

## FreeNX Manual Installation How-To

In light of my own difficulties installing FreeNX and the lack of a concise step-by-step guide, I decided to write one. The following is the sum total of several weeks of experimenting, failing, reformatting, starting over, reloading and starting over again.

**Disclaimer:** I make no guarantee that this is the right, or even the only way to install FreeNX and the accompanying NX libraries. All of my experience is with the Centos 4 and Centos 5 Linux distributions so please take the differences of your distro into account when following these instructions. I am not responsible if you happen to screw up and break something. You should always test new software on a non-production server that can be easily sacrificed to a reformat if things go terribly wrong. This howto is based on FreeNX 0.7.1 and the 3.0.0 NX Libraries

I cannot begin a FreeNX how-to without first saying thanks to Fabian Franz for his work making FreeNX a reality.

### Requirements:

You will need the following NX pieces from nomachine.com:

```
nxclient-3.0.0
nxnode-3.0.0
nxserver-3.0.0
nxcomp-3.0.0
nxcompext-3.0.0
nxcompshad-3.0.0
nxesd-3.0.0
```

These procedures should be performed as root. We will start the procedure by creating an nx directory under /usr/src.

```
[server ]# mkdir /usr/src/nx
[server ]# cd /usr/src/nx
```

Now we download the pieces from nomachine:

```
[server ]# wget http://64.34.161.181/download/3.0.0/Linux/nxclient-3.0.0-84.i386.tar.gz
[server ]# wget http://64.34.161.181/download/3.0.0/Linux/nxnode-3.0.0-88.i386.tar.gz
[server ]# wget http://64.34.161.181/download/3.0.0/Linux/FE/nxserver-3.0.0-74.i386.tar.gz
```

These files should be extracted to the /usr directory.

```
[server ]# cd /usr
[server ]# tar -xzf /usr/src/nx/nxserver-3.0.0-74.i386.tar.gz
[server ]# tar -xzf /usr/src/nx/nxclient-3.0.0-84.i386.tar.gz
[server ]# tar -xzf /usr/src/nx/nxnode-3.0.0-88.i386.tar.gz
```

This will create the /usr/NX/ and the appropriate subfolders with the various files in the appropriate places.

Now we download and compile the libraries:

First we switch back to our staging area in the /usr/src folder then download and compile.

```
[server ]# cd /usr/src/nx
[server ]# wget http://64.34.161.181/download/3.0.0/sources/nxcomp-3.0.0-48.tar.gz
[server ]# tar -xzf nxcomp-3.0.0-48.tar.gz
[server ]# cd nxcomp
[server ]# ./configure
[server ]# make
[server ]# cp --preserve libXcomp.so* /usr/NX/lib
[server ]# cd ..
```

(The --preserve flag to cp is necessary to keep the symlinks as symlinks)

The nxcompext library requires some X11 libraries to be on your system. Under Centos 5 the headers and libraries are in a couple of different places. Not being an expert at compiling software I came up with a way to pass the appropriate options to the configure script rather than editing the makefile after configure. Take note of the following configure line the following block and modify it for your distribution as needed. If nxcompext doesn't compile and complains of missing files, use locate to find them and change the paths passed to configure.

```
[server ]# wget http://64.34.161.181/download/3.0.0/sources/nxcompext-3.0.0-18.tar.gz
[server ]# tar -xzf nxcompext-3.0.0-18.tar.gz
[server ]# cd nxcompext
[server ]# ./configure --x-includes="/usr/include/xorg -I/usr/include/X11"
[server ]# make
[server ]# cp --preserve libXcompext.so* /usr/NX/lib
[server ]# cd ..
```

**Note:** Before Building nxcompext on Centos 5 I had to make a modification to NXlib.c before I could run the make command. I had to add:

```
#define _XGetIOError(dpy) \
    (dpy -> flags & XlibDisplayIOError)
```

between #include MD5.h and #define PANIC, otherwise this line will show up in my nxserver logfile and I would be unable to connect:

```
/usr/NX/bin/nxagent: symbol lookup error: /usr/NX/lib/libXcompext.so.3: undefined
symbol: _XGetIOError
```

You may or may not need to add this line on your distribution, but you won't know until you build and test.

The nxcomshad library is next:

```
[server ]# wget http://64.34.161.181/download/3.0.0/sources/nxcompshad-3.0.0-19.tar.gz
[server ]# tar -xzf nxcompshad-3.0.0-19.tar.gz
[server ]# cd nxcompshad
[server ]# ./configure
[server ]# make
[server ]# cp --preserve libXcompshad.so* /usr/NX/lib
[server ]# cd ..
```

If you will be using the audio redirection to the client machine you will need the NX sound server libraries. If you do not care about audio redirection you can skip this block.

```
[server ]# wget http://64.34.161.181/download/3.0.0/sources/nxesd-3.0.0-4.tar.gz
[server ]# tar -xzf nxesd-3.0.0-4.tar.gz
[server ]# cd nxesd
[server ]# make
[server ]# make install
[server ]# cd ..
```

Now that we have all the libraries built and put where they are supposed to go, we'll download the FreeNX packages and install them. Note the "patch" command. The patch must be applied if you are using the /usr/NX directory structure that the nomachine libs expect.

```
[server ]# wget http://download2.berlios.de/freenx/freenx-0.7.1.tar.gz
[server ]# tar -xzf freenx-0.7.1.tar.gz
[server ]# cd freenx-0.7.1
[server ]# patch -p0 < gentoo-nomachine.diff
[server ]# cp -f nxkeygen /usr/NX/bin/
[server ]# cp -f nxloadconfig /usr/NX/bin/
[server ]# cp -f nxnode /usr/NX/bin/
[server ]# cp -f nxnode-login /usr/NX/bin/
[server ]# cp -f nxserver /usr/NX/bin
[server ]# cp -f nxsetup /usr/NX/bin
[server ]# cp -f nxcups-gethost /usr/NX/bin
```

One piece of the FreeNX package needs to be compiled so we do that now:

```
[server ]# cd nxserver-helper
[server ]# make
```

Now that that piece is compiled we continue copying things where they need to go and making a few symlinks. Note that some of these are distro-specific and may need to be adjusted for your distribution

```
[server ]# cp -f nxserver-helper /usr/NX/bin/
[server ]# cd ..
[server ]# ln -s /usr/NX/bin/nxserver /usr/bin/nxserver
[server ]# ln -s /usr/NX/bin/nxsetup /usr/sbin/nxsetup
[server ]# ln -s /usr/NX/bin/nxloadconfig /usr/sbin/nxloadconfig
[server ]# ln -s /usr/NX/lib/libXrender.so.1.2.2 /usr/NX/lib/libXrender.so.1.2
[server ]# ln -s /usr/NX/bin/nxagent /usr/NX/bin/nxdesktop
[server ]# ln -s /usr/NX/bin/nxagent /usr/NX/bin/nxviewer
[server ]# ln -s /usr/bin/foomatic-ppdfile /usr/lib/cups/driver/foomatic-ppdfile
[server ]# ln -s /etc/X11/xinit /etc/X11/xdm
[server ]# ln -s /sbin/mount.cifs /sbin/smbmount
[server ]# ln -s /sbin/umount.cifs /sbin/smbumount
```

At this point you would normally run `nxsetup --install` and complete the installation. On Centos 5, however, you need to make some modifications to `/usr/NX/bin/nxsetup` before you can continue. These may not be appropriate changes and I take no responsibility if they break something.

For my `nxsetup` to complete successfully I had to edit `/usr/NX/bin/nxsetup` and comment out all of the

```
[ -f /etc/nscd.conf ] && { nscd --invalidate group; }
```

lines. After completeing all of these steps your NX setup should be complete and functional.

**NOTE:** On multi-monitor systems the `nxclient` will not display rootless mode windows properly unless they are run on the primary monitor of the video card used to boot the system. This is not to be confused with the monitor designated as Primary under the XP/Vista display properties. What NX considers the primary monitor and what the OS considers the primary monitor are two very different things. Some research on the `nomachine` site leads me to believe that this behavior is due to the fact that the Windows NoMachine client contains pieces of the Window version of the X.Org server, and the X.Org server does not currently handle multiple monitors correctly when running under a Microsoft OS.

I hope this how-to helps others avoid some of the confusion I encountered during my initial foray into the wonderful world FreeNX.