

# Installation and Testing of NMM (Mac OS X)

**Motama GmbH, Saarbruecken, Germany**  
**(<http://www.motama.com>)**

**April 2010**

Copyright (C) 2005-2010  
Motama GmbH, Saarbruecken, Germany  
<http://www.motama.com>

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with the Invariant Sections being all sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license can be found in the file COPYING.FDL.

THE DOCUMENT IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE DOCUMENT BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE DOCUMENT OR THE USE OR OTHER DEALINGS IN THE DOCUMENT.

This document gives an introduction on how to install and test the Open Source version of NMM for Mac OS X.

In this document, we assume that NMM package is called `nm-2.2.0.tar.gz` and will be extracted to directory `/Users/bob/nm-2.2.0`. Please replace these names as appropriate, e.g. with the name of the NMM package that you are actually using.

# 1. Requirements

## 1.1. Hardware and Software Requirements

You need a properly configured Mac OS X 10.6 operating system. On Mac OS X, most NMM examples and applications require a graphics board. So far NMM provides only a display component for Mac OS X using X11. Thus, the two packages X11User.pkg and X11SDK.pkg must be installed for video playback. Both packages are included on the CDs that are delivered with your Mac. This version of the display component accepts data in RGB format only. Furthermore, you must run any NMM application for video playback within an xterm to ensure that the X11 server is running. Finally, you need a sound board or chip that can natively playback different sampling rates such as 44.1 kHz. All other hardware, such as cameras, is optional.

## 1.2. Network Configuration

To allow one running NMM system to access another running NMM system, the port 22801 and the port range 5000-6000 must not be blocked by a firewall.

Please note that some Linux distributions include the line `127.0.0.2 <your-hostname>` into `/etc/hosts`. In this case you must replace the numeric IP address `127.0.0.2` with the real IP address of your system or you have to remove this line.

## 1.3. libtool

You have to use the libtool package delivered with your Mac.

# 2. Installation of NMM

## 2.1. Download NMM

Download NMM from here ([../nmmdownload.html](http://../nmmdownload.html)).

## 2.2. Extract

Extract the tar.gz package of release:

- `cd /Users/bob`

- `tar xvfz nmm-2.2.0.tar.gz` (will extract to `/Users/bob/nmm-2.2.0`)

The directory structure is described in Directory Structure (`../../nmmdocs_dirstruct.html`).

## 2.3. Configure

- Go to the NMM source directory, e.g.:

```
cd /Users/bob/nmm-2.2.0/
```

- If your Mac uses an Intel CPU, you have to enter the following command:

```
./configure --prefix=/Users/bob/nmm-2.2.0-installed \  
  --enable-macosx --disable-nasm \  
  --disable-all --enable-macosx-all
```

- If your Mac uses a PowerPC CPU, you have to enter the following command:

```
./configure --prefix=/Users/bob/nmm-2.2.0-installed \  
  --enable-macosx --disable-nasm --enable-ppc \  
  --disable-all --enable-macosx-all
```

Note that the options `--disable-all` and `--enable-macosx-all` disable all features for building and enable only those which are available on Mac OS X, provided that the required libraries are found. Also note that not all of these features have been fully tested and some of them might not work at all. We currently enable them in the "macosx-all" profile only because we see no reason why they should not work on Mac OS X at all.

As an alternative, you may specify only `--disable-all` and any combination of `--enable-...` switches you like. You can run `./configure --help` for a list of available switches.

You must specify an installation prefix on Mac OS X, because NMM will not work without running `make install` at least once. You can then run NMM from the source tree, but we recommend running NMM from the installation prefix instead.

If `./configure` terminated successfully, you will see a long list of NMM features, which are either enabled or disabled. If a feature is disabled, this is mostly due to unresolved dependencies. However, you should still be able to build and use NMM. All features can also be disabled manually. Please refer to `./configure --help` for a list of all options. In particular, you might want to set the installation directory for NMM by calling `./configure --prefix=<path for installing NMM>`. For example, we assume in the following that NMM is to be installed in `/Users/bob/nmm-2.2.0-installed/` and you want to compile with all features available under Mac OS X.

## **2.4. External Libraries**

Some external libraries are required for building the NMM base system. For building optional features of NMM (e.g. encoding/decoding plug-ins), many more external libraries are needed if these features should be available. In the output of the configure script, you can see what libraries were found or not, if the NMM base system can be compiled or not, and which optional features of NMM will be available. Before you try to compile NMM, you should make sure that all external libraries are available that are needed for building the NMM base system and all the optional features you want.

For a complete list of optional features and what libraries are required for each features, please refer to Features and Plug-ins of NMM ([../../nmmdocs\\_features.html](#)). For detailed information on all libraries required for building the NMM base system, please refer to External Libraries ([../../nmmdocs\\_external-libraries.html](#)). Detailed information on all libraries required for building optional features of NMM can be found here (<http://graphics.cs.uni-sb.de/NMM/Download/external/nmm-2.2.0/external-libraries-optional/index.htm>).

For each external library you need, you have the following options to obtain and install the library:

- Install the corresponding binary package of your Linux distribution
- Download the source code of the library and compile and install the library yourself

Note that not necessarily all options may be available for all libraries. A good strategy is to first try to install a binary package of each library you need and then try to compile the library yourself if the binary package does not work.

### **2.4.1. Option 1: Compiling Libraries from Sources**

Download, compile, and install the source code archives of external libraries. Source code archives of libraries required for building the NMM base system are available here ([../external-libraries/index.htm](#)). Source code archives of libraries required for building optional features of NMM are available here (<http://graphics.cs.uni-sb.de/NMM/Download/external/nmm-2.2.0/external-libraries-optional/index.htm>).

### **2.4.2. Option 2: Installing Developer Packages**

Install binary packages provided with your operating system (e.g. using `sudo port install`). You can use binary versions of external libraries if the versions match.

## 2.5. DYLD\_LIBRARY\_PATH

Set the environment variable `DYLD_LIBRARY_PATH` appropriately. The paths you specify in this variable must include all paths where external libraries needed by NMM are installed. For example, if you installed libraries in `/Users/bob`, you must set the variable as follows:

- `export DYLD_LIBRARY_PATH=/Users/bob/lib:$DYLD_LIBRARY_PATH` (for bash)
- `setenv DYLD_LIBRARY_PATH /Users/bob/lib:$DYLD_LIBRARY_PATH` (for tcsh)

## 2.6. Configure, Again

Run the configure script for the second time:

- `cd /Users/bob/nmm-2.2.0/`
- If your Mac uses an Intel CPU, use the command

```
./configure --prefix=/Users/bob/nmm-2.2.0-installed \  
  --with-extra-libs=/Users/bob/lib \  
  --with-extra-includes=/Users/bob/include \  
  --disable-nasm --enable-macosx \  
  --disable-all --enable-macosx-all
```
- If your Mac uses a PowerPC CPU, use the command

```
./configure --prefix=/Users/bob/nmm-2.2.0-installed \  
  --with-extra-libs=/Users/bob/lib \  
  --with-extra-includes=/Users/bob/include \  
  --enable-macosx --disable-nasm --enable-ppc \  
  --disable-all --enable-macosx-all \  
  --enable-ppc
```
- Additionally, you might want to add some options as printed out by `./configure --help`

## 2.7. Build

Build NMM by running:

- `make`
- `make parallel -j2` will usually speed up compile time. We also recommend `distcc` if you have more than one host available for compiling (hey, NMM is a 'networked' multimedia architecture.)
- If compilation fails, try `make -k` and see how far you get.

## 2.8. Install

Install NMM, either in the default directory, or in the directory specified using `./configure --prefix=<path for installing NMM>` by calling:

- `make install`
- If compilation failed in the previous, try `make install -k`

## 3. Testing NMM

You need to configure and test NMM. Before you can execute any binaries, you have to install nmm like described in last paragraph. All following steps should be performed on a xterm to be able do display video.

### 3.1. Environment Variables

You might want to permanently set the `DYLD_LIBRARY_PATH` as described above: For example, extend your personal `~/tcshrc` (for `tcsh`) or `~/bashrc` (for `bash`)

If you are using the installed version of NMM (i.e. `make install`), which is recommended, you do not need to set any additional environment variable. If, however, you would like to run NMM from the source directory, you need to set the `NMM_DEV_DIR`:

- `export NMM_DEV_DIR=/Users/bob/nmm-2.2.0` (for `bash`)
- `setenv NMM_DEV_DIR /Users/bob/nmm-2.2.0` (for `tcsh`)

### 3.2. NMM Registry

Setup the NMM registry:

- Change to directory of NMM registry:
  - `cd /Users/bob/nmm-2.2.0-installed/bin` if you are using the installed version of NMM (recommended).
- Run `./serverregistry -s` and wait until all plug-in information is generated. This step is also performed automatically when you start some NMM application or example for the first time.

### 3.3. Test Audio/Video Rendering

Test NMM using the application called 'clirc', which is a very powerful tool.

- Change to the directory of 'clirc'.
  - `cd /home/bob/nmm-2.2.0-installed/bin` if you are using the installed version of NMM (recommended)
- Test audio rendering by playing back a WAV file.
  - `./clirc ../share/nmm/gd/macosx/playback/audio/wavplay.gd -i <some wav file>` if you are using the installed version of NMM (recommended)

You should hear the WAV file being played back.

- Test video rendering.
  - `./clirc ../share/nmm/gd/macosx/playback/video/noise_rgb.gd` if you are using the installed version of NMM (recommended).

You should see a window showing some white noise.

If everything works fine, you might want to read the documentation on clic and check out the other graph descriptions (GD files) available for your platform.

### 3.4. Security

All security settings are optional, but recommended.

- Copy `/Users/bob/nmm-2.2.0/resources/nmmrc_sample` to your home directory as `.nmmrc` and edit it:
  - `cp /Users/bob/nmm-2.2.0/resources/nmmrc_sample ~/.nmmrc`
- By setting `allowedreadpaths` you can restrict the paths from which plug-ins are allowed to read data, e.g. your MP3 files.
- By setting `allowedwritepaths` you can restrict the paths to which plug-ins are allowed to write data.
- By setting a `passwd` you can restrict access between NMM processes (and therefore systems). Only processes using the same password are allowed to interact. For example, if you start a `serverregistry` on host A and another user at host B wants to connect to this `serverregistry`, both of you need to agree on the same password.

If you are behind a firewall and only connected to trusted hosts and users, you do not necessarily need these settings at all.

If you want to access devices, e.g. `/dev/dsp` for audio output, you need to add this to the 'allowed' paths. Remember: the usual restrictions of your operating system still apply, e.g. adding `/dev/` to `allowedreadpaths` and `allowedwritepaths` is not necessarily unsafe.