en

NOMACHINE		Introduction to NX software components	
Prepared by: Silvia Regis		*N°:* D-708_002-SFW-CMP	
Approved by: Sarah Dryell	*Signature:*	*Date:* 23/08/2007	*Amended:* A

Introduction to NX software components

Index

1. Introduction to NX software components

This document offers a brief introduction to tools and libraries that make up the NX Server and NX Client. This document is destined for system administrators that want to understand the interaction and dependencies between NX components, to fix installation problems or provide help to their customers.

1.1. NX Installation Directories

NX Server products for Linux Operating System are provided as RPM or DEB packages or tar.gz archives, while NX Server products for Solaris SPARC are provided as sparc.gz packages. You may install them by using the sudo utility or as superuser (root).

Installation of NX Server requires the download and installation of three packages: client, node and server. The client is needed because it ships libraries used by the node. The node is needed because it ships tools needed by the server.

More detailed installation instructions are available at:

http://www.nomachine.com/installation.php

If installation was succesful, the node and server post-install scripts will be automatically executed in background, without user intervention. The post-install scripts will perform all the basic configuration steps needed to enable users to login to the new NX server.

All NX software components are installed under the NX root directory, usually /usr/NX. This is how your /usr/NX directory should appear after installation:

```
# ls -la /usr/NX
drwxr-xr-x  9 root root 4096 2007-08-02 15:03 .
drwxr-xr-x 14 root root 4096 2007-08-03 17:14 ..
drwxr-xr-x  2 root root 4096 2007-08-03 17:13 bin
drwxr-xr-x  3 nx   root 4096 2007-08-03 18:40 etc
drwxr-xr-x  3 root root 4096 2007-06-19 22:08 home
drwxr-xr-x  3 root root 4096 2007-08-02 15:03 lib
drwxr-xr-x  5 root root 4096 2007-08-02 15:03 scripts
drwxr-xr-x 13 root root 4096 2007-08-02 15:03 share
drwxr-xr-x  5 nx   root 4096 2007-08-02 15:03 var
```

This is how your /usr/NX/bin directory should appear:

```
# ls -la /usr/NX/bin/
drwxr-xr-x 2 root root    4096 2007-08-03 17:13 .
drwxr-xr-x 9 root root    4096 2007-08-02 15:03 ..
-rwxr-xr-x 1 root root 4507816 2007-08-03 13:23 nxagent
-rwxr-xr-x 1 root root 5995184 2007-08-03 11:37 nxclient
-rwxr-xr-x 1 root root   55128 2007-08-03 11:37 nxesd
-rwxr-xr-x 1 root root   12380 2007-08-03 11:37 nxkill
-rwxr-xr-x 1 root root 7945680 2007-08-03 13:23 nxnode
-rwxr-xr-x 1 root root    6912 2007-08-03 11:37 nxprint
-rwxr-xr-x 1 root root  145524 2007-08-03 13:23 nxsensor
```

```
-rwxr-xr-x 1 root root 12662064 2007-08-02 15:03 nxserver
-rwxr-xr-x 1 root root    24028 2007-08-03 11:37 nxservice
-rwxr-xr-x 1 root root   767968 2007-08-03 13:23 nxspool
-rwxr-xr-x 1 root root   302852 2007-08-03 11:37 nxssh
-rwxr-xr-x 1 root root 11144848 2007-08-02 15:03 nxstat
-r-sr-xr-x 1 root root    11528 2007-08-03 13:23 nxuexec
```

Basic NX server software components are the *nxserver* and *nxnode* command line tools. They implement terminal server functionalities on top of Linux or any other Unix operating system. They are not "daemons", strictly speaking, as they are activated by the SSH server running on the system. To make a comparison, they leverage SSH exactly as a Web application using the Apache Web server to handle communication with clients using the HTTP protocol.

Other NX server software components are the advanced NX proxying agent, *nxagent* and *nxssh*.

- *nxagent* implements encapsulation and protocol translation of X.
- *nxssh* handles secure access to the server's and nodes' resources.
- Some other additional tools, like *nxuexec* and *nxspool* help NX server to handle its tasks.
- While *nxsensor* and *nxstat* daemons are used by NX Server Manager, the graphical administration interface of NX Server, to retrieve and display NX statistics.

A NX Client suitable for the target server platform is also needed to handle dialog boxes inside sessions (or through a remote execution mechanism on clients) and allows the user to perform basic administrative operations (like killing sessions or displaying statistics). Basic NX client components are *nxclient* (a multifunctional GUI manager) and *nxssh* (the SSH client used to login to the remote NX server).

On clients, *nxauth*, which is a command line tool based on xauth, replaces the standard *xauth* that comes with X distributions. *nxauth* is able to handle X authorization cookies without the need for X11 libraries, saving space in installation packages.

1.2. NX Libraries Installation Directory

This is how your /usr/NX/lib directory should appear:

```
# ls -la /usr/NX/lib/
drwxr-xr-x 3 root root    4096 2007-08-02 15:03 .
drwxr-xr-x 9 root root    4096 2007-08-02 15:03 ..
lrwxrwxrwx 1 root root 18 2007-08-03 17:12 libcrypto.so -> libcrypto.so.0.9.8
-rwxr-xr-x 1 root root 1092172 2007-08-03 11:37 libcrypto.so.0.9.8
-rwxr-xr-x 1 root root   22321 2007-05-30 11:31 libesddsp.so.0
-rwxr-xr-x 1 root root   80955 2007-05-30 11:31 libesd.so.0
```

```
lrwxrwxrwx 1 root root      17 2007-08-03 17:13 libexpat.so -> libexpat.so.0.4.0
lrwxrwxrwx 1 root root      17 2007-08-03 17:13 libexpat.so.0 -> libexpat.so.0.4.0
-rwxr-xr-x 1 root root  159376 2007-08-03 13:23 libexpat.so.0.4.0
lrwxrwxrwx 1 root root      20 2007-08-03 17:13 libfreetype.so -> libfreetype.so.6.3.7
lrwxrwxrwx 1 root root      20 2007-08-03 17:13 libfreetype.so.6 -> libfreetype.so.6.3.7
-rwxr-xr-x 1 root root  522904 2007-08-03 13:23 libfreetype.so.6.3.7
lrwxrwxrwx 1 root root      17 2007-08-03 17:12 libjpeg.so -> libjpeg.so.62.0.0
lrwxrwxrwx 1 root root      17 2007-08-03 17:12 libjpeg.so.62 -> libjpeg.so.62.0.0
-rwxr-xr-x 1 root root  126672 2007-08-03 11:37 libjpeg.so.62.0.0
lrwxrwxrwx 1 root root      13 2007-08-03 17:12 libpng12.so -> libpng12.so.0
lrwxrwxrwx 1 root root      19 2007-08-03 17:12 libpng12.so.0 -> libpng12.so.0.1.2.8
-rwxr-xr-x 1 root root  238336 2007-08-03 11:37 libpng12.so.0.1.2.8
lrwxrwxrwx 1 root root      13 2007-08-03 17:13 libX11.so -> libX11.so.6.2
lrwxrwxrwx 1 root root      13 2007-08-03 17:13 libX11.so.6 -> libX11.so.6.2
-rwxr-xr-x 1 root root  904200 2007-08-03 13:23 libX11.so.6.2
lrwxrwxrwx 1 root root      20 2007-08-03 17:13 libXcompext.so -> libXcompext.so.3.0.0
lrwxrwxrwx 1 root root      20 2007-08-03 17:13 libXcompext.so.3 -> libXcompext.so.3.0.0
-rwxr-xr-x 1 root root   53676 2007-08-03 13:23 libXcompext.so.3.0.0
lrwxrwxrwx 1 root root      20 2007-08-03 17:13 libXcomposite.so -> libXcomposite.so.1.0
lrwxrwxrwx 1 root root      20 2007-08-03 17:13 libXcomposite.so.1 ->libXcomposite.so.1.0
-rwxr-xr-x 1 root root    6480 2007-08-03 13:23 libXcomposite.so.1.0
lrwxrwxrwx 1 root root      21 2007-08-03 17:13 libXcompshad.so -> libXcompshad.so.3.0.0
lrwxrwxrwx 1 root root      21 2007-08-03 17:13 libXcompshad.so.3 ->libXcompshad.so.3.0.0
-rwxr-xr-x 1 root root   35668 2007-08-03 13:23 libXcompshad.so.3.0.0
lrwxrwxrwx 1 root root      19 2007-08-03 17:12 libXcompsh.so -> libXcompsh.so.3.0.0
lrwxrwxrwx 1 root root      19 2007-08-03 17:12 libXcompsh.so.3 -> libXcompsh.so.3.0.0
-rwxr-xr-x 1 root root   76732 2007-08-03 11:37 libXcompsh.so.3.0.0
lrwxrwxrwx 1 root root      17 2007-08-03 17:12 libXcomp.so -> libXcomp.so.3.0.0
lrwxrwxrwx 1 root root      17 2007-08-03 17:12 libXcomp.so.3 -> libXcomp.so.3.0.0
-rwxr-xr-x 1 root root  936540 2007-08-03 11:37 libXcomp.so.3.0.0
lrwxrwxrwx 1 root root      17 2007-08-03 17:13 libXdamage.so -> libXdamage.so.1.0
lrwxrwxrwx 1 root root      17 2007-08-03 17:13 libXdamage.so.1 -> libXdamage.so.1.0
-rwxr-xr-x 1 root root    6440 2007-08-03 13:23 libXdamage.so.1.0
lrwxrwxrwx 1 root root      14 2007-08-03 17:13 libXext.so -> libXext.so.6.4
lrwxrwxrwx 1 root root      14 2007-08-03 17:13 libXext.so.6 -> libXext.so.6.4
-rwxr-xr-x 1 root root   65344 2007-08-03 13:23 libXext.so.6.4
lrwxrwxrwx 1 root root      16 2007-08-03 17:13 libXfixes.so -> libXfixes.so.3.0
lrwxrwxrwx 1 root root      16 2007-08-03 17:13 libXfixes.so.3 -> libXfixes.so.3.0
-rwxr-xr-x 1 root root   14432 2007-08-03 13:23 libXfixes.so.3.0
lrwxrwxrwx 1 root root      15 2007-08-03 17:13 libXft.so -> libXft.so.2.1.2
lrwxrwxrwx 1 root root      15 2007-08-03 17:13 libXft.so.2 -> libXft.so.2.1.2
-rwxr-xr-x 1 root root   81436 2007-08-03 13:23 libXft.so.2.1.2
lrwxrwxrwx 1 root root      16 2007-08-03 17:13 libXrandr.so -> libXrandr.so.2.0
lrwxrwxrwx 1 root root      16 2007-08-03 17:13 libXrandr.so.2 -> libXrandr.so.2.0
-rwxr-xr-x 1 root root   10192 2007-08-03 13:23 libXrandr.so.2.0
lrwxrwxrwx 1 root root      19 2007-08-03 17:13 libXrender.so -> libXrender.so.1.2.2
lrwxrwxrwx 1 root root      19 2007-08-03 17:13 libXrender.so.1 -> libXrender.so.1.2.2
-rwxr-xr-x 1 root root   32000 2007-08-03 13:23 libXrender.so.1.2.2
lrwxrwxrwx 1 root root      14 2007-08-03 17:13 libXtst.so -> libXtst.so.6.1
lrwxrwxrwx 1 root root      14 2007-08-03 17:13 libXtst.so.6 -> libXtst.so.6.1
-rwxr-xr-x 1 root root   22608 2007-08-03 13:23 libXtst.so.6.1
lrwxrwxrwx 1 root root      13 2007-08-03 17:12 libz.so -> libz.so.1.2.3
lrwxrwxrwx 1 root root      13 2007-08-03 17:12 libz.so.1 -> libz.so.1.2.3
-rwxr-xr-x 1 root root   69484 2007-08-03 11:37 libz.so.1.2.3
drwxr-xr-x 3 root root    4096 2007-08-03 17:13 perl
```

libXcomp, the NX transport and compression library, is the core of NX technology. It implements new semantics of X client-server connections. It allows an X client like Mozilla, for example, to "listen" for connections from the X server. Or, as in NX sessions, it enables agents to wait for a client's connections from the NX proxy. *libXcomp* also handles compression of X protocol and provides implementation of *libXcompext* extensions to the X protocol in a way that is transparent to the real X server.

libXcompext library provides *nxagent* with a user-friendly wrapper to some of X compression and optimization capabilities built in *libXcomp*, while the modified libraries *libX11* and *libXext* implement NX transport layer functions and other minor optimizations. Among them is a cleanup of padding bytes of core X protocol requests and images, enhancements that are vital to compression and caching effectiveness of NX.

libXcompshad library provides *nxagent* with the screen-exporting functions needed to implement the desktop sharing functionality, i.e. it provides functions to monitor the changed areas of the screen (initially on a generic X server, later on Windows and Mac) and to display the image of the modified areas on a remote X server, by a simple XPutImage.

Other shared libraries shipped by the node package, as *libXdamage*, *libXrandr* and *libXrender* or shipped by the client package, as *libjpeg*, *libpng* and *libz* ensure that the NX software can be installed and run on the widest range of Linux distributions.

The *perl* directory contains critical perl libraries and modules that NX server and NX node must find in the target environment.

2. The Other NX Directories

Here is the /usr/NX/etc directory:

```
# ls -al /usr/NX/etc/
drwxr-xr-x  3 nx   root  4096 Aug  6 18:52 .
drwxr-xr-x  9 root root  4096 Aug  6 18:52 ..
-rw-------  1 nx   root     0 Aug  6 18:52 administrators.db
-rw-r--r--  1 nx   root     0 Aug  6 18:52 administrators.db.lock
-rw-r--r--  1 root root     0 Aug  2 15:03 administrators.db.sample
-rw-------  1 nx   root     0 Aug  6 18:52 guests.db
-rw-r--r--  1 nx   root     0 Aug  6 18:52 guests.db.lock
-rw-r--r--  1 root root     0 Aug  2 15:03 guests.db.sample
drwxr-xr-x  2 nx   root  4096 Aug  6 18:52 keys
-rw-r--r--  1 root root 13781 Aug  6 18:52 node.cfg
-rw-r--r--  1 root root 13268 Aug  3 13:23 node-debian.cfg.sample
-rw-r--r--  1 root root 13270 Aug  3 13:23 node-fedora.cfg.sample
-r--------  1 root root   679 Aug  6 18:52 node.lic
-rw-r--r--  1 root root 13270 Aug  3 13:23 node-mandrake.cfg.sample
-rw-r--r--  1 root root 13270 Aug  3 13:23 node-redhat.cfg.sample
-rw-r--r--  1 nx   root     0 Aug  6 18:52 nodes.db
-rw-r--r--  1 root root 13266 Aug  3 13:23 node-suse.cfg.sample
-rw-------  1 nx   root     0 Aug  6 18:52 passwords.db
-rw-r--r--  1 nx   root     0 Aug  6 18:52 passwords.db.lock
-rw-r--r--  1 root root     0 Aug  2 15:03 passwords.db.sample
-rw-------  1 nx   root     0 Aug  6 18:52 profiles.db
-rw-r--r--  1 nx   root     0 Aug  6 18:52 profiles.db.lock
-rw-r--r--  1 root root     0 Aug  2 15:03 profiles.db.sample
-rw-r--r--  1 root root 22108 Aug  6 18:52 server.cfg
-rw-r--r--  1 root root 22108 Aug  2 15:03 server-debian.cfg.sample
-rw-r--r--  1 root root 22108 Aug  2 15:03 server-fedora.cfg.sample
-r--------  1 nx   root   673 Aug  6 18:52 server.lic
-rw-r--r--  1 root root 22108 Aug  2 15:03 server-mandrake.cfg.sample
-rw-r--r--  1 root root 22108 Aug  2 15:03 server-redhat.cfg.sample
-rw-r--r--  1 root root 22108 Aug  2 15:03 server-suse.cfg.sample
-rw-------  1 nx   root     0 Aug  6 18:52 users.db
```

```
-rw-r--r--  1 nx   root     0 Aug  6 18:52 users.db.lock
-rw-r--r--  1 root root     0 Aug  2 15:03 users.db.sample
```

The *etc* directory contains the NX server and NX node configuration files, i.e. *server.cfg* and *node.cfg*, the NX databases for managing users and nodes, and the SSL keys, stored in the *keys* subdirectory, used by the server to access the node's resources on behalf of NX users.

Using the NX terminology, a node is the host that runs user sessions. In the case of NX Advanced Server product, there can be multiple nodes, while in all the other NX Server implementations only one node is present in the system, where the NX Server have been installed.

The *var/db* directory contains either the sessions databases and some temporary files used, for example, to send messages to some or all running sessions, while the *var/log* directory contains log files produced during the installing or updating of the NX node and server. Finally, the *var/run* directory is used to store pids of the running daemons, as for example nxsensor and nxstat, used to elaborate NX statistics about either the host machine and the NX sessions. These statistics are displayed via the NX Server Manager Web tool.

The *share* directory contains images and icons used by NX Client. The *share/documents* directory, instead, is used to store all the documents provided by the NX software products, as the NoMachine EULA and additional information to activate, for example, the CUPS backend.

The *scripts/init* directory contains those scripts to start and stop the server and the other NX daemons, as *nxstat*, at machine reboot/shut-down. The privileged scripts run by NX server to perform opearions that requires administrative privileges are stored in the *scripts/restriced* directory. While the NX node and server setup scripts are available in the *scripts/setup* directory.

The last directory is home. It is the home directory of nx user. The nx user is the account used by NX server to perform most operations. It corresponds to user nobody or www of Apache web server. The nx account is unprivileged. This ensures that in case of severe malfunctions of NX's software or exploited security bugs, an attacker cannot take control of the whole server.

3. The User NX Directory

NX Client uses the *.nx* directory in the user's home directory to store all its configuration files. This same directory is used by NX server to store temporary files and the session logs at the time a new session is being run. The User NX directory on client side can be configured via the NX Client GUI -> Environment panel, while, on server side, the NX node configuration file allows to specify a .nx directory different that the default one.

Let's go through the .nx directory of the guest user *test0766* on NoMachine TestDrive*:*

```
test0766@test00:~/.nx> ls -la
total 44
drwxr-xr-x 11 test0766 nxusers 4096 2007-08-13 15:30 .
drwxr-xr-x 20 test0766 nxusers 4096 2007-08-13 15:30 ..
drwxr-xr-x  2 test0766 nxusers 4096 2007-08-13 14:14 cache-gnome
drwxr-xr-x  2 test0766 nxusers 4096 2007-08-13 15:28 cache-kde
drwxr-xr-x  2 test0766 nxusers 4096 2007-08-13 15:30 cache-shadow
drwxr-xr-x  2 test0766 nxusers 4096 2007-08-13 15:29 cache-vnc
drwxr-xr-x  2 test0766 nxusers 4096 2007-08-13 15:28 cache-windows
drwx------  2 test0766 nxusers 4096 2007-08-13 15:27 config
drwxr-xr-x  4 test0766 nxusers 4096 2007-08-13 14:14 C-test00-1183-A76C562ECFA489A...
drwx------  2 test0766 nxusers 4096 2007-08-13 15:30 S-test00-1195-57280BC9AE8876C...
drwx------  9 test0766 nxusers 4096 2007-08-13 15:30 temp
```

The client directory *config* contains all session configuration files of the user. The files are stored in XML format and specify the host and port where connection takes place, as well as the other settings selected by user at the time of session setup. These XML files can also contain the password that will be used to connect to the remote server. NX Client uses a simple scrambling algorithm to make this password illegible. However, storing session passwords on clients is an inherently insecure operation and users are discouraged from doing so.

When a session is running, you can find directories whose names have as format: *C-test00-1183-A76C562ECFA489AB35E5E305D527EE2F* or *S-test00-1195-57280BC9AE8876CAA773467719FC1027*. The first letter specifies if it is a NX server or NX client session. Contrarily to what normal people could guess, *C* stands for server and *S* for client. The reason being that this name is similar to the X (and TCP/IP) convention where Mozilla, the client running on NX (read application) server, is connected to X server, running on NX client. What follows is the name of the NX server running the session, the display number and a MD5 constituting the unique session ID.

The session directory contains all information needed by agents and libraries to run the session, as for example the X authorization cookie needed by agent to connect to the X server. In case of failures, you should check the *session* and *errors* files you find in the failed session's directory. Failed sessions' directory have name prepended by *F*. *session* contains log output of agent and NX libraries, *clients* file contains the output of X clients run by session. Logging of X clients output can be disabled in the NX Node configuration file. *errors* is libXcomp specific and can contain some warnings. Warnings are always harmless. They are often used by NX developers to have suggestions on possible optimizations.

Session directories are automatically removed by NX server (at the end of session) and the client (at next run), unless they have been instructed not to do so.

The last mention goes to cache directories. They are maintained by libXcomp library to speed up session startup and compress the X traffic better, based on experience accumulated in previous runs. Caches depend on the session type, so multiple cache directories can be present. You can safely remove cache directories when you need space on your hard disk. This, of course, will cause sessions to run slower until the new cache is populated.