

Lomse library. Installing on Microsoft Windows

This document contains detailed instructions for building Lomse library and tests program from sources, on Microsoft windows systems. If you find things that need to be fixed in this document and would like to contribute, you are welcome. Please post a message on the Lomse list. Thank you.

For installing the Lomse library you have to use CMake, version 2.8.0 or greater. CMake will generate for you a makefile for doing all necessary steps, as well as additional ones, such as building a test program for checking the library, and an example program.

If you don't have CMake, download and install the latest distribution from <http://www.cmake.org/cmake/resources/software.html>. Download the installer .exe file and run it.

Requirements

To build the lomse library, the following software should be installed in your system:

- CMake version 2.8 or higher (<http://www.cmake.org>)
- A command line SVN client
- UnitTest++ 1.3.0 or higher (<http://unittest-cpp.sourceforge.net/>)
- FreeType 2.3.5-1 or higher (<http://www.freetype.org/>)
- Boost Threads 1.42 or higher (<http://www.boost.org/>)
- zlib (<http://zlib.net/> and <http://gnuwin32.sourceforge.net/packages/zlib.htm>)
- libpng (<http://www.libpng.org/pub/png/libpng.html>)

For building, the script require access to a command line SVN client. One of the most common SVN clients used in Windows is TortoiseSVN (<http://tortoisesvn.net/>). But the command line binaries are not installed by default. If you install TortoiseSVN, remember to check the option "command line client tools" during TortoiseSVN installation.

Installation steps

Once you have checked that you have the required packages you can proceed to create the makefile or project file for building Lomse.

1. Get the Lomse source code

Create a folder for Lomse sources. For instance, "C:\MyData\Projects\lomse". Using your SVN client software download a working copy of Lomse project. The URL for the SVN repository is "https://lomse.svn.sourceforge.net/svnroot/lomse/trunk".

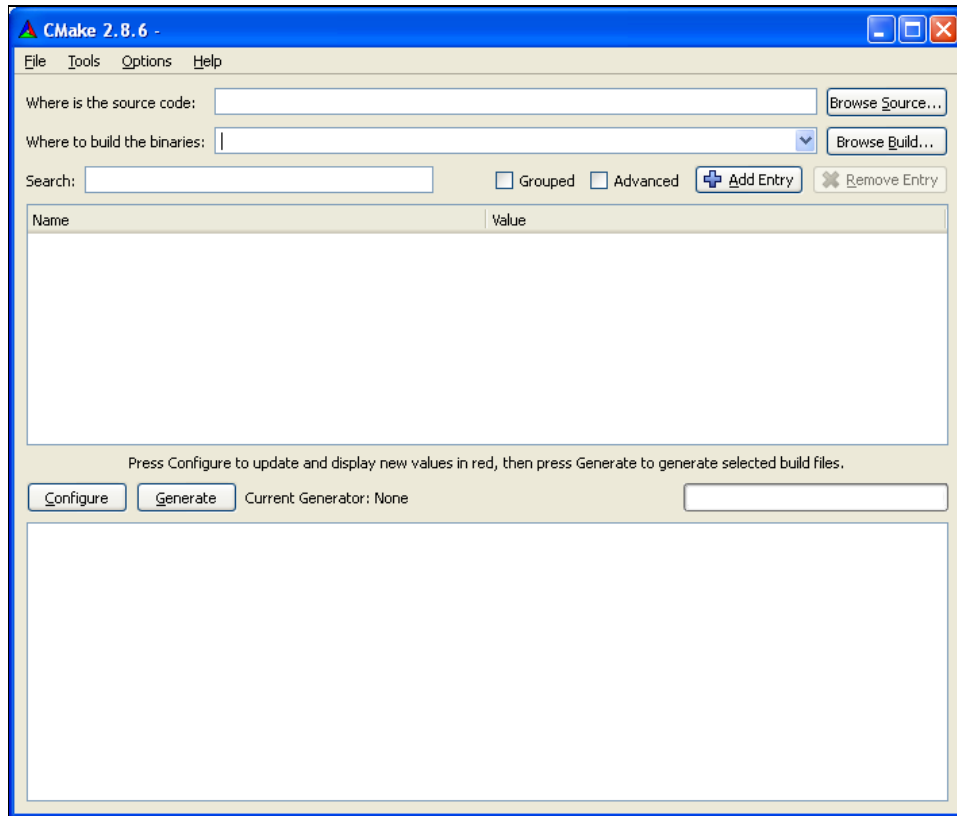
If you are using a command window, the command to issue would be:

```
svn co https://lomse.svn.sourceforge.net/svnroot/lomse/trunk .
```

2. Generate Makefile

Now, using CMake, proceed to create the makefile:

1. Open CMake GUI program. (usually at C:\Program Files\CMake\bin\cmake-gui.exe). It will display this window

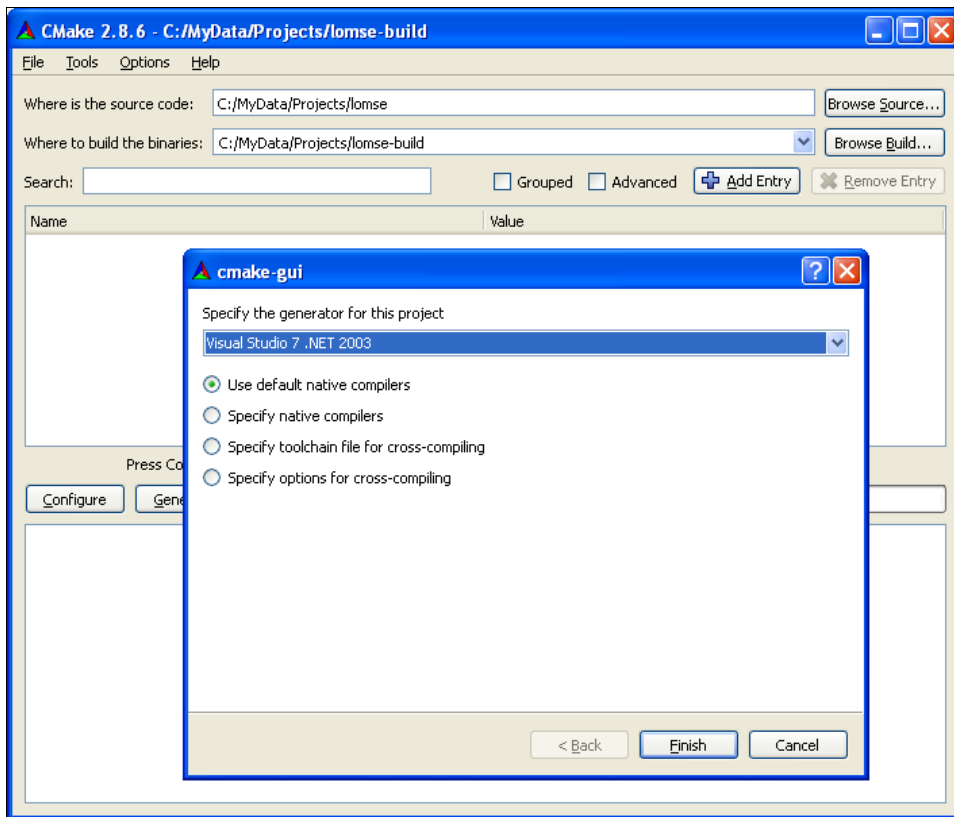


2. Click on the 'Browse Source...' button in the top right hand corner of the window. Use the file browser popup to locate the Lomse source directory (i.e.: C:\MyData\Projects\lomse), and click OK.
3. Now we create a directory in which to create the Visual Studio project files and hold the build products. This directory should not be the same as, or inside, the the source directory. Here we will create this build directory alongside our source directory.

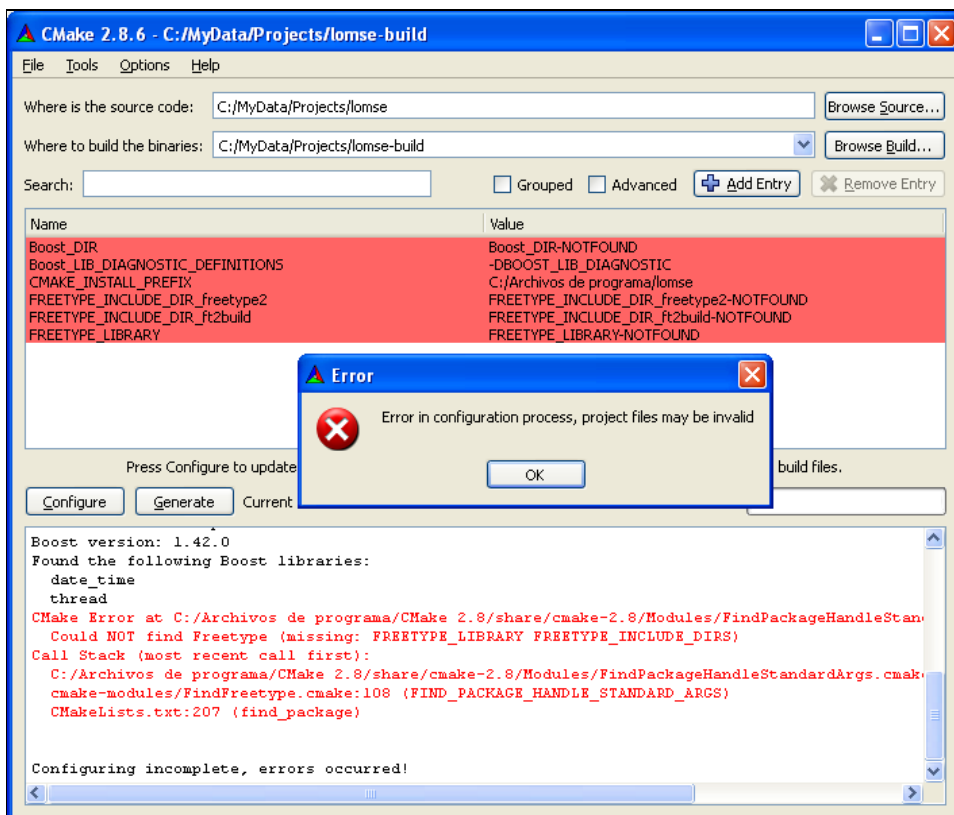
Click on the 'Browse Build...' button in the top right hand side of the CMake GUI window. Use the file browser popup to browse back to C:\MyData\Projects\, and click on the Make New Folder button. Rename the created folder to lomse-build, and click on the OK button.

The two text entries at the top of the GUI should now contain C:/MyData/Projects/lomse and C:/MyData/Projects/lomse-build respectively (Note: CMake always represents Windows paths with forward slashes).

Lomse installation

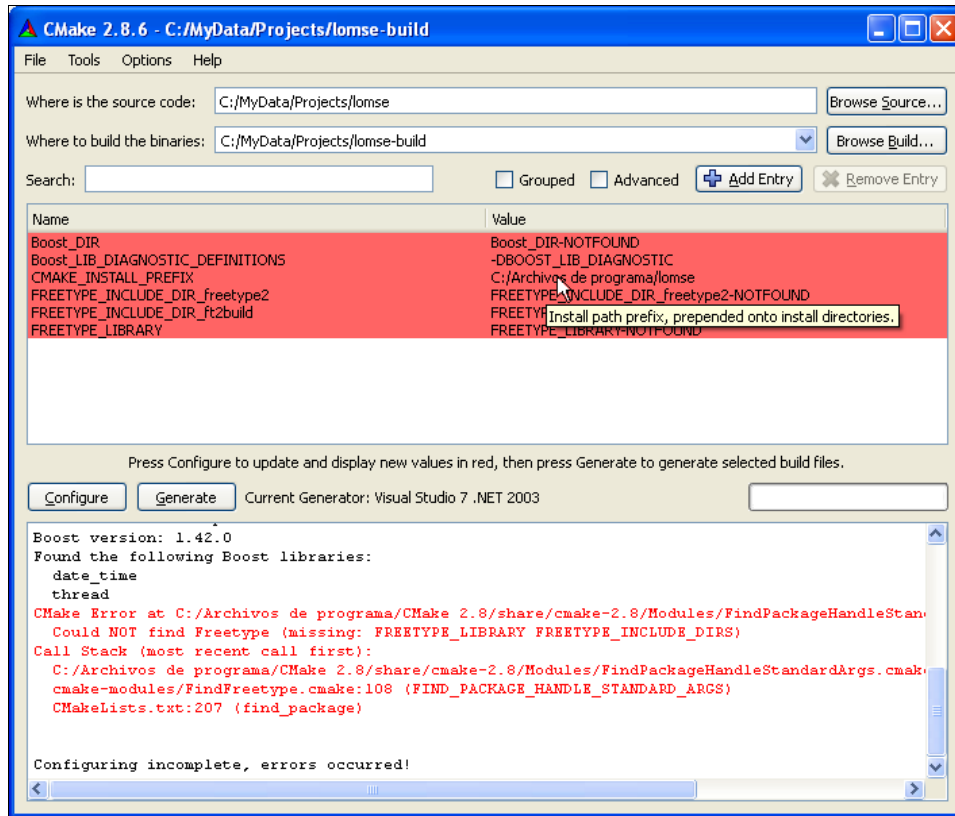


4. Click on the 'Configure' button on the left hand side. A window will open and you will be asked for which tool you would like to generate the makefile for. Select, for instance, "Visual Studio 7 .NET". Ensure only the 'Use default native compilers' radio box is checked. Then click on the 'Finish' button.



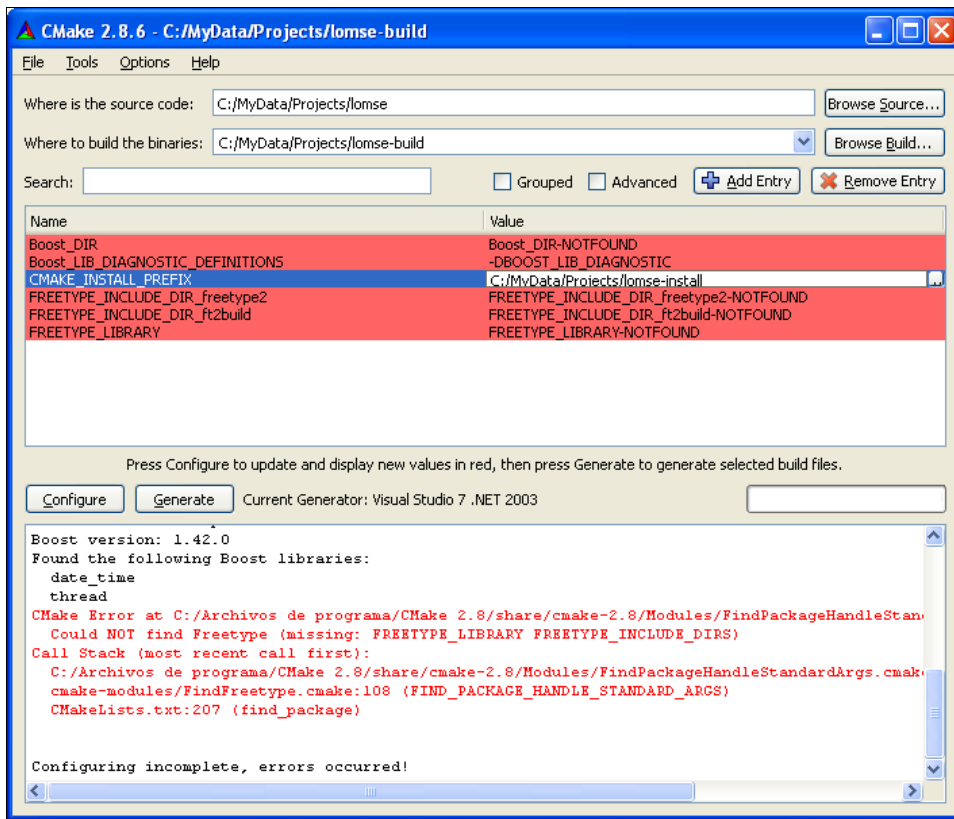
Lomse installation

Once the generator is selected, the configuration process will start. At some point, you will get a listing of all relevant options (CMakeCache entries). All entries at this point will be highlighted in red to signal that you may want to modify some of them. Mouse-hover over entries will generate a brief description, as shown below.



Most of the default settings should be fine, though you may want to change CMAKE_INSTALL_PREFIX. By default, CMake will configure the build to install Lomse under C:\Program Files\lomse. If you do not wish to use this directory, or do not have permission to install there, you can change the installation location. Click on the arrow next to the CMAKE entry in the central options display to expand all CMake specific configuration options. Scroll down to locate the CMAKE_INSTALL_PREFIX entry. Click on the entry and edit the path directly or via the file browser popup. In this example we will modify this install directory to C:/MyData/Projects/lomse-install.

Lomse installation



5. Once you have adjusted any options, click on the Configure button again. After this has finished, all the options listed in the central option display should be white. If you still have entries in red, click Configure to reconfigure until all entries are white.

Building with CMake is an iterative process:

- ◆ Select values, press the Configure button.
- ◆ Set the settings, run configure, set the settings, run configure, etc.
- ◆ Repeat until all values are set (no red lines) and the OK button becomes available.
- ◆ To set the value of a variable, click on that value:
 - ◇ If it is boolean (ON/OFF), a drop-down menu will appear for changing the value.
 - ◇ If it is file or directory, an ellipsis button will appear ("...") on the far right of the entry. Clicking this button will bring up the file or directory selection dialog.
 - ◇ If it is a string, it will become an editable string.
- ◆ Some variables (advanced variables) are not visible right away. To see them, toggle to advanced mode ("Show Advanced Values" toggle).

It may happen that CMake won't find some needed libraries, such as FreeType. In this case you will have three or four variables highlighted in red that you have to fix. You have two options for this:

1. Modify the CMAKE_PREFIX_PATH variable to hint search locations to CMake. For instance, if library FreeType is installed at 'C:\MySoftware\freetype' add this path to CMAKE_PREFIX_PATH.
 2. The other alternative is to directly set up the required variables (i.e. FREETYPE_INCLUDE_DIR_freetype, FREETYPE_INCLUDE_DIR_ft2build and FREETYPE_LIBRARY) to point to appropriate paths.
6. When no more red lines, click on the 'Generate' button. This will create the Visual Studio project files and will be placed in the specified folder. Finally, CMake will report Configuring done, Generating done in the logging window at the bottom of the GUI. You can now close the CMake GUI.

3. Build the library

At this point you've generated project files using CMake. Now, just run MS Visual Studio, open the solution file, choose your configuration, and build the ALL_BUILD project.

4. Install Lomse

Once you have built Lomse you can install it by building the INSTALL target in Visual Studio. In the Solution Explorer, right click on the INSTALL solution, and select Build from the popup menu. The solution will now build and install to the directory you chose for CMAKE_INSTALL_PREFIX earlier. If you wish to build a different configuration (Release if you built Debug first, for example), then simply change the solution configuration as described in the previous paragraph, and build the INSTALL solution again.

In case of problems ...

Please notice that:

- While running CMake, all configure options are saved in file CMakeCache.txt, Before repeating all the build procedure (after fixing the errors), you should delete the whole content of the build folder and start again from step 3. Otherwise, values saved in the cache would interfere and cause problems. So move to folder to build and delete all its content before using CMake again.
- CMake has knowledge about the standard places to look for third party libraries, but my experience is that few Windows users know about these places and they install software here and there. As a consequence, sometimes it is difficult for my current CMake script to locate installed software, generating too many 'red lines' and forcing users to specify all needed paths. And this is not very user friendly.
- Probably my CMake script is a bullshit. But you are a good developer and building Lomse is just compiling the source code, including the headers for the required libraries. Probably you can prepare your own solution and project file for MS Visual Studio. That's what I do for building in Windows (yes, I don't use CMake). Or better, if you are good at MS Windows and CMake, and would like to help me to improve Lomse installation procedures for Windows, you are welcome. Please post a message in the Lomse list. Thank you.