Installation and Testing of NMM (Linux)

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

April 2010

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

In this document, we assume that NMM package is called `nmm-2.2.0.tar.gz` and will be extracted to directory `/home/bob/nmm-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 Requirements

You need a properly configured operating system, e.g. a Linux on a PC or some other platform. On Linux, most NMM examples and applications require a graphics board with configured Xv extension (refer to the output of `xvinfo`) and 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.

2. Installation of NMM

2.1. Download NMM

Download NMM from here (../../../nmmdownload.html).

2.2. Extract

Extract the tar.gz package of release:

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

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

2.3. Configure

Run configure script for the first time:

- `cd /home/bob/nmm-2.2.0/`
- `./configure`

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 `/home/bob/nmm-2.2.0-installed/`

- `./configure --prefix=/home/bob/nmm-2.2.0-installed/`

2.4. External Libraries

No external libraries are required for building the NMM base system on Linux. However, for building optional features of NMM (e.g. encoding/decoding plug-ins), many 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). 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 developer package of your Linux distribution
- Use a precompiled library
- 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. The easiest way to get all external libraries you need is to install the precompiled libraries for your platform and use the developer packages provided by your distribution for all remaining libraries.

2.4.1. Option 1: Using precompiled libraries (recommended)

Download the pre-compiled versions and corresponding headers of external libraries for your platform. Precompiled 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). Note that no external libraries are required for building the NMM base system on Linux, so we do not provide any.

Please note that some external libraries are not included in these packages. Please use one of the other two options to make these libraries available. Precompiled external libraries are available only for 32 bit linux systems. For 64 bit systems, this option is currently not available at all.

Please note that these pre-compiled libraries are only provided for your convenience. On some systems, these pre-compiled libraries might not work at all, and therefore NMM will not work using these libraries. You are free to not use these pre-compiled versions, and choose option 2 or 3 as described below. However, there is a good chance that these libraries will work on your system. In this case, this is the fastest and easiest way to install NMM.

If you want to use the provided libraries, you need to perform following steps:

- `cd /home/bob/`
- `tar xvfz nmm-2.2.0-optional-external-libs-linux.tar.gz`

2.4.2. Option 2: Compiling Libraries from Sources

Download, compile, and install the source code archives of external libraries. 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).

2.4.3. Option 3: Installing Developer Packages

Install the corresponding development packages provided by your GNU/Linux distribution. Note that many distributions provide the header files of a library in a package separate from the library itself, so-called developer packages. You need to install the developer package of a library to be able to use a library for building NMM. Developer packages typically have package names like `"libfoo-devel"` or `"libfoo-dev"`.

If the version of the library provided by your distribution does not match the version of the library listed in our external library list, then there is no guarantee that NMM will work with this version of the library. If you encounter problems, please use one of the other two options instead.

2.5. LD_LIBRARY_PATH

Set the environment variable LD_LIBRARY_PATH appropriately. The paths you specify in this variable must include all paths where external libraries needed by NMM are installed. If you chose Option 1 for external libraries, this results in:

- `export LD_LIBRARY_PATH=/home/bob/nmm-2.2.0-external-libs/lib:$LD_LIBRARY_PATH` (for bash)
- `setenv LD_LIBRARY_PATH /home/bob/nmm-2.2.0-external-libs/lib:$LD_LIBRARY_PATH` (for tcsh)

2.6. Configure, Again

Run the configure script for the second time:

- `cd /home/bob/nmm-2.2.0/`
- Call `./configure --with-extra-libs=<path to extracted libs> --with-extra-includes=<path to extracted includes> --prefix=<path for installing NMM>`
- If you chose option 1 in step 5, call `./configure --with-extra-libs=/home/bob/nmm-2.2.0-external-libs/ --with-extra-includes=/home/bob/nmm-2.2.0-external-libs/include --prefix=/home/bob/nmm-2.2.0-installed/`
- 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.

3.1. Environment Variables

You might want to permanently set the LD_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=/home/bob/nmm-2.2.0` (for bash)
- `setenv NMM_DEV_DIR /home/bob/nmm-2.2.0` (for tcsh)

3.2. NMM Registry

Setup the NMM registry:

- Change to directory of NMM registry:
  - `cd /home/bob/nmm-2.2.0-installed/bin` if you are using the installed version of NMM (recommended).
  - `cd /home/bob/nmm-2.2.0/apps/registry` if you want to run NMM from the source directory.
- 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 'clic', which is a very powerful tool.

- Change to the directory of 'clic'.
  - `cd /home/bob/nmm-2.2.0-installed/bin` if you are using the installed version of NMM (recommended)
  - `cd /home/bob/nmm-2.2.0/apps/clic` if you want to run NMM from the source directory

- Test audio rendering by playing back a WAV file.
  - `./clic ../share/nmm/gd/linux/playback/audio/wavplay.gd -i <some wav file>` if you are using the installed version of NMM (recommended)
  - `./clic gd/linux/playback/audio/wavplay.gd -i <some wav file>` if you want to run NMM from the source directory

  You should heat the WAV file being played back.

- Test video rendering.
  - `./clic ../share/nmm/gd/linux/playback/video/noise_yv12.gd` if you are using the installed version of NMM (recommended)
  - `./clic gd/linux/playback/video/noise_yv12.gd` if you want to run NMM from the source directory

  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 `/home/bob/nmm-2.2.0/resources/nmmrc_sample` to your home directory as `.nmmrc` and edit it:
  - `cp /home/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.