

Lomse library installation instructions

Important

Lomse distribution does not include makefiles. All makefiles are generated with CMake build system. CMake can generate different kinds of native build files for your system (e.g. Unix Makefiles, Eclipse CDT 4.0 project files, Visual Studio project files).

You need CMake 2.8.0 or later on your platform. You can download it from <http://www.cmake.org/cmake/resources/software.html>. Just download and install it (very simple). In Linux, you can install it using the package manager. For instance, in Debian based systems do:

```
sudo apt-get install cmake
```

This document contains detailed instructions for building Lomse library and tests program from sources, on Linux and MS Windows. If you find things that need to be fixed or you install Lomse in other platforms and would like to contribute, you are welcome. Please post a message on the Lomse list. Thank you.

I have only generated and tested on following platforms:

- Windows XP, using Visual Studio .NET 2003 (VS7) (generator "Visual Studio 7 .NET")
- Linux Ubuntu 10.4, 11.04. Using CodeBlocks with linux makefiles (generator "CodeBlocks - Unix Makefiles")
- Linux Ubuntu 10.4, 11.04. Using linux makefiles (generator "Unix Makefiles")

No idea if there will be problems for other platforms/configurations. But if you try, please let me know for updating the documentation.

Table Of Content

- Requirements.
- Quick installation guide
- Building on Linux
- Building on Windows
- "In-source" vs. "Out-of-source" builds

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>)
- 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/>)
- libpng (<http://www.libpng.org/>)

Lomse installation

In Linux, normally these packages are already installed in your system, so normally you do not have to install them.

For Windows XP, pre-built libraries for these packages, except CMake, are included in the lomse distribution package, so the only thing you have to do is to download and install CMake.

For other MS Windows versions or other operating systems you will have to check if your system has these packages installed, and install any missing one. Please refer to each package website for instructions.

Quick installation guide

This procedure should work for any operating system supported by CMake, using CMake command-line interface. Do the following steps:

1. Download lomse source tree from SVN. Let's assume the root folder containing the source tree is at folder `"/projects/lomse"`.
2. (Recommended) Create a directory to hold your build files (e.g. `"/projects/lomse-build"`)
3. Change directory to the directory you created in the previous step.
4. Run `cmake` to generate the makefiles. For example, in linux:

For using CodeBlocs:

```
cmake -G "CodeBlocks - Unix Makefiles" -DCMAKE_BUILD_TYPE=Debug ../lomse
```

For using makefiles:

```
cmake -G "Unix Makefiles" -DCMAKE_BUILD_TYPE=Debug ../lomse
```

At this point you've generated a makefile or project files using `cmake`. Now, just run `'make'` or use your favorite IDE to build and install the library.

Building on Linux

You need CMake 2.8.0 or later.

First, you need to install the required packages. Usually the Boost libraries (<http://www.boost.org/>) and the FreeType library (<http://www.freetype.org/>) are already included in your Linux distro. So, usually, the only package you need to install is UnitTest++. Installing it should be as simple as typing the following commands in your terminal (on Debian-like systems):

```
sudo apt-get install libunittest++-dev
```

Now you can proceed to build Lomse.

In following instructions it is assumed you want to generate Makefiles. Open a terminal window, move to the folder in which you have unpackaged the lomse sources, and create a folder for building. For instance:

checkout a working copy:

Lomse installation

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

create folder to build:

```
cd ..
mkdir build-lomse
cd build-lomse
```

Now, using CMake, proceed to create the makefile:

```
cmake -G "Unix Makefiles" -DCMAKE_BUILD_TYPE=Debug ../lomse
```

Now, build the lomse library, the library unit-testing program (testlib) and the examples:

```
make

...
Linking CXX executable bin/testlib
Lomse library tests runner

Success: 832 tests passed.
Test time: 8.77 seconds.
[100%] Built target testlib
```

Run the example:

```
cd bin
./example_1
```

Alternatively, build step-by-step, checking results:

build the library:

```
make lomse

...
[100%] Built target lomse
```

build testlib and check library is ok:

```
make testlib

...
Lomse library tests runner

Success: 832 tests passed.
Test time: 8.77 seconds.
[100%] Built target testlib

The number of tests will grow over time, and the times reported will
depend on your computer.
```

build example application:

Lomse installation

```
make example_1

...
[100%] Built target example_1

cd bin
./example_1
```

In case of problems, before repeating al the build procedure (after fixing the errors), the makefile should be re-created. The safest way to proceed is to delete the whole content of the build folder and start againg from step 1. So move to folder to build and clean all:

```
cd ../build-lomse
rm * -r      #AWARE: BE SURE YOU ARE IN THE build-lomse FOLDER !!!!
```

and repeat build process from step 1.

When build is ok and test and example works, you can install lomse:

```
sudo make install
```

That installs Lomse on your system. By default, Lomse library is installed in [prefix]/lib and header files in [prefix]/include/lomse, with [prefix] defaulting to usr/local.

You can change the install location by running cmake like this:

```
cmake -G "Unix Makefiles" -DCMAKE_INSTALL_PREFIX=/new/install/prefix \
      -DCMAKE_BUILD_TYPE=Release ../lomse
```

Building from source in the repository is recommended, since code in the repository should always be stable. If it's not, that's a bug to report.

Building on Windows

Important

I'm no longer a Windows user and, therefore, I only use and test the installer in Windows, from time to time, to ensure that it still works for my old WindowsXP system. So, probably in your Windows system my CMake installer will fail. If you would like to contribute by fixing things or developing a stronger installer, you are welcome. Please post a message on the Lomse list.

You need CMake 2.8.0 or later.

In this description it is assumed you want to generate Visual Studio .NET 2003 project files. The procedure for another MS Visual Studio version or for another tool (i.e. Make) or IDE, such as CodeBlocks, is similar. Just replace Visual Studio with the desired target name.

To create the project file / makefile you have two options:

Lomse installation

1. Using the CMake GUI program.
2. Using a command window.

2.a) Using CMake GUI program.

1. Open CMake GUI program. (usually at C:\Program files\CMake\bin\cmake-gui.exe)
2. Enter requested data in following fields:

- Where is the source code: [path-to-lomse-root]
i.e.: D:/projects/lomse
- Where to build the binaries: [folder/to/place/makefiles]
i.e.: D:/projects/lomse/build

3. Press the 'Configure' button. A window will open and you will be asked for which tool you would like to generate the makefile. Select, for instance, "Visual Studio 7 .NET".
4. If there are lines marked in red color. press again the 'Configure' button. Repeat this until no red lines. It may happen that CMake won't find some needed libraries, such as zlib. If so you have to set CMAKE_PREFIX_PATH to point to the directory where the missing libraries are stored. For example, if CMake can't find ZLIB, and you have installed it at C:\software\zlib you will have add this path to CMAKE_PREFIX_PATH.
5. Press the 'Generate' button.

Done. Makefiles for the selected platform and generator will be created in the specified folder.

Now open the makefiles/project files and proceed to build the library and the test program. All tests will run automatically when the tests program is built.

2.b) Using a command window:

The procedure is similar, but using the command window.

1. Open the command window and move to the lomse root folder:

```
cd [path-to-lomse-root]
```

2. Create a new folder for the makefiles. i.e. [path-to-lomse-root]/build/

```
mkdir build
```

3. >Now generate the makefiles:

```
cmake -G "Visual Studio 7 .NET" ./build
```

It may happen that CMake won't find some needed libraries, such as zlib. If so you have to set CMAKE_PREFIX_PATH to point to the directory where the missing libraries are stored. For example, if CMake can't find ZLIB, and you have installed it at C:\software\zlib you will have add this path to CMAKE_PREFIX_PATH:

```
cmake -G "Visual Studio 7 .NET" cmake -D_ZLIB_DIR="C:\software\zlib" ./build
```

After running cmake it should create .sln and .vcproj files for Visual Studio in the 'build' directory.

Lomse installation

You can also generate project files for some other VS version. You can get a list of all supported generators by typing in and running cmake.

"In-source" vs. "Out-of-source" builds

Recommended way of installing Lomse: Out-of-source

When you generate the makefiles, they have to go somewhere. An "in-source" build puts them in your source tree (i.e. /lomse/build). In the next example the makefiles are placed in the same directory as the source code. Therefore, it is an "in-place" build:

```
cd lomse
mkdir build
cmake -G "Unix Makefiles" ./
make
```

An out-of-source build puts them in a completely separate directory, so that your source tree is unchanged. Out-of-source builds are recommended, as you can build multiple variants in separate directories, i.e.:

```
/projects
  /lomse           <---- the source tree
  /lomse-debug     <---- to create debug build
  /lomse-release   <---- to create release build
```

In the following example, an out-of-place build is performed:

```
cd lomse
mkdir lomse-debug
cd lomse-debug
cmake -G "Unix Makefiles" ../lomse
make
```

Note: Before performing an out-of-source build, ensure that all CMake generated in-source build information is removed from the source directory, e.g., CMakeFiles directory, CMakeCache.txt.