

Installing OpenSceneGraph

Get OpenSceneGraph from <http://www.openscenegraph.org> .

There are two options for installing OpenSceneGraph:

- * download and install prebuild packages for your system (e.g. [from AlfaPixel](#))
- * get the OpenSceneGraph source package and build it from source

If you encounter problems with the prebuild packages you can try to build from source. Follow the instruction on the OSG website. Do not forget to get the required 3rd Party dependencies for OpenSceneGraph (you need the appropriate libraries to load scenes that use specific image file format like .png .jpg). The latest version of CurvaceSimulator uses version 3.0.1 of OpenSceneGraph.

Environment Variables for OSG

The Following is an excerpt from the osg website. Setting those paths is recommended for developing OSG applications and ensures that cmake can find your osg installation

Excerpt from the osg website on setting of visual studio projects

Environment variables

If you select the first method, a way to simplify things and to make your project files work on multiple different machines is to use some standard environment variables. I use the following:

OSG_ROOT points to the base of the OSG file structure (the directory that contains include, src etc. subdirectories)

```
OSG_BIN_PATH = %OSG_ROOT%bin
```

```
OSG_INCLUDE_PATH = %OSG_ROOT%include
```

```
OSG_LIB_PATH = %OSG_ROOT%lib
```

```
OSG_SAMPLES_PATH = %OSG_ROOT%shareOpenSceneGraphbin
```

```
OSG_FILE_PATH = ???OpenSceneGraph-Data-X.X
```

Then, add %OSG_BIN_PATH% and %OSG_SAMPLES_PATH% to your PATH environment variable. That way, not only can you run examples easily, but the latest DLLs will always be found. When starting an application, Windows looks for the required DLLs first in the executable's directory, then in the PATH.

Make sure you restart Visual Studio if it was open when you added/changed these environment variables, so that it picks up the changes.

In your project's properties, use those environment variables to get Visual Studio to find the OSG libraries. Here are the settings I use:

(again you can ignore this when using the simulator solution file)

Properties - C/C++ - General - Additional Include Directories = \$(OSG_INCLUDE_PATH)

Properties - C/C++ - Preprocessor - Preprocessor Definitions = WIN32;_WIN32;NDEBUG

Properties - Linker - General - Additional Library Directories = \$(OSG_LIB_PATH)

Properties - Linker - Input - Additional Dependencies = (any OSG library your project needs - for example: osg.lib osgGA.lib osgDB.lib osgViewer.lib osgText.lib osgUtil.lib OpenThreads.lib)

Remember to do the same thing in your project's Debug configuration, but add a d to the end of the OSG and OpenThreads library names (osg.lib becomes osgd.lib and so on). Also change NDEBUG to _DEBUG in the Preprocessor Definitions.

Once that's done, your project should be able to compile, link and run. If not, you can test your paths in a console (Start-Run, type cmd, press enter), by for example typing echo %OSG_LIB_PATH% or dir %OSG_LIB_PATH%. With those commands you should be able to see what is wrong. **A2 - Getting CMake**

OpenSceneGraph as well as CurvaceSimulator require CMake to build.

Get it from:

<http://www.cmake.org/>

A3 - Installing QT4

For windows you can get the qt4 libraries from <http://qt.nokia.com/downloads> and follow the installation instructions. Be sure to get the visual studio libraries (Qt libraries... for windows (VS ...)) if you use VS, since those are not included in the SDK. Create an environment variable QT_PATH pointing to the installation directory of QT and add %QT_PATH%bin to your PATH variable.

(**Note** that some other programs might come with their own version of Qt and add their path to your Path environment variable. This can cause conflicts if the wrong Qt version is loaded.)

Be sure to get the 'Qt libraries ... for windows (VS ...)'

Follow the appropriate installation instructions.

Don't forget to set the appropriate path variable QT_PATH

For linux you can usually install qt4 through your packet manager.