# Connecting X Terminals to Linux Mini−HOWTO


## Salvador J. Peralta

<speralta@willamette dot edu>



**Revision History**

| | | |
|---|---|---|
| Revision 1.2 release | 2003−05−20 | Revised by: sjp |
| Added more information on XFS and XDM configuration. | | |
| Revision 1.0 release | 2002−12−21 | Revised by: sjp |
| First official release. Removed old revision logs. | | |



This Mini−HOWTO describes how to connect X Terminals with a Linux host using nfs, xfs, xdm and xdmcp. Additionally, the document discusses configuration recommendations to optimize Linux for use as a host system on a network of thin clients. The most current version can be obtained here.

# Table of Contents

**1. Copyright**......1

**2. Introduction**......2
- 2.1. What is XFree86......2
- 2.2. What is an X Terminal?......2
- 2.3. Infrastructure Overview......2

**3. Host Environment**......3
- 3.1. Hardware......3
- 3.2. Software......3

**4. Basic Host Configuration**......4
- 4.1. Configuring NFS......4
- 4.2. Configuring XDM......4
- 4.3. Configuring Xaccess......5
- 4.4. Configuring Host Access......5
- 4.5. Configuring the X Font Server......5
- 4.6. Configuring the X Session......6

**5. Advanced Host Configuration**......7
- 5.1. Setting User and Group Limits......7
- 5.2. More Limits......7

**6. Client Configuration**......9
- 6.1. The Boot Process......9

**7. Additional Resources**......10
- 7.1. Linux Howto's......10
- 7.2. Man Pages......10
- 7.3. Linux Guides......10
- 7.4. Books......10

**8. Errata**......11
- 8.1. System Versions......11
- 8.2. Disclaimer......11
- 8.3. Feedback......11
- 8.4. Credits......11

# 1. Copyright

# 2. Introduction

This section provides a brief introduction to the X Window System and the client server model for X terminals and their Linux−based host systems.

## 2.1. What is XFree86

XFree86 is an Open Source implementation of the X Windows System (X). It provides a client−server interface between mouse, keyboard, video display, and a windowed desktop environment. Because X is network transparent, client programs can be run on a different host machine than the one on which an X Server resides.

The XFree86 bundle includes a large array of programs including a terminal emulator, clocks, and a mail interface. This document will focus exclusively on the portions of XFree86 required for maintaining an X Session between an X Terminal and a Linux host. These are the X Display Manager, xdm, an access control programs and the X Font Server, xfs.

## 2.2. What is an X Terminal?

X Terminals are diskless networked microcomputers. These systems will generally consist of mouse, monitor, keyboard, RAM, battery−powered NVRAM, and secondary ports for connecting peripheral devices such as printer, bar code scanner, etc. Because they are diskless, X Terminals load their operating system, configuration information, and fonts into memory from a host machine running nfs or tftp, and xfs.

The Operating System of an X Terminal will provide basic administrative interface for configuration parameters such as tcp, xdmcp, font path, boot path, and other host−related configurations. It's primary function, however, is to act as an X Server which exchanges mouse, keyboard, and other display information with client applications loaded on a remote host.

## 2.3. Infrastructure Overview

The canonical way of deploying X Terminal systems on an IP−based network is to use the Bootstrap protocol, bootp, to provide client and host ip addresses and to request a bootable image to mount the operating system. Clients may also store this information from their own NVRAM which mitigates the need for fetching it off of a network. NFS or TFTP is utilized to load the os image and system fonts into RAM on the client machine. XDM provides ip−based authentication and management of the display session using XDMCP. XFS may optionally be run on the host to provide anti−aliasing fonts to the client system (highly recommended). TCP is commonly the transport layer for all client−server interaction.

# 3. Host Environment

In an X environment, a host is a system that provides resources for an X Server. In a diskless workstation environment, the services provided by a host system will generally include session authentication using xauth and xhost, session management using xdm, file transfer services such as nfs and tftp which provide system fonts and a bootable image to be loaded into the X Terminal's RAM, and windowed client applications (e.g., Mozilla, Open Office).

## 3.1. Hardware

Our environment consists of 25 diskless workstations with 15 workstations using Mozilla, 6 worsktations utilizing a proprietary graphical application written in Java, and 4 workstations with an application running under a terminal emulator.The 6 java−based workstations use Gnome as their desktop environment. The remaining 19 workstations use FVWM2. To support that environment, we use 2 dual processor pentium 3's. One system has 1 gig of ram, the other has 500 megs. We have found that the stations running Gnome and Java use dramatically more memory and cpu than the stations running Mozilla.

## 3.2. Software

In order to use Linux as a host for a network of thin client workstations, you will want to get the latest release of the the XFree86 X Windows System, FVWM2, and nfs−utils. Assuming that you are using an RPM−based distribution, you will want the following packages: Xfree86−font−utils, Xfree86−libs, Xfree86−devel, Xfree86, Xfree86−truetype−font, Xfree86−xdm, Xfree86−base−fonts, nfs−utils, fvwm2, fvwm2−icons. To verify installed packages on an rpm−based distribution such as Redhat, you can use "rpm−qa | grep (packagename)". To install or update those packages simply download them from their maintainer or from your installation media with and "rpm −Uvh (packagename)". Alternately, you can use a graphical front end to rpm such as gnoRPM to maintain the required packages are installed on your system. For debian−based systems, or for Gentoo, please consult your system documentation on apt−get and portage, respectively.

# 4. Basic Host Configuration

Assuming that the required software is installed, we need to configure NFS to mount the fonts and operating system for our thin client workstations, configure XFree86 to provide remote display access, configure our Xsession to deliver our choice of window manager to our client systems.

## 4.1. Configuring NFS

In order to load the configuration information, base system fonts, and os image from the host machine using NFS, you need to provide read access to the directory or directories where that data is located. To do this, you will need to edit the `/etc/exports` file.

`/etc/exports` is an access control list which provides an entry for any directory that can be exported to an NFS client using the exportfs command. Configuring `/etc/exports` is easy.

To provide read access to any system which requests resources in `/home/ncbridge`, you would enter the following line into `/etc/exports`:

```
/home/ncbridge *(ro)
```

For better security, you can also specify access to individual client systems. To do this, simply enter:

```
/home/ncbridge client1.yourdomain.com (ro)
/home/ncbridge client2.yourdomain.com (ro)
/home/ncbridge client3.yourdomain.com (ro)
```

`/etc/exports` also provides facilities such as globbing for grouping like named groups together. For example, to provide read–only access to similarly named clients:

```
/home/ncbridge client*.yourdomain.com (ro)
```

## 4.2. Configuring XDM

The X Display manager is a program which authenticates, authorizes and manages sessions for an X server. The standard configuration from the X Consortium is appropriate for use with X Terminals.. However, this file needs to be configured to not use display :0 as the default console, and XDM should be configured to listen for XDMCP requests.

In `/etc/X11/xdm/xdm-config` make the following configuration changes – Comment out the lines which would set display :0 as the console:

```
! The following three resources set up display :0 as the console.
!DisplayManager._0.setup:       /etc/X11/xdm/Xsetup_0
!DisplayManager._0.startup:     /etc/X11/xdm/GiveConsole
!DisplayManager._0.reset:       /etc/X11/xdm/TakeConsole
```

Comment out the line which sets the requestPort for the display manager to :0.

```
!DisplayManager.requestPort:    0
```

## 4.3. Configuring Xaccess

Xaccess is a file which XDM reads to determine which X Terminals can login and access software running on the host machine. An X Terminal sends a request to XDM which in turn runs an Xsetup script which sets up the login screen and displays the xlogin widget which handles authentication and on successful authentication sets up a session between client applications on a host machine and an X Terminal.

The only configuration changes typically needed here involve setting up the access control to restrict global access to the xserver, and to add individual authorized hosts by ip address. Assuming that you are allowing multiple clients to log in, your configuration should look something like this:

```
#*                  # Allow any X server can get a login window
158.100.5.121       # Only allow specific systems to login.
158.100.5.122
158.100.5.123
158.100.5.124
158.100.5.125
!*  # Prevent all other systems from using XDM
```

## 4.4. Configuring Host Access

If you are not running terminals from a DNS server and/or are not allocating IP addresses using DHCP (as we are not), you will need to add an entry for each X Terminal in /etc/hosts on the Linux host machine.

```
158.100.5.121         # One system per line with optional
158.100.5.122         # parameters for lprhost and loghost
158.100.5.123
158.100.5.124
158.100.5.125
```

## 4.5. Configuring the X Font Server

The X Font Server (xfs) provides system fonts to X display servers. It is the only way to provide anti−aliased fonts such as freetype to most X Terminals.

On most Linux systems, xfs is started from a script in `/etc/init.d`, and/or on startup from one or more scripts, `/etc rc.d/init.d` or in one of the run−level initialization scripts in /etc/rc[0−6].d. The most important thing to verify is that the X Terminal and the X Host are broadcasting and listening on the same port. Some X Terminals use the default xfs port, 7100, for other services, so xfs can be run on an unassigned port, such as 7102, in order to avoid a conflict.

On most systems, the configuration for XFS is in `/etc/X11/fs/config`. In order to enable XFS to be accessed remotely, comment out the no−listen = tcp directive from the configuration file. Be sure to set a max client limit equal to or greater than the total number of clients you will be serving.

## 4.6. Configuring the X Session

The Xsession file controls much of the information regarding a particular host machines X environment. Examples of things that might be set in this file includes logging, which desktop manager to run, and which keyboard mappings to use. These settings are highly environment specific, and for that reason will not be addressed in this document. Suffice to say that you will want to decide whether to use user−level or global preferences for the desktop environment and that those configuration changes must be made in this file and in a .xsession file in the user's home directory.

---

# 5. Advanced Host Configuration

The section addresses some optional configuration options that may improve performance on a Linux X Host.

## 5.1. Setting User and Group Limits

One of the dangers associated with the thin client model is that a runaway process might eat up all of the available system memory and/or cpu on the host system. When this happens, the performance on that system can degrade resulting in system hangs, freezes, and a host of other generally undesirable consequences.

Fortunately, Linux comes with a set of facilities to mitigate these kinds of problems. The first tool that we will look at is the limits.conf file located in /etc/security. limits.conf provides the ability to specify user and group level limits to certain types of system resources, including memory. Limits set in this file are set on a per user or per group basis. Though there are many limits that can be set in this file, we will address only a few in this tutorial.

The basic syntax for limits.conf consists of individual lines with values of the following types: (domain) (type) (item) (value) where domain is user or group, type refers to a hard or soft limit, item refers to the resource being limited and value referring to the value associated with the limit being set. For example, setting the following value:

```
guest           hard     priority        19
```

places a hard limit on the priority with which jobs are scheduled for a user named 'guest'. In this case, guest is always scheduled at the lowest possible priority.

As another example, to set a max cpu time of 10 minutes for user guest, set the following value:

```
guest           hard     cpu             10
```

As a final example, to set a limit of 3 failed logins for a group called guest, set the following value:

```
@guest  hard    maxlogins       3
```

## 5.2. More Limits

A second method for limiting the potential impact of runaway proceses is to set limits on a per process basis. This can be achieved by setting the ulimit command in `/etc/profile`.

To set a soft limit on the maximum amount of memory available to a given process to a value that is less than the total amount of memory on the system on a system with 1 gig of real memory and 500 megs of virtual memory you would set the following values in `/etc/profile`:

```
ulimit -S -m 1000000
ulimit -S -v  500000
```

With this value set, the system will kill any process that tries to take up more resources than you have set as a limit.

# 6. Client Configuration

This section attempts to discuss, in generic terms, the configurations needed to connect a generic X Terminal to a host system.

---

## 6.1. The Boot Process

When an X Terminal is booted from a host on the network, it may either use bootp or rarp to determine ip and host addresses, or it may store those values locally in NVRAM on the terminal. The advantage to using bootp is that the host information and ip addresses may be stored centrally in a single configuration. An advantage to using the values locally on the client system is that it allows for a more fine−grained control over tcp/ip and host information.

Assuming that host and ip address information is properly configured for the X Terminal, step two in the boot process is for the X Terminal to load its operating system and core fonts. In order to do this, the client needs to know the full path to the OS image and system configurations and base fonts. It needs to have a boot method specified for fetching these files. On most systems, that is tftp or nfs. Last, the X Terminal needs to know which will be managing the XDMCP communication ( i.e., the system from which client applications will be deployed ). All of this information is stored in NVRAM on the X Terminal.

Step three in the boot process occurs when the X Terminal connects to the client machine. Most X Terminals can send a direct or indirect query to an xdmcp session manager on a network. In an indirect query, all available xdmcp session managers will respond and display a menu of possible hosts. In the case of a direct query, xdm on the host system runs the Xsetup script which sets up the display for the login screen and provides the xlogin widget which requires a valid userid and password to login and start an xdmcp session.

---

# 7. Additional Resources

## 7.1. Linux Howto's

This document is intended as a replacement and extension of the unmaintained X Terminal HOWTO. The following Mini−HOWTO's can be found at the Linux Documentation Project (TLDP) at www.tldp.org: XWindow−Overview−HOWTO, XWindow−User−HOWTO, XDM−Xterm Mini−HOWTO, NCD X terminal Mini−HOWTO.

## 7.2. Man Pages

TLDP, the XFree86 project, and the GNU Organization have a series of online manuals which describe many of the services dealt with in this Mini−HOWTO. Man pages relevent to this document include pages for xdm, nfs, exports, X, and ulimit.

## 7.3. Linux Guides

Chapter 14 of the LDP "Network Administrator's Guide" (NAG) has useful information on using NFS in Linux.

## 7.4. Books

The "X Windows System Administrator's Guide" by O'Reilly and Associates is highly recommended.

# 8. Errata

## 8.1. System Versions

The methods outlined in this Mini−HOWTO have been used in a production environment utilizing Redhat 7.3, 7.1, 6.3, and 6.1 with XFree86 versions beginning with 3.2 through 4.2. The X Terminals used in this environment are Network Computing Devices (NCD) Network Computer versions nc900, nc500, and nc350.

## 8.2. Disclaimer

This document is provided "as is". Neither the author nor the distributors of the document are responsible for any physical, financial, or other form of damage incurred by the suggestions provided in this text.

## 8.3. Feedback

If you find any portion of this document useful, or if you have suggestions, please email the maintainer, speralta [at] willamette [dot] edu.

## 8.4. Credits

My sincere thanks goes to the Willamette University for employing me to build and maintain Linux−based systems like the ones described in this document, and to Michael Spalti for introducing me to this technology and for his development and support of these systems.