**OSGEarth开发指导**

# 编写目的

OSGEarth是3D GIS平台开源项目，它的所有源代码对使用者完全开放，但开源项目毕竟不同于商业开发平台，它的学习文档和帮助手册是比较缺乏的。使用者需要对源代码进行编译，利用代码中的样例来学习OSGEarth是如何被开发出来，内部功能是怎样实现的，这样才能根据需要对已有代码进行修改或者增加新的功能。很多程序员不能够顺利地编译源代码，因而放弃了对该开源项目的研究，非常可惜。本文档为有志于从事OSGEarth开发的程序员提供搭建开发环境的指导，减少前期遇到的困难，帮助他们快速入门。此外，为他们进行后续的开发工作提供一些帮助。

# 读者对象

进行OSGEarth二次开发的程序员或设计人员。

# 开发语言

OSGEarth开发所使用的语言为ANSI C++，所以要求开发人员具有C++的开发经验和GIS专业技术知识，以便更容易入门并掌握开发技能。

# 所依赖框架

OSGEarth使用OSG作为三维引擎，它在OSG的基础上完成了3D地理信息的基本功能。OSGEarth独立于桌面开发框架的，所以开发者可自行选择C++程序的桌面开发框架，笔者使用的是QT。

**OSGEarth（地理信息模块）**

**QT桌面程序开发框架**

OSG：它是开源高性能3D图形开发工具，在视觉仿真、游戏、虚拟现实、数据建模和可视化领域被广泛使用。它完全使用标准C++封装了OpenGL的各项功能，能运行在各版本的Windows、OSX、GNU/Linux、IRIX、Solaris、HP-Ux、AIX和FreeBSD 系统上。更多信息可访问网站[www.openscenegraph.org](http://www.openscenegraph.org)。

**OSG（三维引擎）**

OSGEarth：它是基于OSG开发的地理信息模块，同样也是开源的。它可使用DOM和DEM数据显示3D地形，还可以加载2D矢量数据，实现了二三维数据一体化显示。更多信息可访问网站[www.osgearth.org](http://www.osgearth.org)。

QT：它是跨平台的桌面程序开发框架，也可运行在嵌入系统和手机中。支持的平台包括Linux、OS X、Windows、VxWorks、QNX、Android、iOS、BlackBerry、Sailfish OS等。它基于ANSI C++编写，并使用MOC预处理工具让C++语言具有signals和slots功能. 它也是开源项目，开发者可使用Clang, GCC, ICC, MinGW 和MSVC进行编译。更多信息可访问网站[www.qt.io](http://www.qt.io)。

# 前期准备

在进行后续步骤前，需要准备如下软件和源代码。

## Visual Studio（以后简称VS）

请先安装VS，它是C++程序的IDE环境，具有编译、链接、调试等功能。笔者使用的Visual Studio 2013。

## OSG依赖库

从OSG网站（[www.openscenegraph.org](http://www.openscenegraph.org)）下载和VS版本对应的依赖库。笔者下载的是3rdParty\_VS2013\_v120\_x86\_x64\_V9\_full.7z，并进行解压。

## OSG源代码

从OSG网站（[www.openscenegraph.org](http://www.openscenegraph.org)）下载最新版本的OSG源代码。把下载的源代码压缩包解压到某目录下，例如D:\OSG\3.4.0；笔者使用的是OSG 3.4.0。

## OSGEarth源代码

从OSGEarth（www.osgearth.org）网站下载最新版本的OSGEarth源代码。把下载的源代码压缩包解压到某目录下，例如D:\OSG\OSGEarth2.7；笔者使用的是OSGEarth2.7。

## QT 4.8.7

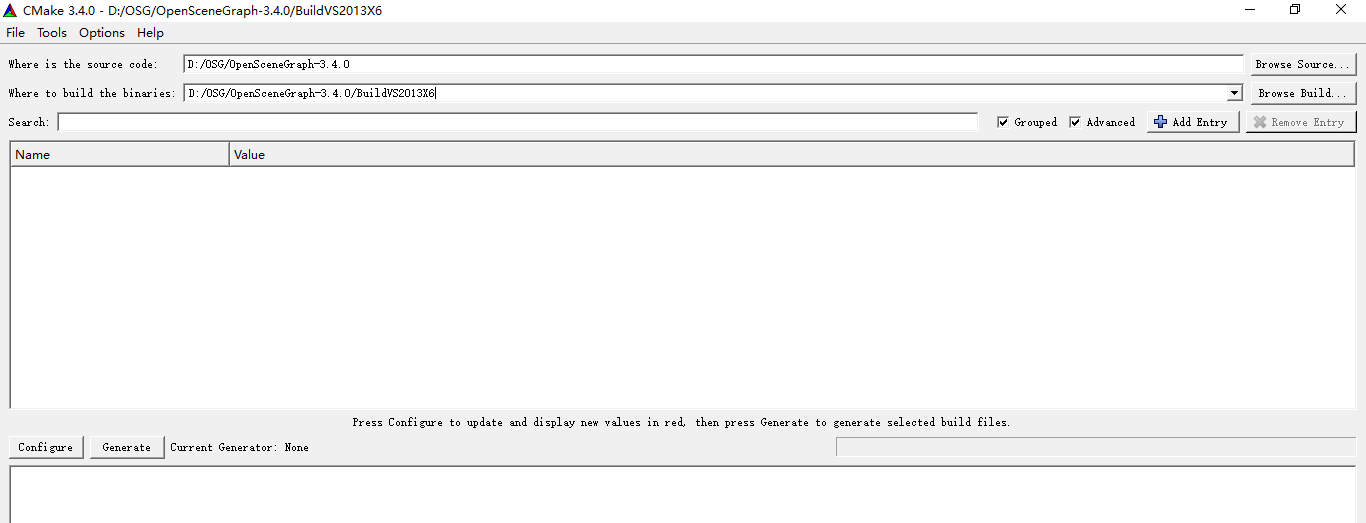
从QT网站（www.qt.io）下载QT4.8.X，笔者使用的是QT4.8.7（注意不要使用QT5，因为QT5和OSG集成时会出现线程方面的问题。笔者决定编译OSGEarth 64位程序，而网上却不能找到QT64位运行程序，所以下载了QT4.8.7源代码，并按照编译64位程序的方法编译了QT4.8.7）。

# 源代码编译

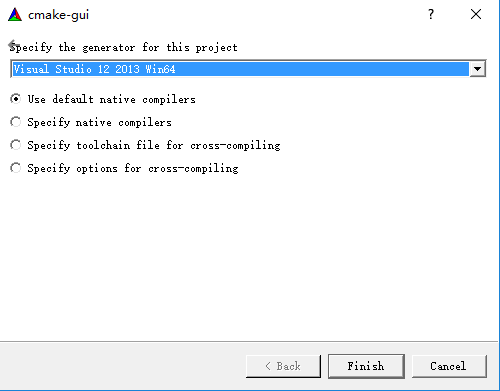
在进行开发工作之前，需要把OSG和OSGEarth编译成功。

## OSG编译

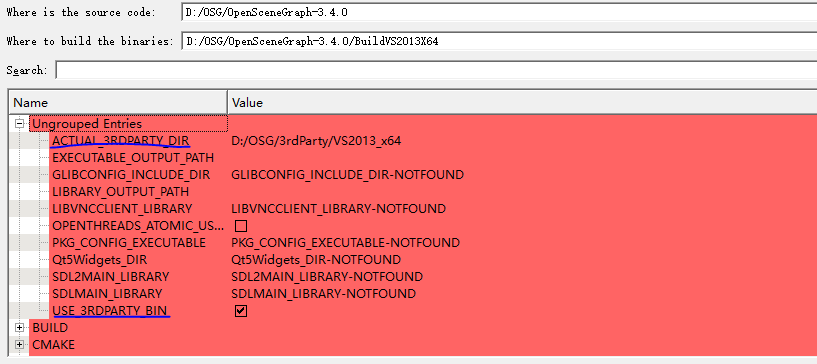
1. 从CMAKE网站（www.cmake.org）下载最新版的安装程序，笔者安装的是CMAKE 3.4.0；CMAKE是开源项目常用的预编译工具，它能根据开发者选择的操作系统、IDE版本、各项配置生成对应的工程文件，方便开发者进行源代码编译；
2. 安装并运行CMAKE程序，启动CMAKE界面，在Where is the source code文本框中选择OSG 3.4.0的解压路径，例如D:\OSG\3.4.0。在Where to build the binaries文本框中选择Visual Studio工程的生成路径，例如D:\OSG\3.4.0\Build，为了避免工程文件掺入到源代码目录中，VS工程生成路径和源代码路径不要相同；



1. 勾选“Grouped”和”Advanced”核选框后，点击Configure按钮，会弹出如下对话框，请选择Visual Studio版本，笔者用的是Visual Stuido 12 2013 Win64，即使用VS2013编译64位程序；



1. 点击“Finish”按钮后，弹出框关闭，会出现如下结果



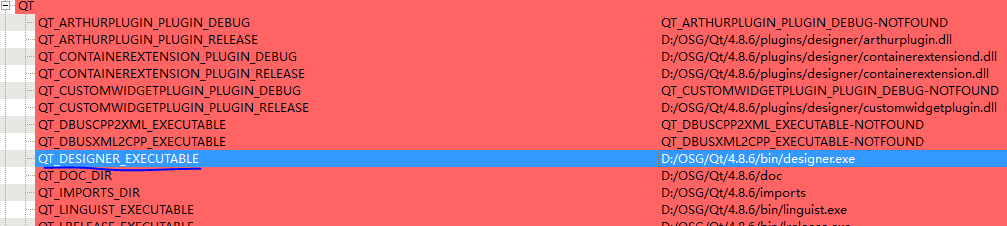
1. 在Ungrouped Entries组下有ACTUAL\_3RDPARTY\_DIR变量，需要设定OSG依赖库的路径，例如D:/OSG/3rdParty/VS2013\_x64，并勾选USE\_3RDPARTY\_BIN变量；
2. 在BUILD组下勾选BUILD\_OSG\_APPLICATIONS和BUILD\_OSG\_EXAMPLES，表示待生成的工程文件中包含OSG应用程序和样例程序；



1. 在OSG组下勾选OSG\_USE\_QT，其他的使用默认设置



1. 在QT组下设定QT\_DESIGNER\_EXECUTABLE的路径，例如D:/OSG/qt-everywhere-opensource-src-4.8.7/bin/designer.exe



1. 继续点击“Configure”按钮，观察CURL、FREETYPE、 GDAL、GIFLIB、JPEG、OPENGL、PNG、QT、TIFF、ZLIB目录下的变量是否都已设定路径（由于对ACTUAL\_3RDPARTY\_DIR变量进行了设置，CMAKE程序会搜索CURL、FREETYPE、GDAL、GIFLIB、JPEG、PNG、TIFF、ZLIB各库的路径并设置相应的变量）；
2. 如果都已被设定，就点击“Generate”按钮，生成VS solution文件。
3. 为了让OSG支持中文路径和中文文件名，需要做如下修改：

在osgDB工程里，打开fstream.cpp文件，把原来的

#ifdef OSG\_USE\_UTF8\_FILENAME

#define OSGDB\_CONVERT\_UTF8\_FILENAME(s) convertUTF8toUTF16(s)

#else

......

#endif

改为

#ifdef OSG\_USE\_UTF8\_FILENAME

#define OSGDB\_CONVERT\_UTF8\_FILENAME(s) convertUTF8toUTF16(s).c\_str()

#else

#ifdef \_MSC\_VER

#define OSGDB\_CONVERT\_UTF8\_FILENAME(s) convertUTF8toUTF16(convertStringFromCurrentCodePageToUTF8(s)).c\_str()

#else

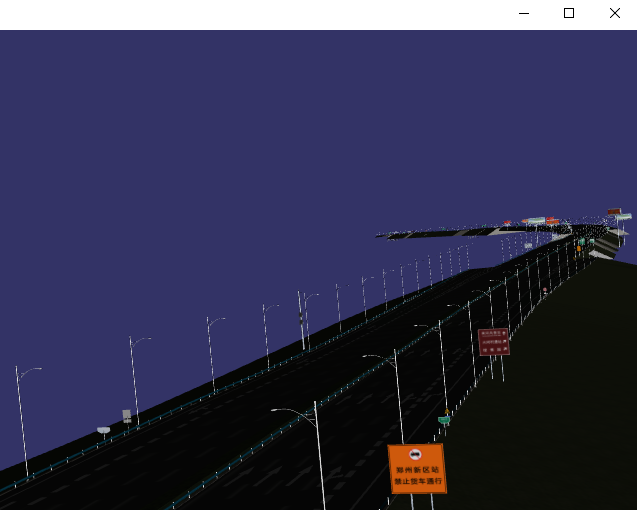
#define OSGDB\_CONVERT\_UTF8\_FILENAME(s) convertUTF8toUTF16(s).c\_str()

#endif

#endif

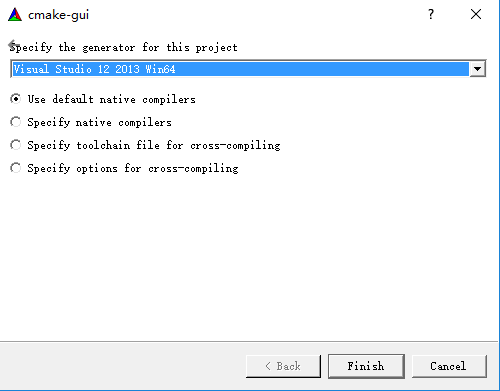
这是因为在windows操作系统里，字符被分配了codepage编码，所以还需要转换字符的codepage编码；

1. 在工程路径下双击OpenSceneGraph.sln文件启动Visual Stuido，在debug和release模式下分别编译OSG Core下的所有工程，然后编译Plugins目录下的所有工程；（debug模型用于调试程序，release模式用于正式发布程序）
2. 编译Applciations目录下的osgViewer工程和osgversion工程；
3. 在cmd窗口下设定path变量，让它包括OSG Core下（例如 osg.dll、osgDB.dll等）的DLL文件路径和OSG依赖库DLL文件路径（例如geos.dll、libcurl.dll等），再运行命令osgviewer 模型文件，模型文件可为3ds、osg、osgb、ive、dae、obj格式。如果能出现如下窗口，证明OSG能正常运行；

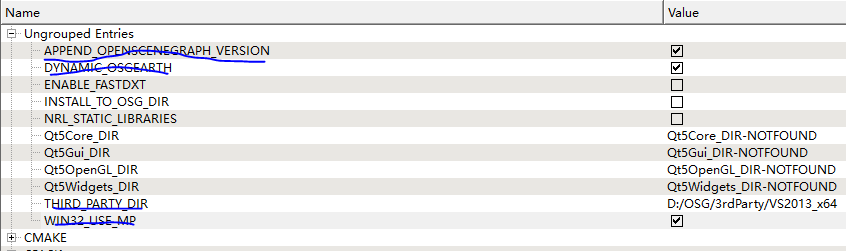


## OSGEarth编译

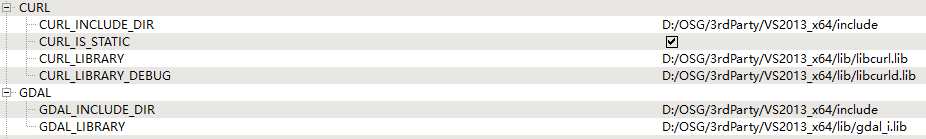
1. 运行CMAKE程序，启动CMAKE界面，在Where is the source code文本框中选择OSGEarth2.7的解压路径，例如D:\OSG\OSGEarth2.7。在Where to build the binaries文本框中选择Visual Studio工程的生成路径，例如D:\OSG\OSGEarth2.7\Build，为了避免工程文件掺入到源代码目录中，VS工程生成路径和源代码路径不要相同；
2. 勾选“Grouped”和”Advanced”核选框后，点击Configure按钮，会弹出如下对话框，请选择Visual Studio版本，笔者用的是Visual Stuido 12 2013 Win64，即使用VS2013编译64位程序；



1. 点击“Finish”按钮后，弹出框关闭，会出现如下结果



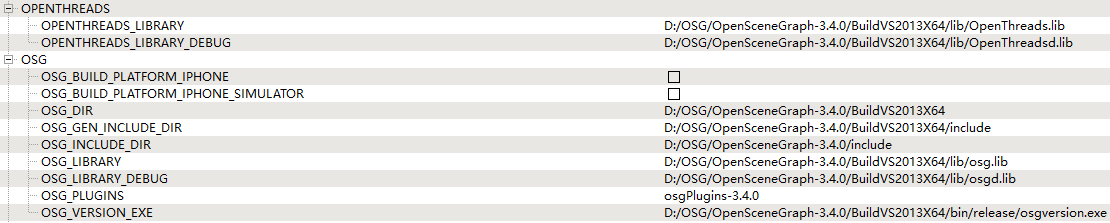
1. 在Ungrouped Entries组下勾选变量APPEND\_OPENSCENEGRAPH\_VERSION、DYNAMIC\_OSGEARTH、WIN32\_USE\_MP，并把变量THIRD\_PARTY\_DIR设置到OSG的依赖库目录，例如D:/OSG/3rdParty/VS2013\_x64。
2. 点击”Configure”按钮，观察CURL、GDAL、ZLIB组下的变量是否已有对应的值，如下图所示：



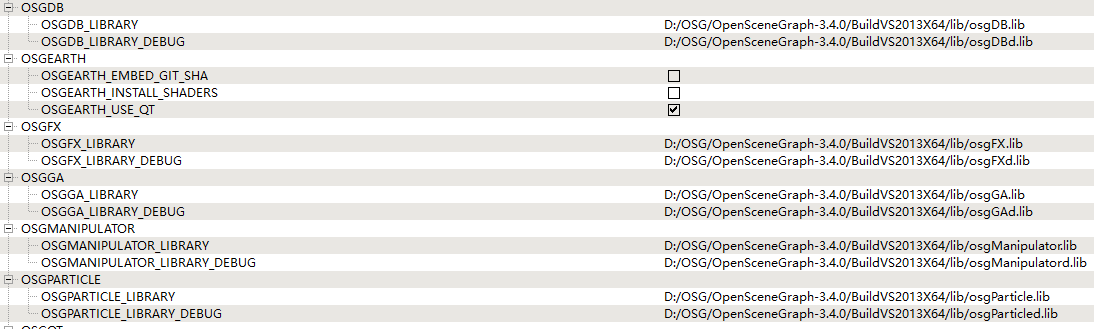
1. 展开”OPENSCENEGRAPH”组，输入OSG对应的版本号，例如MAJOR\_VERSION为3，MINOR\_VERSION为4，PATCH\_VERISON为0。



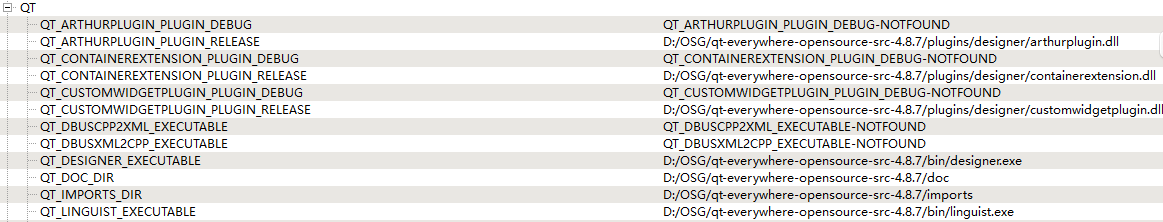
1. 分别展开OPENTHREADS组和OSG组，设置各自的LIBRARY和INCLUDE变量值，其中OSG\_DIR指向OSG的工程路径，OSG\_GEN\_INCLUDE\_DIR指向工程路径下的include目录，它为OSG编译时生成的目录。OSG\_VERSION\_EXE是OSG Applications目录下的应用程序。



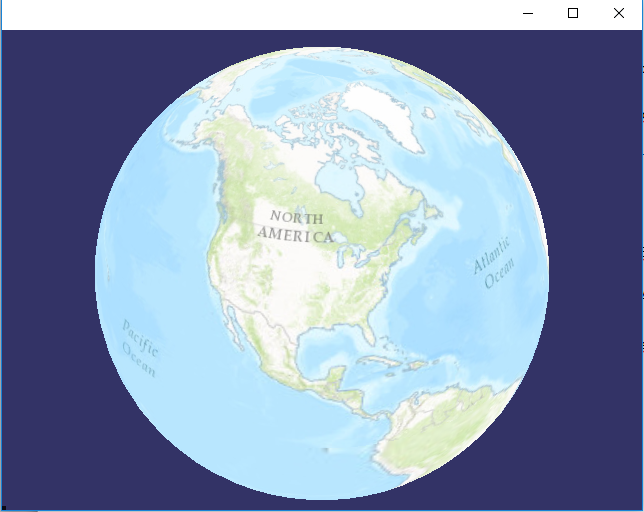
1. 相应地设置OSGDB、OSGFX、OSGGA、OSGMANIPULATOR、OSGPARTICLE、OSGQT、OSGSHADOW、OSGSIM、OSGTERRAIN、OSGTEXT、OSGUTIL、OSGVIEWER、OSGWIDGET组中变量的值。（XXX\_LIBRARY指向静态库文件xxx.lib，XXX\_LIBRARY\_DEBUG指向调试版本静态库文件xxxd.lib）



1. 展开OSGEARTH组，勾选OSGEARTH\_USE\_QT；
2. 在QT组下设定QT\_DESIGNER\_EXECUTABLE的路径，例如D:/OSG/qt-everywhere-opensource-src-4.8.7/bin/designer.exe；



1. 继续点击“Configure”按钮，观察OSGDB、OSGFX、OSGGA、OSGMANIPULATOR、OSGPARTICLE、OSGQT、OSGSHADOW、OSGSIM、OSGTERRAIN、OSGTEXT、OSGUTIL、OSGVIEWER、OSGWIDGET目录下的变量是否都已设定路径；
2. 如果都已被设定，就点击“Generate”按钮，生成VS solution文件；
3. 在工程路径下双击OSGEARTH.sln文件打开Visual Stuido，在debug和release模式下编译Libs下的所有工程，然后编译Plugins目录下的所有工程；
4. 编译Tools目录下的osgearth\_viewer工程；
5. 在cmd窗口下设定path变量，让它包括OSG和OSGEarth（例如 osg.dll、osgDB.dll、osgEarth.dll、osgdb\_osgearth\_wms.dll等）的DLL文件路径和OSG依赖库DLL文件路径（例如geos.dll、libcurl.dll等），再运行命令osgearth\_viewer earth文件，例如 osgearth\_viewer arcgisonline.earth。如果能出现如下窗口，证明OSGEarth已能成功运行；



osgearth压缩包解压后，在tests目录下有很多样例earth文件。

## OSGEarth和QT的集成

在OSGEarth工程下有osgearth\_qt、osgearth\_qt\_simple、osgearth\_qt\_windows三个osgearth和QT集成的例子。开发者可通过学习这几个例子来掌握如何在QT环境中集成OSGEarth代码。

## CMAKE文件编写

如果想让自己编写的代码也具有跨平台的特性，那么需要给自己的代码编写CMakeLists.txt文件，如下所示：

CMAKE\_MINIMUM\_REQUIRED(VERSION 2.6.4 FATAL\_ERROR)

if(COMMAND cmake\_policy)

# Works around warnings libraries linked against that don't

# have absolute paths (e.g. -lpthreads)

cmake\_policy(SET CMP0003 NEW)

# Works around warnings about escaped quotes in ADD\_DEFINITIONS

# statements.

cmake\_policy(SET CMP0005 OLD)

# disable autolinking to qtmain as we have our own main() functions (new in Qt 5.1)

if(NOT "${CMAKE\_VERSION}" VERSION\_LESS 2.8.11)

cmake\_policy(SET CMP0020 OLD)

endif(NOT "${CMAKE\_VERSION}" VERSION\_LESS 2.8.11)

endif(COMMAND cmake\_policy)

#

# Set up CMake to use Solution Folders in VS.

#

SET\_PROPERTY( GLOBAL PROPERTY USE\_FOLDERS ON )

SET\_PROPERTY( GLOBAL PROPERTY PREDEFINED\_TARGETS\_FOLDER "CMake Targets" )

PROJECT(OSGEARTH)

SET(OSGEARTH\_MAJOR\_VERSION 2)

SET(OSGEARTH\_MINOR\_VERSION 6)

SET(OSGEARTH\_PATCH\_VERSION 0)

SET(OSGEARTH\_SOVERSION 0)

SET(OSGEARTH\_PLUGIN\_PREFIX "")

CMAKE语言可通过开源代码所带的例子来学习，也可通过www.cmake.org网站来学习CMAKE的语法。