Linux XDMCP HOWTO

Thomas Chao

<tomchao@alcatel-lucent.com>

Revision History

Revision v1.4	11 June 2007
Adding info for now popular Ubuntu 7.0.4, RH Commercial Workstation v.3, Fedora Core 6 and 7, new Mandriva.	
Revision v1.3	2 January 2003
Adding info for Red Hat 7.3 & 8.0, Mandrake 8.2 & 9.0, SuSE Linux configuration and contents update.	
Revision v1.2	15 March 2002
Adding more info for Red Hat 7.2, Mandrake 8.1 and Slackware 8.0 Linux configuration and SSH X11 Forwarding.	
Revision v1.1	20 March 2001
Revision and adding RH 7.0.	
Revision v1.0	01 November 2000
Initial revision and release.	

This HOWTO describes how you can use the combination of X Display Manager (xdm, kdm and gdm) and XDMCP (X Display Manager Control Protocol) to provide a solution for the X−Terminal and to provide a platform of efficient Remote X Apps environment. This document will takes the focus on how to setup the X connection using XDMCP.

Table of Contents

1. Introduction....1
1.1. Disclaimer....1
1.2. Feedback....1

2. The Procedure....2
2.1. Before you begin, some background....2
2.2. Security Reminder....2
2.3. The System I use....3
2.4. Remote Client Piece....3
2.5. Server Preparation....4
2.6. Steps to Complete the Procedures....5
2.7. Testing....8

3. X11 Forwarding using SSH....9

4. Troubleshooting....10

5. XDMCP and GDM (Gnome Display Manager)....12

6. Additional References....13

7. Authors....14

8. Copyright Information....15

1. Introduction

X Window System is the display and networking protocol developed by MIT. The X is built with network in mind with the capability to run a (graphical) session on a remote computer. In it, an X Display Manager is used to start a session from a local system or from another computer. The request and the start of the session is handled by the XDMCP, which stands for "X Display Manager Control Protocol" and is a network protocol. It provides a way of running the X−Terminal to run on your PC (or MAC) and it uses the X Server to provide a client/server interface between display hardware (the mouse, keyboard, and video displays) and the desktop environment while also providing both the windowing infrastructure and a standardized application interface (quoted from XFree86 Project home page). The X−Terminal can be displayed with an individual window or multiple windows, based on your X window system's software capabilities and setup.

I am always looking for the best way to use Linux, both at home and in work. One of the biggest advantages among all is the ability to re−use the older systems (like Pentium, Pentium II, Pentium III and even the 486 and AMD x86 CPUs) as a Xterminal (by using the Win32 apps; like Hummingbird's Exceed, Reflection X, X−Win32 or X−ThinPro. For MAC, try eXodus) to run from any of your PC remotely. I found out, somehow very surprising, that there are many documents on the INTERNET that can help you to set it up, but not with a step by step HOW−TO format! This is how I came up with this document as a way to share my experiences with all users. By using X and XDMCP, you can build a good, reliable and low cost X− environment for your home or work IT solution. Best of all, it is free! You will also find out that those long abandoned PC by the current Windows system can run the X in Linux just fine! It can save you money and spare the mother earth!

In recent years, new Linux distributions are getting easier to use by adding new user UI. However, I still believe much in manual control of the system and application if I can, because in this way, I know what I am going to change in my system in the way I wanted it. Therefore, I will focus this document on manual configuration.

1.1. Disclaimer

No liability for the contents of this documents can be accepted. Use the concepts, examples and other content at your own risk. As this is a new edition of this document, there may be errors and inaccuracies, that may of course be damaging to your system. Proceed with caution, and although this is highly unlikely, the author(s) do not take any responsibility for that.

All copyrights are held by their by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

Naming of particular products or brands should not be seen as endorsements.

You are strongly recommended to take a backup of your system before major installation and backups at regular intervals.

1.2. Feedback

Feedback is most certainly welcome for this document. Without your submissions and input, this document wouldn't exist. Please send your additions, comments and criticisms to the following email address : <tomchao@alcatel-lucent.com>.

2. The Procedure

This section details the procedure for setting up Xterminal using XDMCP. The pre−requisite is to have a (any) Linux distribution installed and running X.

2.1. Before you begin, some background

Before you begin, it is better to have a basic understanding of how this works. The X server is usually started from the X Display Manager (DM). In this X DM Wiki page, it gives you a basic understanding of how it works! (More details are at the Resources below and LDP HOWTO page)

Almost all the Linux distributions include the xdm, kdm and gdm to you as your choices. (This document will use gdm and kdm as an example). The Display Manager provides a nice and consistent interfaces for general users (X−based login, starting up a window manager, clock, etc.). X Display Manager manages a collection of X displays, which may be on the local host or remote servers. It is worth noting that the **Xsession** file is what runs your environment.

When xdm runs, it offers display management in two different ways. It can manage X Server running on the local machine and specified in "Xservers", and/or it can manage remote X Servers (typically Xterminals) using XDMCP as specified in the "Xaccess" file. (refer to the xdm man page).

For kdm (which comes with the KDE desktop), it is a replacement of xdm and configures the same way, except its files are in `/etc/X11/kdm` in Caldera/SCO, `/etc/kde/kdm` in Red Hat (and Fedora Core) and `/usr/share/config/kdm`, which is a symbolic link to `/etc/kde/kdm`, in Mandrake.

The gdm (Gnome Display Manager) is a re−implementation of the well known xdm. gdm has similar functions to xdm and kdm, gdm is the Gnome Display Manager, and its configuration files are found in `/etc/X11/gdm/gdm.conf`. The `gdm.conf` file contains sets of variables and many options for gdm, and the Sessions directory contains a script for each session option; each script calls `/etc/X11/xdm/Xsession` with the appropriate option. gdm has similar functions to xdm and kdm, but was written from scratch and does not contain any original XDM / X Consortium code.

RH 8.0 introduces the new graphical interface called "Bluecurve". The new interface is aimed for XP feel and styles. The setup makes no difference in this case!

Other good references for the similar setup can be found in the following documents:

- The XDM and Xterminal mini−HOWTO, by Kevin Taylor
- Linux Remote X Apps mini HOWTO A very good reference for Remote X in both theoretical and practical view. By Vincent Zweije
- The Connecting Xterminal mini−HOWTO, by Salvador J. Peralta
- The Using and Managing GDM [PDF] from The GNOME Project.

2.2. Security Reminder

Do not believe the myth that Linux (or UNIX) is a safer OS than the MS Windows! All OSs are vulnerable to the hackers, if the user does poor configuration job or maintaining the security updates!

You need to bare this in mind that both X and XDMCP is inherently insecure, and that's why many of the distributions shipped as it's XDMCP default turned off. If you must use XDMCP, be sure to use it only in a trusted networks, such as corporate network within a firewall. Never use it in the open network (or Internet) environment without a firewall protection! If you are using at home, remember to add a firewall equipped router for protection.

A good way to test your network security is to test it using the ShieldsUp by Gibson Research. It is free and easy to use!

XDMCP connection opens up UDP ports; therefore, it is not natively able to use it with SSH. Currently, SSH1 and SSH2 are not implemented to securely forward the UDP communication. To secure the connection with SSH, the technique is called X11 TCP/IP Port Forwarding. Check this Why Port Forwarding? site and the Resources area for additional HOW−TO information. If you would like to experiment this, I have add a little section below to show you how it works. I will give you only the basic idea how it works, and I will leave the more advanced way of running it to other experts and/or HOWTOs.

2.3. The System I use

I have tested the setup running a GNOME (gdm), as well as KDE (kdm) on the following distributions:

- Red Hat: From RH 8.0 down to 6.0. RH Workstation v.3 (commercial).
- Fedora Core v.5 to v.7. (The new RH free version)
- Mandrake Linux from 7.2 to 10.0 and Limited Edition 2005. I would also like to test it out on the new Mandriva 2007 Spring version.
- Ubuntu version 6.x, 7.04.

SuSE 7.2 (SuSE is now the new Novell Linux) and Slackware 8.0's setup are tested by the users, thanks to Peter Van Eerten and others, who helps the test for this HOW−TO. (I would like to thank all users who help me on this project). The other I have tried on is Caldera eDesktop 2.4 (now owned by SCO), which is similar to RH's setup, except that it uses KDE. I have not had a chance to test it on other Linux flavors like Debian, Turbolinux, Gentoo, etc. However, the setup should be similar and should work just fine. If you have successfully setup one other than the distribution listed above, please share it with me. I will add them into this document.

The PC hardware that I am using is an IBM PC clone running an Intel Celeron 2.9 GHz with 1 GB memory and a 160 GB ATA−133 Hard Drive. The oldest system I current have (in 2007) for the testing are using the Intel Pentium II 450 MHz PC with 128 MB memory and it is running with good performance. (I test run on an old Pentium 100 MHz PC in 2003 and it runs OK). I use a built−in Fast Ethernet NIC in my Intel clone M/B. In my old machine, I use the 3Com 10/100 (3C509B) NIC with an ATAPI DVD−ROM and an IOMEGA ZIP drive. I have also test it on my IBM T21 laptop connecting using my Agere Wireless LAN card. I have also test the setup on one of my system at home that is using the AMD 64−bit CPU running the Fedora Core 6.

2.4. Remote Client Piece

I use the Hummingbird Exceed 10.0 (Exceed 6.x and 7.0 are also working fine) on my PC and have tested them on Windows NT 4.0, Windows 2000 Pro, Windows XP. I found out that other popular choices are X−Win32 and X−ThinPro, but I did not have a chance to test them out. There are also many open−source applications, as well as commercial one available, if you happen to have one.

2.5. Server Preparation

In RH 7.x and other newer dists, you would need to setup DNS lookup, in order for some networking function to work properly (such as **telnet** that we will use to test the setup). You can use "**netstat –r**" and/or "**arp –a**" command to verify your DNS setup or response time. If you are in a small environment (like home or small office) that do not have your own DNS and are relying on your ISP's DNS Server, then add the entry of your Linux workstation or server name(s) in the "`/etc/resolv.conf`" file. If you are only use it in the lab or at home, then, you can add the host name of all workstations in your local static hosts table in "`/etc/host`". You would need the root privileges to update the naming information.

To prepare your X Server for XDMCP session, you would need to make sure the following are properly installed:

1. Install your Linux OS. In my case, I use mostly Fedora Core 6 in my lab and Ubuntu 7.04 at home. If you plan to use SSH Port Forwarding, you need to install the OpenSSH package or compile SSH with your kernel. Also, most dists now come with firewall installed by default (unless you choose not to). You may encounter problem, if you do not add firewall rules or temporary disable it in setting up XDMCP. I will not cover the firewall rules here in details, since this is not the focus of this document. I will share with you only on how to make it works first and you can fine–tune it yourself.

   To show your firewall rules, in kernel 2.2x, use the command **ipchains –L** to list your default rule sets. To temporary disable it, use this command **ipchains –F** to flush the rules (Don't worry, it will restore by re–loading or re–boot). For kernel 2.4x and up, replace the command **ipchains** with **iptables**. To start with it, you can try to edit this **/etc/sysconfig/ipchains** file and commented out this rule (this is a feedback from a user. You can test it by yourself):

   ```
   -A input -p upd -s 0/0 -d 0/0 0:1023 -j REJECT
   ```

   and insert these two rules to allow packets pass through port 177:

   ```
   -A input -p udp -s 0/0 -d 0/0 0:176 -j REJECT
   -A input -p udp -s 0/0 -d 0/0 178:1023 -j REJECT
   ```

   (Note: XDMCP uses TCP, UDP port 177 and TCP port 6000 to 6005. xfs server is using port 7100 in our setup).

   You should be able to use the **iptables** in the similar way. (Check for iptables references at the Resources area or this setup example).

   For more firewall details, check the IP Masquerade HOWTO page.

   One other easy way is to add rules that only accept certain IP address(es) from your trusted workstations. Please feel free to experiment it by using the **iptables** command. Again, I will not cover the details here. I am the lucky one, because I have my company's firewall to protect me from the outside world.

   If you would like to use the GUI tool to configure the firewall using iptables, try this good one: the Firestarter.
2. Setup your Networking. To test it out, you can use the **ping**, **ftp** and **telnet** command to determine if your are networking. RH 7.x and up do not have **telnet** daemon turn on by default (for security reason). Remember to enable it, if you prefer to use it for your test. You can always turn it off when you are done (Using **ntsysv** in RH, or **rcconf**, **sysvconfig** in Ubuntu and Debian, with root privilege).

One other thing is to remember firewall rules are there. Add your own rules or temporary disable it (as mentioned above) to make these commands work.

3. Setup X. Do *not* setup with a resolution higher than what the remote users are able to use for their display. The newer version is now capable of probing the video chipset and determine that for you. Some older (X) version may not! Test the X Server by typing either **startx** or **telinit 5**. Make sure X is running properly.
4. Creates the necessary user account(s) (and associated group) for user who will access via the Xterminal.

2.6. Steps to Complete the Procedures

Although X can use the local fonts, it is better to use the xfs font server in an networking environment. If this is what you want in Linux X environment, you need to provide font using either X font server (xfs) or hard coded font path in XF86Config and XF86Config–4 configuration files. If you plan to use xfs font server (check here to see the xfs advantages). xfs server can also offload the burden from your local workstations. If you plan to use local fonts, you can skip step 1.

These are the steps I used to setup the X Server for accepting XDMCP requests:

1. In earlier version of RH and Mandrake, modify `/etc/rc.d/init.d/xfs` and make the following changes. Change all lines(this is where the Font Server port), if the port is not set to 7100.

```
daemon xfs -droppriv -daemon -port -1
```

to:

```
daemon xfs -droppriv -daemon -port 7100
```

In some new distributions, it is by default, for security enhancement, not listening to TCP port any longer! If you would like to setup X font server, you need to do the following steps:

Change this line in `/etc/rc.d/init.d/xfs (or in /etc/init.d/xfs for some dists):`

```
daemon xfs -droppriv -daemon
```

to:

```
daemon xfs -droppriv -daemon -port 7100
```

In Ubuntu 7.04 Desktop version, you need to download and install the xfs package. then modify `/etc/init.d/xfs` and change the following line:

```
start-stop-daemon --start --quiet $SSD_START_ARGS -- -daemon \
```

to:

```
start-stop-daemon --start --quiet $SSD_START_ARGS -- -droppriv -daemon -port 7100 \
```

Then, in `/etc/X11/fs/config`, comment out this line:

```
# don't listen to TCP ports by default for security reasons
#no-listen = tcp
```

If you change or add the port, use this command to restart your X font server (requires root):

```
service xfs restart
```

You do not have to use port 7100. You can set a different port, as long as you carefully plan it first to make sure no conflicts in using the port number and change it accordingly. It is better to consult your Linux admin before doing so, so that he/she knows the port has been taken! Different Linux distribution may put the xfs in different folder under /etc/rc.d. You may search for it if that's the case.

2. If you plan to use the XDM, modify `/etc/X11/xdm/xdm-config` and make the following change. Be default (in most Linux distributions), this line is set, so that it is not listening to XDMCP connection. This is for security reason. For Caldera and other dists that uses kdm, this file is at `/etc/X11/kdm`. Find this line:

```
DisplayManager.requestPort:     0
```

and comment it out as:

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

Remember, this does not affects gdm. For gdm setup, it is in the following section.

3. In `/etc/X11/xdm/Xaccess`, change this. (this allow all hosts to connect). For Caldera using kdm, this file is at `/etc/X11/kdm`. Set the security to 644 (chmod 644):

```
#*     # any host can get a login window
```

to:

```
*      # any host can get a login window
```

The above setup is in a Broadcast mode, which will list all the X Server that are listening and willing to manage your X connection. If you only want to allow certain connections, use the **CHOOSER** section in this same file. An example can be found in the Resources.

4. If you plan to use the GDM as default, one benefit of gdm login window is that it allows you to switch between KDE and GNOME. For gdm, edit `/etc/X11/gdm/gdm.conf`. This activates XDMCP, causing it to listen to the request. For kdm (if you pick KDE as your DM in your installation), edit `/usr/share/config/kdm/kdmrc` for Mandrake and `/etc/kde/kdm/kdmrc` for Red Hat or `/opt/kde2/share/config/kdm/kdmrc` for Slackware version (KDE2). Change this line:

```
[xdmcp]
Enable=false (may shown as 0 in some distributions)
```

to:

```
Enable=true (or 1 in some distributions)
```

Make sure "**Port=177**" is at the end of this block, i.e., by commenting out the line "#Port=177".

(As a side note for Ubuntu user who care only about ease of use, this is what you can do (just turn on XDMCP w/o xfs). From "System" menu, go to "Administration" and the "Login Window" Alternatively, you can use "sudo gdmsetup" command). Click the "Remote" tab and in "Style", select "Same as Local". Then click the bottom "Configure XDMCP" button to verify the setup. If you choose "Remote login disabled" in style, it will disable the XDMCP. Additional setup is in the "Security" tab and the lower "Configure X Server..." button and select "Chooser" in Server. You must restart gdm to enable it! Doing this is quick and simple, but you lose the sense of what files are being touched and changed! Easy of use or controllability is your choice here!)

5. (For Ubuntu and new Debian see notes below) Now edit `/etc/inittab` and change the following line. The digit here meaning the default runlevel. For X, the runlevel should be "5".

```
id:3:initdefault:
```

to:

```
id:5:initdefault:
```

In Slackware, the X11 mode is number "4", not "5". Refer to this runlevel wiki page for different dists' definition.

This is switching from Text Mode login to Graphical Mode using Display Manager. Before changing this line, you can use the **telinit** command to test prior to modifying the line. Use either **telinit 3** to set to level 3, or **telinit 5** to set to level 5, graphics mode (you can issue this command on the second machine that telnets into this server).

Runlevel 2–5 is the same in Debian and Ubuntu. Since Ubuntu 6.10 (and future Debian), the way to start the runlevel were changed from the init daemon to the Upstart, with which the tasks and services are managed by events. Each runlevel is defined by the files in the system in the format of `/etc/rcx.d`, where the "x" represent. Each event is trigger (or changed) by issuing the **telinit 3** command.

6. Make sure the proper security of the file `/etc/X11/xdm/Xservers` is set to 444 (chmod 444).
7. Locate `/etc/X11/xdm/Xsetup_0` and **chmod 755** this file.
8. Edit the `xorg.conf` file in the `/etc/X11` folder and change the line (for older version, it is either `XF86Config` or the `XF86Config-4` file for XFree86 4.x):

```
FontPath    "unix/:-1"
```

to:

```
FontPath    "unix/:7100"
```

If you decide to use the port number other than the usual 7100, be sure to change both in "/etc/rc.d/init.d/xfs" (or in "/etc/init.d/xfs") file and here!

To save your time and energy, I recommend you to add the FontPath in the xorg.conf (or XF86Config and/or XF86Config–4) configuration files. If you are not sure what fonts are available to you, you can use this command to check it out (requires root):

```
chkfontpath --list
```

The following are some of the example fonts for your reference. Make sure you have these fonts before editing these path.

```
        FontPath  "/usr/X11R6/lib/X11/fonts/75dpi/"
        FontPath  "/usr/X11R6/lib/X11/fonts/misc/"
        FontPath  "/usr/X11R6/lib/X11/fonts/CID/"
        FontPath  "/usr/X11R6/lib/X11/fonts/Speedo/"
        FontPath  "/usr/X11R6/lib/X11/fonts/100dpi/"
        FontPath  "/usr/X11R6/lib/X11/fonts/Type1/"
```

If you don't have the chkfontpath command and you are using the local fonts, you can simply edit the file "/etc/X11/fs/config". Find the line that starts with "catalog=", and add your directory at the end of the list, separated by a comma. An example are like this:

```
    catalogue = /usr/X11R6/lib/X11/fonts/misc:unscaled,
                /usr/X11R6/lib/X11/fonts/100dpi:unscaled,
                /usr/X11R6/lib/X11/fonts/100dpi,
                /usr/X11R6/lib/X11/fonts/75dpi
```

9. (You **do not** have to make this change. You can keep the default setting, but this is what I prefer. If you are not sure, leave this alone.) Change this line to the end of `/etc/inittab`:

```
x:5:respawn:/usr/bin/gdm
```

If you decided not to change this line, it is fine! This is not a required step, but of a personal preference! There is no need to do this in Ubuntu and newer Debian dist.

You are now ready to run a test.

One other thing to know (that some users have asked) is how to display with **Willing to manage** message with load info As I know this is available in xdm by adding the following to the `/etc/X11/xdm/xdm-config`.

```
DisplayManager.willing:  su noboby -c /etc/X11/xdm/Xwilling
```

and the XWilling script must exist. For gdm, add this line to the `/etc/X11/gdm/gdm.conf` in `[security]` section:

```
Willing=/etc/X11/gdm/Xwilling
```

A sample of Xwilling script is here for your reference. Adding this script or not is your preference. It is not required step here!

2.7. Testing

To test if your XDMCP with X Server is ready to accept connection(s), do these steps. I find it easier using the X Server and another machine to test it:

1. (Re−)Start your X (which is in runlevel 5 or runlevel 2 in Ubuntu). If you are not sure how to do this, simply reboot your system (but this is really not necessary, if you know how to restart it using command line. That's the beauty of Linux, when comparing it to MS Windows).
2. If you have not modify your firewall rules, you need to temporary disable it by using **iptables −F** (or **ipchains −F**).
3. Make sure the graphical login page comes up. Make sure the display resolution and mouse work. Log in from the console to see if the local access is OK. If OK, do not log off.
4. Setup Hummingbird Exceed (or other X Client software) to either query this machine (using the IP address or fully qualified DNS name) or set to use XDMCP−Broadcast and try to connect to the X Server. You should see the X Session come up and the login screen appear.

3. X11 Forwarding using SSH

As I have explained earlier, using XDMCP to display X across Internet is basically a no−no, due to it's lack of encryption across the Internet. One way to enforce the traffic security is to use the SSH by the way of X11 tunnelling or port forwarding. SSH (Secure Shell) is developed in 1995 by Tatu Ylonen to replace the insecure telnet, ftp, scp, rcp, rlogin, rsh, etc. The first thing you need to know is that X11 forwarding using SSH is different from your regular, non−secure way of running X Window.

To start this setup, you need an additional piece of information. First, you must have your SSH package installed. In Linux, they are the OpenSSH packages. Check your distribution to decide what package you need to install (some installed it as standard packages). Secondly, you need a Windows SSH Client (other OS version, like MAC, are also available). I recommend PuTTY. It is a wonderful free SSH client and you can download them from this link. Remember to download the document and read them carefully. The other good free SSH clients are: Tera Term Pro + TTSSH: An SSH Extension to Tera Term, SSH Secure Shell Client by SSH.com (only free for non−commercial use). I will break down again into steps, so it is easy for you to follow.

1. Open up the command **putty.exe** by double−click it. It will brings up the interface. First, setup the connection info in Host Name (or use IP) field and select SSH (SSH is using port 22). In Connection Category, find the Connection tree. In SSH, expand it and you will see Tunnels window. Click "Enable X11 forwarding". It is setting the default to X display at "localhost:0". Now, go back to Session and save this session with a name you like. I normally use the Host Name to make me easily remember where I am connecting to.
2. In the example of Hummingbird Exceed, this is what you need to do. (For other X client, the setup is similar). Open up the Xconfig from your Exceed folder. In your "Screen Definition", change to "Multiple" Window mode and save it. Next, open up your "Communication" icon and set the Startup mode to "Passive".
3. Now you are done. To test it, first using PuTTY (or other SSH client) to connect to your server. The first time connection, it will ask you whether you want to cache the Security Key or not. (Yes is normal choice). Once log in is done, fire up your Exceed. It will stay in the background. Now you can execute any of your X application and it should forward the X application via SSH to your local screen. For example:

   ```
   $ xclock &
   ```

   We should now see the Xclock is running on your local screen.

Now you see the difference is that you do not see all your X Window. You are simply running X application one by one and forwarding via SSH to your local screen. Therefore, you need to know the command for running each X application. All the control are done via SSH client window. To me, the security is worthy than the slightly inconvenience!

Hummingbird Exceed's newer version now support the SSH connection. I am sure other X application may be able to do the same in their latest new version. Check the application web site you are using or the Resources belowfor for more details).

If you are using X−Win32 and you want to use SSH with Port Forwarding, you can use this reference to set it up.

---

4. Troubleshooting

- If X cannot come up and is broken:

  If X is broken and the connection fails, most of the time it has this error messages:

```
_ FontTransSocketUNIXConnect: Can't connect: errno = 111
failed to set dafault font path 'unix:-1'
Fatal server error:
could not open default font 'fixed'
```

  This is likely due to xfs not finding the correct port for the Font Server or the font path is not set correctly! To resolve this, check steps 1 and 8 above. Make sure the configuration are pointing to (port) 7100 and make sure you have the following fonts installed (if not re−install the XFree86 font packages from your CD). Check the listing in `XF86Config` file (if you are using XFree86 4.x, the file is `XF86Config-4` and `xorg.conf` in newer X11 version) at `/etc/X11`:

```
FontPath  "/usr/X11R6/lib/X11/fonts/75dpi/"
FontPath  "/usr/X11R6/lib/X11/fonts/misc/"
FontPath  "/usr/X11R6/lib/X11/fonts/CID/"
FontPath  "/usr/X11R6/lib/X11/fonts/Speedo/"
FontPath  "/usr/X11R6/lib/X11/fonts/100dpi/"
FontPath  "/usr/X11R6/lib/X11/fonts/Type1/"
```

  Use the command **startx** (on local) to restart the X server (or use **telinit 5** to switch the runlevel). To restart xfs, use the command in step 1.

  I found out in my RH 7.3 that if my xfs is not setup, it will crash the Exceed connection if I use the GNOME. (Using KDE is fine and it does not affect my Mandrake GNOME). After I fix it and start up my xfs, it works fine.
- If Exceed has no respond (in blank screen):

  In this case, most likely your xdm (or gdm, depending upon which is used in `/etc/inittab`) is not starting correctly. Issue the command: **ps –ef | grep gdm** (or xdm or kdm, replace it in the command). Also, if your box has udp port turned on for XDMCP, you can type **netstat –l | grep xdmcp** and you should see this:

```
udp    0    0  *:xdmcp          *:*
```

  If the process is not running, check the steps on the setup above (make sure there are no typo's and that the correct path is given). Restart X using the command **telinit 5**. If the udp port is not there for XDMCP, do step 2 as above.

  Another possibilities are that your DNS setup is incorrect and/or firewall is enabled. An easy way to find out is simply **ping** or **telnet** your host and if the reply takes a long time, then that's DNS problem. If by using **telnet** and you got a "`Connection Refused`", then this is a firewall problem (assuming that you have your telnet daemon turned on already)! Check the section above for details how to resolve this.
- PC Box with PPPoE (PPP over Ethernet):

  A user using PPPoE told me that if you have PPPoE, you might experience problem using XDMCP. After uninstall it, he then is able to get XDMCP working. I personally do not have the environment to test this, so you can test it yourself.

- Linux to Linux Display export:

  If you are using another Linux with X, you do not need to use XDMCP to manage your display. You can actually export your display right from your X box. To do this, you must enable your access control to allow other to make connection to the X Server. The common error you will get without doing so are:

```
xlib: Connection refused (error 111): unable to connect to X server xlib: No such proc
```

  To resolve the problem, use the command below:

```
$ xhost +
$ export DISPLAY=(your local host IP):0.0
```

  Always remember to enable access control by using the command "`xhost -`" again. One thing to remind you, you do not need this, if you are using PC as X−Terminal using XDMCP. This is only required when you have Linux to Linux or Linux to UNIX connection.

  If you are using many Linux X boxes and you would like to setup the Chooser to pick from which X to login, you need to enable the following in the `/etc/X11/gdm/gdm.conf`:

```
[daemon] Chooser=/usr/bin/gdmchooser --disable-sound --disable-crash-dialog
...
[xdmcp] Enable=1
HonorIndirect=1
```

- I got a "Signal 11" error:

  The "Signal 11" error, also called "Segmentation Fault", can sometimes be a problem of your hardware and/or software. If you have this problem in bring up the X Server, you need to fix it before configuring XDMCP. Unfortunately, there is no simple way to fix the problem due to many possible causes. For details, please check this SIG 11 while compiling the Kernel.

---

5. XDMCP and GDM (Gnome Display Manager)

The following is taken from the Gnome Display Manager Reference Manual:

GDM also supports the X Display Manager Protocol (XDMCP) for managing remote displays. GDM listens to UDP port 177 and will respond to QUERY and BROADCAST_QUERY requests by sending a WILLING packet to the originator. GDM can also be configured to honor INDIRECT queries and present a host chooser to the remote display. GDM will remember the user's choice and forward subsequent requests to the chosen manager. GDM only supports the MIT−MAGIC−COOKIE−1 authentication system. Little is gained from the other schemes, and no effort has been made to implement them so far. Since it is fairly easy to do denial of service attacks on the XDMCP service, GDM incorporates a few features to guard against attacks. Please read the XDMCP reference section below for more information.

Even though GDM tries to outsmart potential attackers, it is still advised that you block UDP port 177 on your firewall unless you really need it. GDM guards against DoS attacks, but the X protocol is still inherently insecure and should only be used in controlled environments. Even though your display is protected by cookies the XEvents and thus the keystrokes typed when entering passwords will still go over the wire in clear text. It is trivial to capture these. You should also be aware that cookies, if placed on an NFS mounted directory, are prone to eavesdropping too.

---

6. Additional References

Some additional references on this subject include:

- Your local xdm man page.
- Your local gdm man page.
- X Display Manager Wikipedia
- GDM and XDMCP Security
- Using and Managing GDM
- Configuring XDM (from Linux Journal)
- Configuring Chooser through X Resources
- XDMCP Documentation (Compressed PostScript file download)
- Should you be running XDMCP?
- Accessing Xterms from Windows
- How to install X−Win32
- Taming the X Display Manager
- Why Port Forwarding?; Port Forwarding; Secure forwarding of services with SSH
- Using Exceed X Server with SSH X11 Tunneling
- X11 Forwarding over SSH using X−Deep/32 and PuTTY
- GNOME Display Manager
- 10 minutes to an iptables−base Linux firewall; iptables command introduction
- Running X Window on MAC
- Securing Services on your system (Debian)
- Remote X using KDM (Caldera)
- HOWTO XDMCP in Gentoo Linux wiki page

---

7. Authors

Current: Thomas Chao, Alcatel–Lucent. <tomchao@alcatel-lucent.com>

8. Copyright Information

This document is copyrighted (c) 2000 – 2007 Thomas Chao and is distributed under the terms of the Linux Documentation Project (LDP) license, stated below.

Unless otherwise stated, Linux HOWTO documents are copyrighted by their respective authors. Linux HOWTO documents may be reproduced and distributed in whole or in part, in any medium physical or electronic, as long as this copyright notice is retained on all copies. Commercial redistribution is allowed and encouraged; however, the author would like to be notified of any such distributions.

All translations, derivative works, or aggregate works incorporating any Linux HOWTO documents must be covered under this copyright notice. That is, you may not produce a derivative work from a HOWTO and impose additional restrictions on its distribution. Exceptions to these rules may be granted under certain conditions; please contact the Linux HOWTO coordinator at the address given below.

In short, we wish to promote dissemination of this information through as many channels as possible. However, we do wish to retain copyright on the HOWTO documents, and would like to be notified of any plans to redistribute the HOWTOs.

If you have any questions, please contact <linux-howto@metalab.unc.edu>