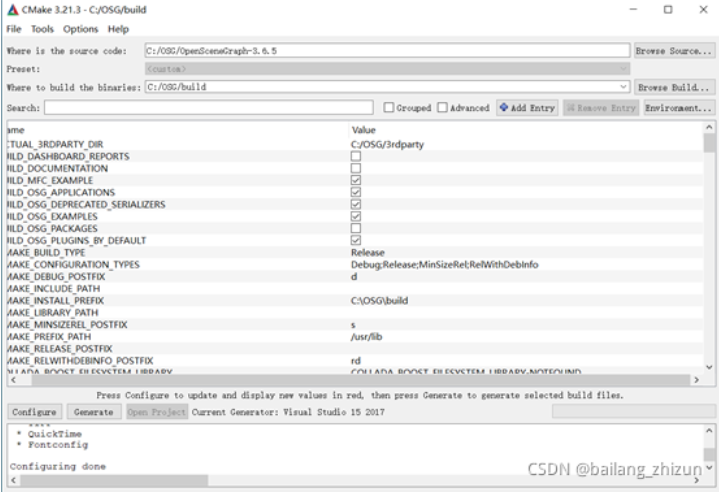


再次点击“Configure”按钮：



还是有红色的选项，BUILD\_MFC\_EXAMPLE 打钩（切记！！！VS2017一定要安装MFC开发组件！！！）

再次点击“Configure”按钮，此时没有红色选项：

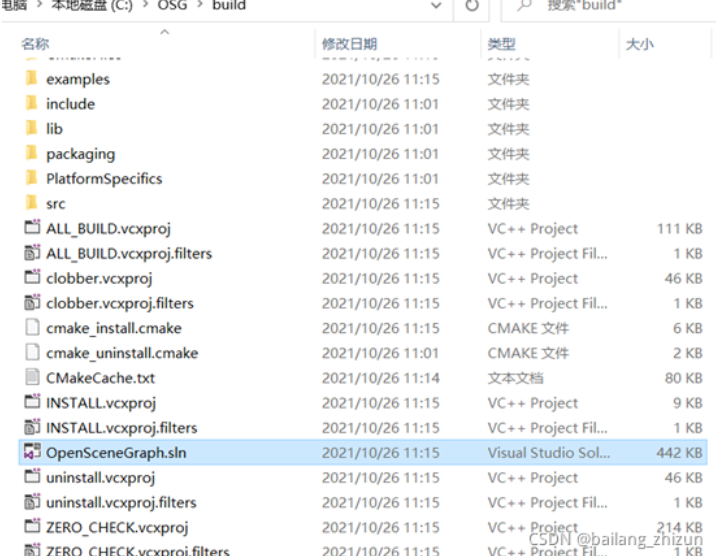


然后点击“Generate”按钮，进行生成。

3、编译

3.1、ALL\_BUILD编译

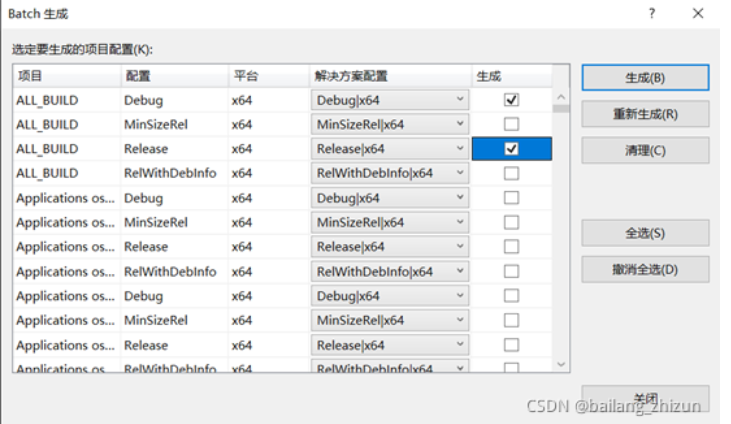
在build目录（）中找到生成的OpenSceneGraph.sln文件，然后使用vs2017打开：



选择生成->批生成：



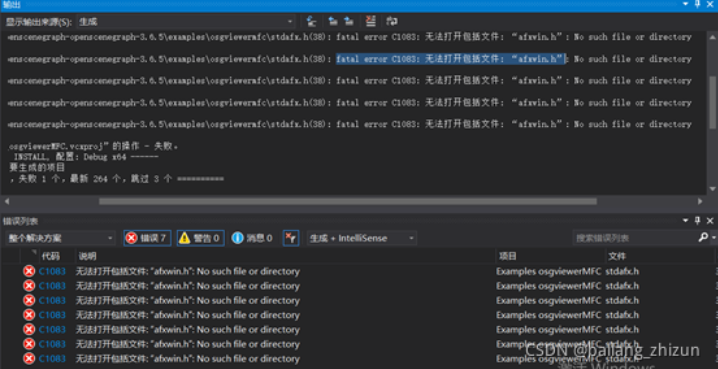
选择 ALL BUILD 的 Debug 和 Release：



然后点击“生成”，此时会花费比较久的时间（我的机器上大概用了3个小时左右）。如果前面配置的没有错误的话，此时应该不会报错。

3.1.1、问题1

在运行过程中提示了以下错误：

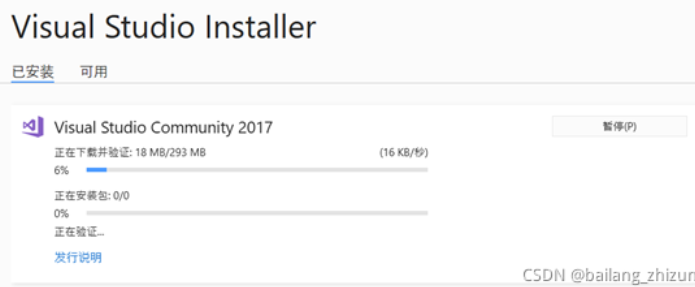


经检查，发现是因为我使用的VS2017版本没有安装MFC组件，所以打开vs2017的安装程序，选择安装MFC组件即可，安装方法：

打开vs2017 community的安装程序：



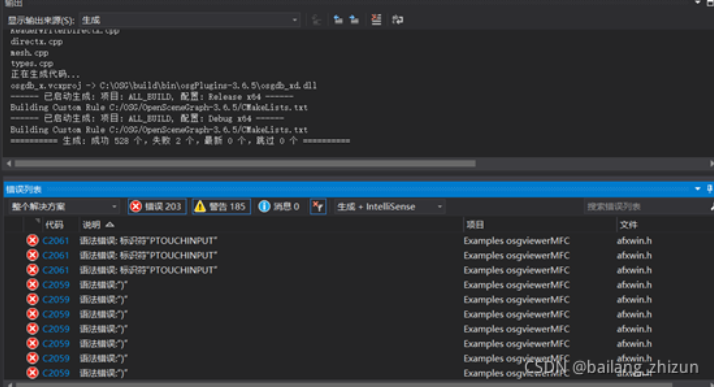
然后点击修改按钮就会开始安装MFC组件：



（20K左右的下载速度，很难受~~）

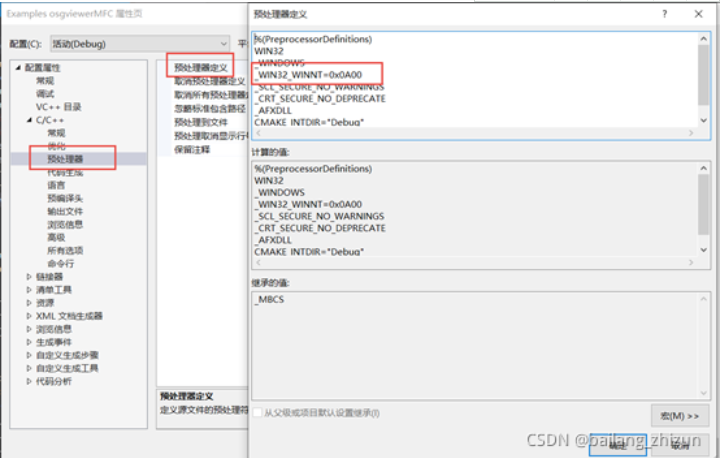
3.1.2、问题2(已经在代码里修正)

安装完成之后，又重新ALL\_BUILD一下，ALL\_BUILD之后出现如下错误：



出现了一大堆的错误，这个时候不要慌，一般这种错误都是因为一个原因，可以看到出现的错误基本上都是在Examples osgviewerMFC工程中，出现问题的文件基本上集中在afxwin.h中，而这个是库文件，一般不会出错。经查，这种问题一般是因为MFC的编译平台的原因。

可以看到该工程的目标编译平台为：

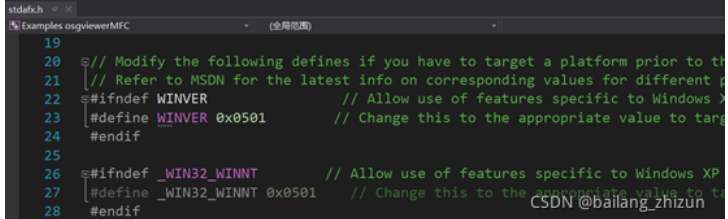


其中的\_WIN32\_WINNT=Ox0A00，表明目标平台是Win10的，对应关系如下所示：

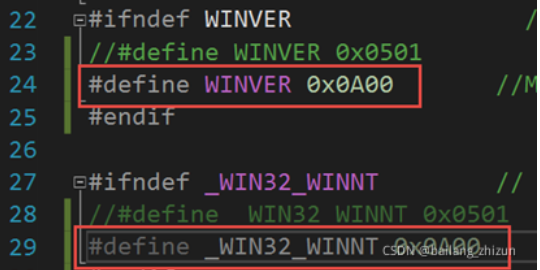


参考链接：使用 Windows 标头 - Win32 apps | Microsoft Docs

而该工程中定义的目标平台为0x0501（该工程中的stdafx.h头文件下定义的）:



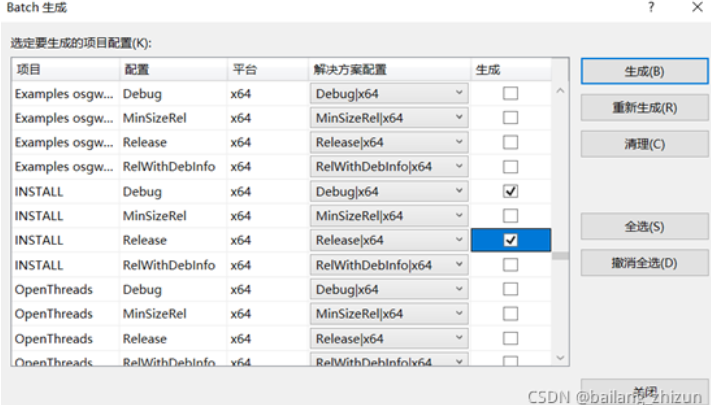
0x0501是面向xp系统的，因为我使用的是Win10操作系统，所以需要进行以下修改：



保存，再重新ALL\_BUILD一下，就没有报错了。

3.2、INSTALL编译

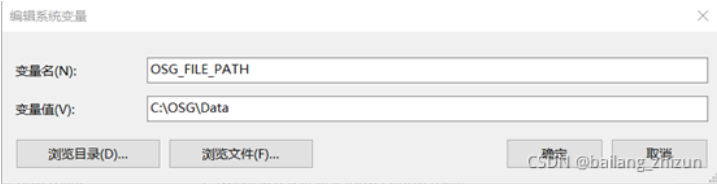
ALL\_BUILD编译完成之后，需要进行INSTALL编译，同样打开生成->批生成，选择INSTALL：



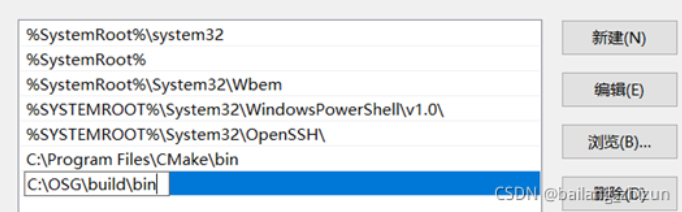
点击“生成”按钮，进行编译生成，这部分会快一些；

4、配置环境变量

添加系统环境变量 OSG\_FILE\_PATH，值为前面的Data的目录（C:\OSG\Data）：



然后在Path变量中添加前面生成目录build目录中的bin目录（C:\OSG\build\bin）：

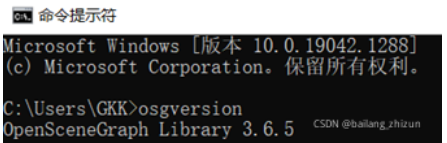


配置完成之后最好重启一下。

5、测试

5.1、命令行测试（解决了两个问题）

在命令行中输入osgversion：



在命令行中输入osglogo：

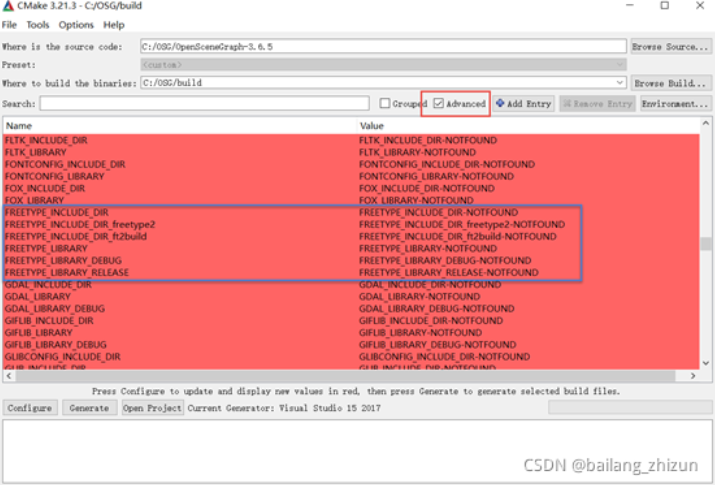


能够进行拖动，缩放等操作，但是图形渲染的有问题。命令行上打印的信息如下：



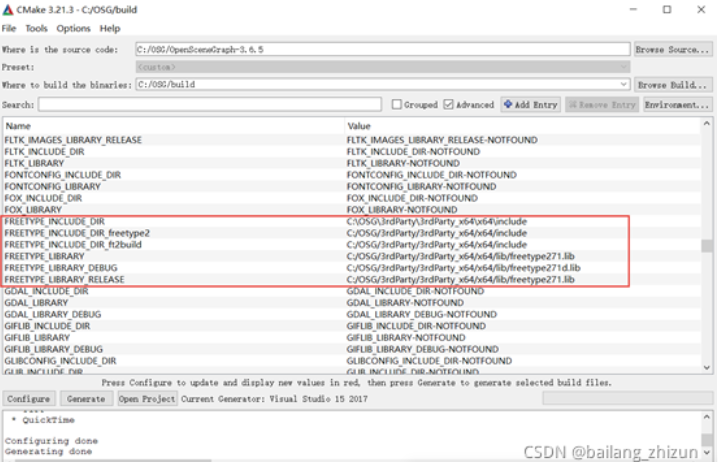
5.1.1、解决arial.ttf: file not handled 的问题

参考链接：[关于在osgearth 中 出现 arial.ttf : file not handled 的问题 - 尚码园](https://www.shangmayuan.com/a/a367ff23dff04a8394ede712.html)

 取消勾选 OSG\_TEXT\_USE\_FONTCONFIG



配置freetype路径:



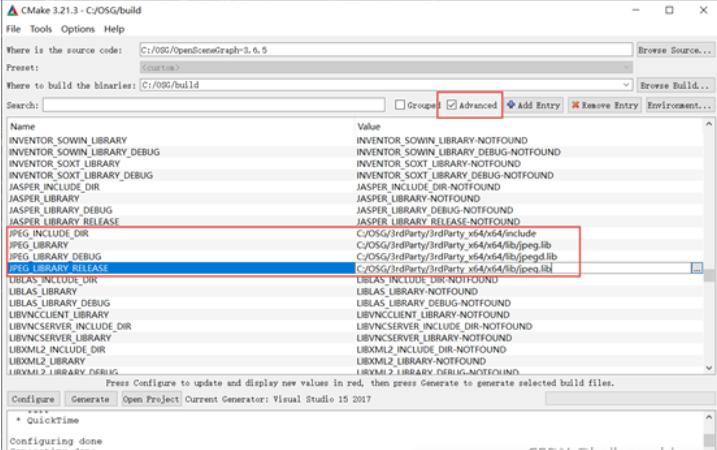
Configure->Generate，然后重新打开 .sln解决方案，生成->批生成，先ALL\_BUILD，然后再 INSTALL。再运行osglogo：



文字能够正常显示了，但是旁边的地球没有显示出来。

5.1.2、解决无法读取.jpg的问题

有了5.1.1的经验，猜测可能是因为缺少了读取.jpg文件的库，所以需要CMake重新配置一下：



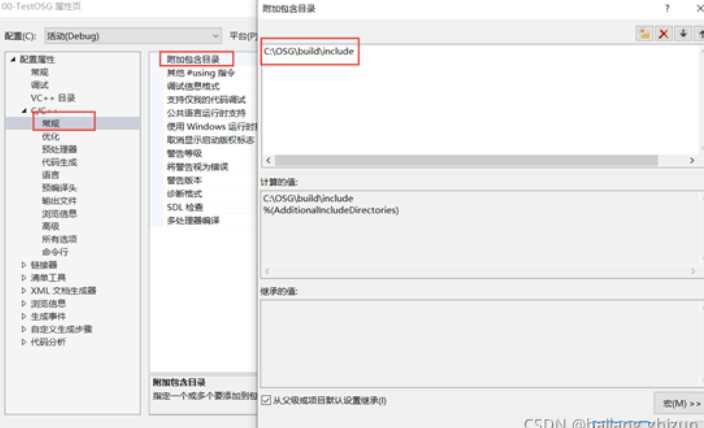
Configure->Generate，然后重新打开 .sln解决方案，生成->批生成，先ALL\_BUILD，然后再 INSTALL，再重新运行osglogo显示如下，问题解决：



而且地球也一直在转。

5.2、新建项目测试

在vs2017中新建C++控制台程序项目，项目属性，选择x64，然后配置头文件目录为build的目录下的include目录，如下所示：



然后配置链接库文件，添加以下链接库：

OpenThreadsd.lib

osgd.lib

osgDBd.lib

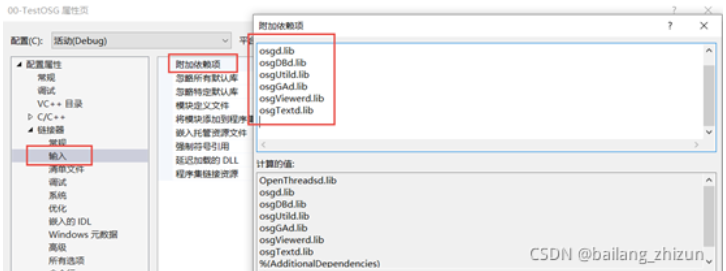
osgUtild.lib

osgGAd.lib

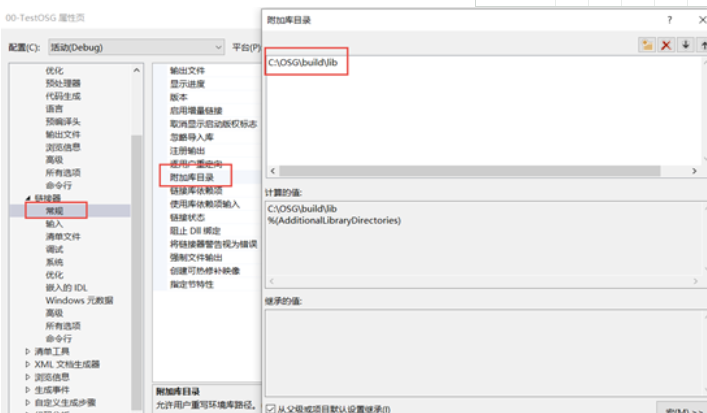
osgViewerd.lib

osgTextd.lib

（注：在编译生成的lib文件中，每个lib文件有一对名称相似的文件，其中文件名最后一个字母为“d”的为Debug版本所使用的，没有“d”的为Release版本所使用的。）



然后配置链接库的目录（build目录下的lib目录）：



在程序文件中编写以下代码：

#include <iostream>

#include <osgViewer/Viewer>

#include <osgDB/ReadFile>

int main()

{

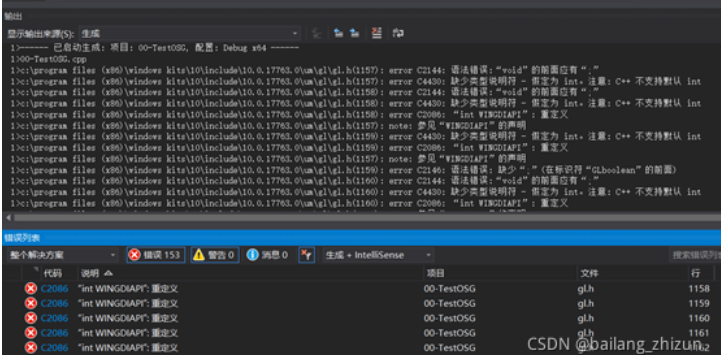
osgViewer::Viewer viewer;

viewer.setSceneData(osgDB::readNodeFile("cow.osg"));

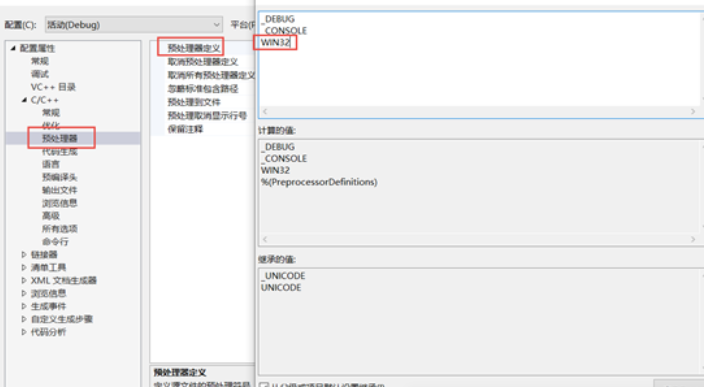
return viewer.run();

}

编译出现以下错误：



经调试发现，main函数中什么语句都不写，只是单纯的包含这两个头文件就会报这些错误，所以应该不是代码的问题，经查，应该是预处理器未定义的原因，需要在预处理器中添加 WIN32：



或者在程序中所有头文件之上添加 #include <Window.h>

编译没问题，运行结果：



（牛~，你终于出来了，喜极而泣）