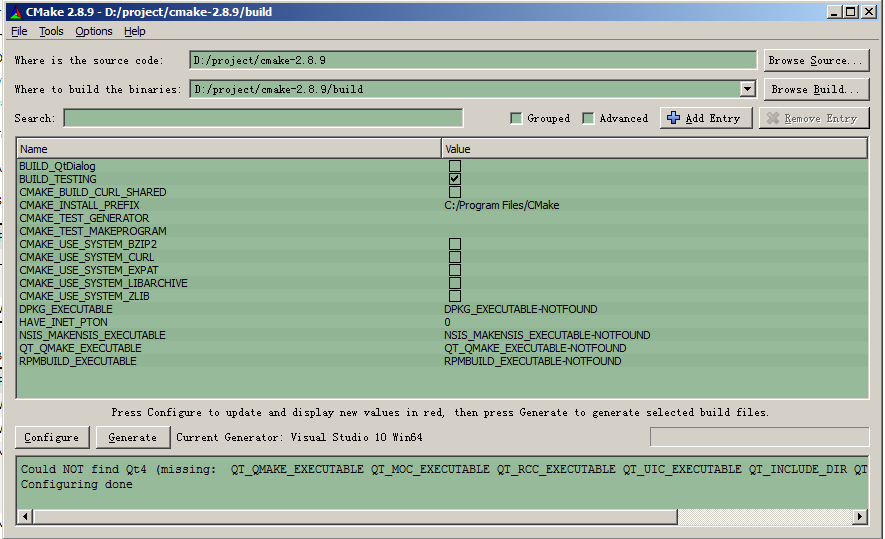
Build Alembic For Maya Windows with Vs2008 sp1

1. **Install CMake**
   1. Download and Install Cmake win32  installer
      1. <http://www.cmake.org/files/v2.8/cmake-2.8.9-win32-x86.exe>
   2. Download Cmake source code and copy to d:/project/cmake-2.8.9
      1. <http://www.cmake.org/files/v2.8/cmake-2.8.9.zip>
   3. Run [C:\Program Files (x86)\CMake 2.8\bin\cmake-gui.exe](file:///C:\Program%20Files%20(x86)\CMake%202.8\bin\cmake-gui.exe).
      1. “source code”: d:/project/cmake-2.8.9
      2. “build the binaries”: d:/project/cmake-2.8.9/build
      3. Push “Configure” button and select “Vusal Studio”, then Generate.



* 1. Open D:\project\cmake-2.8.9\build\ CMake.sln with VS2008 sp1 and build it. The build folder is D:\project\cmake-2.8.9\build\bin\Release. Please add this to your system PATH variable.

1. **Install Python**
2. **Install GLew libraries**
   1. <https://sourceforge.net/projects/glew/files/glew/1.9.0/glew-1.9.0-win64.zip/download>
   2. copy the headers and libraries into their destination directories:

|  |  |  |
| --- | --- | --- |
| bin/glew32.dll | to | %SystemRoot%/system32 |
| lib/glew32.lib | to | {VC Root}/Lib |
| include/GL/glew.h | to | {VC Root}/Include/GL |
| include/GL/wglew.h | to | {VC Root}/Include/GL |

1. **Install GLUT libraries**
   1. [**http://user.xmission.com/~nate/glut/glut-3.7.6-bin.zip**](http://user.xmission.com/~nate/glut/glut-3.7.6-bin.zip)
   2. Like Install Glew libraries

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

1. **Download the following and place the source in the corresponding folders in %ALEMBIC\_ROOT%\contrib:**
   1. boost\_1\_42\_0\ <http://www.boost.org/users/history/version_1_42_0>
   2. hdf5-1.8.5-patch1\ <http://www.hdfgroup.org/ftp/HDF5/prev-releases/hdf5-1.8.5-patch1/src/>
   3. ilmbase-1.0.2\ <http://www.openexr.com/downloads.html>
   4. openexr-1.6.1\ <http://www.openexr.com/downloads.html>
2. **Install ZLIB 1.2.5**
   1. Download source code and copy to d:\project\zlib-1.2.5
      1. <http://sourceforge.net/projects/libpng/files/zlib/1.2.5/zlib-1.2.5.tar.gz/download?use_mirror=hivelocity&download=>
   2. Open [d:\project\zlib-1.2.5\contrib\vstudio\vc9\zlibvc.sln](file:///d:\project\zlib-1.2.5\contrib\vstudio\vc9\zlibvc.sln) with Microsoft Visual C++ 2008.0 and build
   3. Copy “[D:\project\zlib-1.2.5\contrib\vstudio\vc9\x64\ZlibStatRelease\ zlibstat.lib](zlibstat.lib)” to [D:\project\alembic\source\contrib\zlib-1.2.5](file:///D:\project\alembic\source\contrib\zlib-1.2.5)
   4. Copy “zconf.h” and “zlib.h” to [D:\project\alembic\source\contrib\zlib-1.2.5\include](file:///D:\project\alembic\source\contrib\zlib-1.2.5\include)
3. **Download Alembic source and modify some files:**
   1. Download source and copy to D:\project\alembic\source
      1. <http://alembic.googlecode.com/files/Alembic_1.1.1_20120921.tgz>
   2. Download <http://pete-win.googlecode.com/hg-history/a9a512e44509b1514609c4d997c9de0f6df62613/IlmBase-vc9.zip> into D:\project\alembic\source \contrib\ilmbase-1.0.2\vc\vc9
   3. Unzip build.zip(<http://abc-hongloull.googlecode.com/files/build.zip>) to D:\project\alembic\source\build and repalce. Files edited list:
      1. init\_Alembic.cmd
         1. set BOOST\_ARGS=--boost\_include\_dir=%rootOut%\boost\include\boost-1\_42 --boost\_thread\_library=%platOut%\boost%db%\lib\libboost\_thread-vc90-mt-1\_42.lib
         2. set ZLIB\_ARGS=--zlib\_include\_dir=D:\project\alembic\source\contrib\zlib-1.2.5\include --zlib\_library=D:\project\alembic\source\contrib\zlib-1.2.5\zlibstat.lib
         3. set BASE\_ARGS=--with-maya=%MAYA\_ROOT% --generator=%Generator%
      2. Edit the “srcDir” line of build\_ilmbase.cmd to point to the folder containg the VC solution file:“set srcDir=%srcRoot%\vc\vc9\IlmBase”
      3. “build\_boost.cmd”:
         1. set BUILD\_THESE\_ONLY= --with-program\_options --with-iostreams --with-thread --with-date\_time --with-python
      4. D:\project\alembic\source\build\bootstrap\alembic\_bootstrap.py
         1. cmake\_extra\_args += " -D BOOST\_LIBRARYDIR:PATH=\"D:\\project\\alembic\\output\\i64\\boost\\lib\""
         2. print "Executing CMake Zlib trycompile command:\n%s" % cmake\_cmd

os.system(cmake\_cmd)

return 0,'e'

1. **Set System Variable**
   1. ALEMBIC\_ROOT defines where the alembic source is stored
      1. ALEMBIC\_ROOT= D:\project\alembic\source
   2. ALEMBIC\_OUT defines where the alembic output is built
      1. ALEMBIC\_OUT= D:\project\alembic\output
   3. PROCESSOR\_ARCHITECTURE = AMD64
   4. Add “D:\project\cmake2.8.9\build\bin\Release;D:\project\alembic\output\i64\boost\lib;C:\Program Files (x86)\Microsoft Visual Studio 9.0\VC\bin\amd64;” to “PATH”
2. **Add “unist.h” to vc include path:**
   1. Get “unist.h” from build.zip and copy to your vc include folder, for example: C:\Program Files (x86)\Microsoft Visual Studio 9.0\VC\include
3. **Build Alembic:**
   1. Run “Visual Studio 2008 x64 win64 command Prompt” tool
   2. Enter “cd /d D:\project\alembic\source\build\Windows” and Return
   3. Enter “init\_all.cmd” and Return
   4. Add “/FORCE:MULTIPLE” to build\_Albemic.cmd:
      1. vcbuild /FORCE:MULTIPLE /nologo %1 %2 %3 %4 %5 %6 %outDir%\Alembic.sln %buildArg%
      2. Enter “build\_Albemic.cmd” and Return
4. **Build Alembic for maya:**
   1. Unzip maya.zip(<http://abc-hongloull.googlecode.com/files/maya.zip>) into “D:\project\alembic\source\maya” and replace. Edited files list:
      1. AbcExport\Foundation.h

Add “**#include** <winsock2.h>

**#include** <windows.h>”

* + 1. AbcImport\main.cpp

Add “**#include** <winsock2.h>

**#include** <windows.h>”

* + 1. AbcImport\ util.h

Add “**#include** <winsock2.h>

**#include** <windows.h>”

* + 1. AbcImport\ NodeIteratorVisitorHelper.cpp
       1. replace “std::min” with “(std::min)”
       2. replace “std::max” with “(std::max)”
    2. AbcImport\ NurbsSurfaceHelper.cpp
       1. replace “std::min” with “(std::min)”
       2. replace “std::max” with “(std::max)”
  1. Open “D:\project\alembic\output\i64\maya\AbcExport\ ABC\_EXPORT\_MAYA\_MODULE.sln” with vs2008 sp1
     1. Edit “AbcImport”: “Properties”🡪”Configuration Properties” 🡪”Linker”🡪”Command Line” add “/FORCE:MULTIPLE”
  2. Open “D:\project\alembic\output\i64\maya\AbcImport\ ABC\_IMPORT\_MAYA\_MODULE.sln” with vs2008 sp1
     1. Edit “AbcExport”: “Properties”🡪”Configuration Properties” 🡪”Linker”🡪”Command Line” add “/FORCE:MULTIPLE”

1. **Alembic AbcImport and AbcExport (v1.1.1) download:**

<http://abc-hongloull.googlecode.com/files/AbcExport_AbcImport_Maya2012x64.zip>

**Known Issues (& solutions):**

Thanks to Robert Durnin and Jesse Lehrman for calling these in.

**1) \_eLut / \_toFloat linker problems**

**Q:** When you use the OpenEXR projects (or header files) you get one of these linker errors?

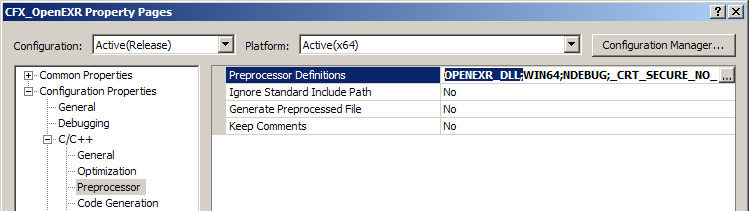
foobar.obj : error LNK2019: unresolved external symbol "private: static unsigned short const \* const half::\_eLut" (?\_eLut@half@@0QBGB)

referenced in function "public: \_\_cdecl half::half(float)" (??0half@@QEAA@M@Z)

foobar.obj : error LNK2019: unresolved external symbol "private: static union half::uif const \* const half::\_toFloat" (?\_toFloat@half@@0QBTuif@1@B)

referenced in function "public: \_\_cdecl half::operator float(void)const " (??Bhalf@@QEBAMXZ)

**A:** You are using the OpenEXR dynamically loadable libraries (DLL's) — which is the default — and did *not* #define OPENEXR\_DLL in *either* your own project settings *or* in your source code before the OpenEXR headers are #include<>d.



## [Error hnk2005:exist already defined in msvcrt.lib(MSVCR100.dll)\_LIBCMTD.lib](http://www.cnblogs.com/chenkai/archive/2013/01/23/2873932.html)

完整解决方式采用强制链接器做法按照正确顺序链接.通过在Properties-Configuration Properties->Linker->Command Line设置参数： /FORCE:MULTIPLE

xtree std::map<\_Kty,\_Ty,\_Pr> const-volatile qualifiers while compiling class template member function 'std::\_Tree\_nod<\_Traits>::\_Node \*std::\_Tree<\_Traits> AbcWriteJob.h

Modify MayaUtility.h:

bool operator()( const MDagPath& lhs, const MDagPath& rhs ) const

bool operator()( const MDagPath& lhs, const MDagPath& rhs )

|  |  |  |
| --- | --- | --- |
|  |  |  |