XDM and X Terminal mini−HOWTO

Kevin Taylor

kevin@northants.lug.org.uk

Revision History

Revision v1.00	2002−05−16	Revised by: KT
Minor updates to all sections. Added details of Windows/Linux interoperability. Added section on starting XDM. Added details of KDM and GDM. Added new software resources.		
Revision v0.05	14 November 2000	Revised by: KT
Added cross−references to other Howtos		
Revision v0.04	6 November 2000	Revised by: KT
Updates after first public draft.		
Revision v0.03	3 July 2000	Revised by: KT
Minor updates from first comments		
Revision v0.02	28 June 2000	Revised by: KT
First SGML source draft from HTML source		
Revision v0.01	27 June 2000	Revised by: KT
First HTML source draft		

This document describes the basic ideas for using XDM to manage X terminals. It is not meant to be a comprehensive discussion of all the features of XDM, but a gentle introduction to what XDM can do.

For a full discussion about the installation and configuration of X terminals, please refer to the 'Thin−client' HOWTO, from the Linux Documentation Project or the Linux Terminal Server Project (see Section 7).

Table of Contents

1. Introduction

1.1. Copyright Information

1.2. 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.

1.3. New Versions

The latest version number of this document may be obtained from The Northants LUG, UK Project Pages .

1.4. Credits

Thanks go to the following people for help with information and proof reading of the document.

Scot W Stevenson for the original X terminal mini−howto document, from 1995, on which some of the material for the section on advanced xdm−config configurations was obtained.

Members of the Northants LUG, UK for proof reading the document.

The writers of the XDM, Xserver man pages and the default XDM scripts.

Credits for 0.05

Everyone who commented on the earlier versions, including Neil Zanella, Rafael Herrera, Paul Hornshaw, Clive Jones, Robert de Geus, Alex Schenkman, Richard Kaszeta, Malcolm Herbet

Credits for 1.00

The following people provided feedback and comments which have led up to the 1.00 release: Alan James, Stephen Eglen, Malcolm Hunter, Holgar Hoffman, Patrick Rynhart, Amedeo Lanza, Alexander Skopalik, Mike Banahan.

1.5. 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: `<xdm−mini@northants.lug.org.uk>`.

1.6. To do

I probably ought to mention some details about the following at some point ...

- Debugging problems with XDM.
- FAQs and common mistakes

---

2. Basic Concepts

2.1. What is covered

This document describes the basic concepts behind using XDM (the X Display Manager) to manage X terminals and X servers, in order to provide 'thin−client' computing, using Linux.

X (or the 'X Window System') is the windowing and graphics environment of choice for Unix systems. Its particular strength (and the key bit that we are interested in for this document) is that it separates the running applications (web browser, word processor, etc) from the actual graphics screen and input devices (mouse, keyboard, etc) via a network communications mechanism.

Essentially, this means that you can be running an application on one machine, but have its input and output redirected to another machine via a network. This is the key feature that makes an X terminal possible.

This document should be treated as a 'getting started with XDM' document, in that it describes the basic terms and concepts for using XDM and X terminals, with simple examples that provide the minimum amount of security.

The reader is advised to consult the list of resources provided at the end of the document in order to proceed beyond these basic facilities − in particular, the configuration of the 'authentication' and security settings should be examined, as the examples given in this document utilise the least secure modes of operation.

Please note − the majority of the information in this document was obtained from systems running Debian 2.1, SuSE 6.4, Mandrake 7.0 and RedHat 6.0.

This document does not discuss the installation or configuration of a network or X on Linux. Please refer to the appropriate HOWTO documents from the Linux Documentation Project for details (see Section 7).

This document also does not attempt to describe how to install and configure Linux for operation as an X terminal. For this information, please refer to the 'thin−client' HOWTO document, provided as part of the Linux Documentation Project, or the Linux Terminal Server Project (see Section 7).

2.2. About this document

This document came about because I wanted to experiment with Linux on a 486 PC as an X terminal to my main Linux box.

After reading the man pages, specifications and current howto documents relating to XDM and X terminals, I ended up getting really confused about where XDM was supposed to run and confusing XDM servers with X servers and the like, and so after an evening or two of experimentation, this document was born.

Once the basic terminology has been sorted out, the documentation for XDM and self−documenting sample files make very good reading − I just could not find a simple introduction to the basic concepts anywhere to get me started. Hopefully this document could prove to be a suitable introduction to someone in a similar position to me.

Oh, and in case you are wondering, a 486dx2/66 with 16 Mb RAM makes a fine X terminal!

3. XDM

3.1. What is XDM

Put simply, XDM (the X Display Manager) can be thought of as a graphical replacement for the command line 'login' prompt. In reality, it can actually do much more than that.

Typically, it would be started by the 'root' user (or the system startup scripts) on power up, and would present a user with a graphical login prompt. It will then manage the users X session once they login – i.e. it will initiate the running of their window manager and applications.

This could be considered a typical 'simple local machine login' configuration, as may be found installed by many Linux distributions by default. However, XDM can also manage remote X servers and provide login prompts to remote 'X terminals'. In short, it is not limited to the local machine – it can easily manage other machines connected via a network.

XDM is a very configurable utility and this document will only just 'scratch the surface' of what may be achieved. This document aims to provide enough information to configure your X terminals and application servers to connect to each other. The reader is referred to Section 7 for further information on the topics discussed here.

A note on security: X (in its default configuration) and XDMCP are not particularly secure. I am assuming that you are running X on a 'trusted' network and that security is not an issue. For details of how to tighten up your X connections (and more details about using the networking capabilities of X) please refer to the 'Running Remote X Applications' Howto document, which is also part of the LDP (see Section 7).

3.2. What is an X terminal

This term could be used to cover various configurations, but at its simplest, is a machine with a network connection, keyboard, mouse and monitor, configured to run the X Windows System to connect to an application server somewhere on the network.

There are several configurations of 'X terminal' with varying levels of functionality, ranging from completely diskless terminals to full X workstations.

3.3. Some Terminology

Before I go any further, I ought to explain the terms I will be using in this document. When talking about X, there is quite a lot of confusion over what is serving facilities to what. This is especially true when you are considering distributed sessions over a network involving X terminals. I will be using the terms described below.

Diskless X terminal

This would be a machine with no local disks, that would perform its boot up from an EPROM (or similar) and utilises a network connection to a server. It would obtain its network configuration, operating system, system configuration and all applications from the server. Once booted however, this would be the same as a 'dumb X terminal' (see below). Typically this configuration would use a combination of the following network protocols in order to boot: BOOTP, DHCP, TFTP, etc. Refer to Section 7 for some references that detail how to build diskless X terminals.

Dumb X terminal

This would be a machine that boots from its local disk into an operating system, and starts the 'X server' program and nothing more. Somehow, a login prompt would be provided on the machine, to enable a user to login to an 'application server' somewhere on the network.

X Workstation

This would be similar to a dumb X terminal, but would provide the option of logging on to the local machine itself, hence would be quite capable of becoming a standalone workstation (i.e. no network connectivity) if required. Most distributions can be configured 'out of the box' as a stand−alone X Workstation, with a graphical login prompt.

Application Server

In the context of this document, I use the term 'application server' to describe a machine that will provide the applications (X clients) that our X terminal will want to run. This can include everything from editors and browsers, through to the actual 'Window Manager' itself.

X Server

This is the program that manages the display of a machine with a physical console (display, keyboard, mouse, etc). It can be thought of as a combined graphics card, keyboard and mouse 'driver'. This will provide these facilities as a service to X clients (hence the term 'server'). Please refer to the X User Howto in Section 7 for more details.

X Client

This is an application that requires the use of an X server to access input (keyboard and mouse) and output (display). An X client cannot produce output without the services of the X server. The X server could be running locally (on the same machine, as is the case with an X workstation) or elsewhere on the network (as is the case with an X terminal connecting to an Application Server).

From the above descriptions, an X Workstation could be thought of as consisting of a dumb X terminal and application server running on the same machine.

This document will be looking at the architecture of the various options listed above and will describe the role that XDM can play in configuring them.

3.4. What can XDM do

XDM is responsible for providing the user with a login prompt and initiating their X session. It can manage local sessions (i.e. people logging into an X workstation) or sessions on remote machines, via a connection to an application server, from a diskless or dumb X terminal.

XDM would typically run on an application server, to permit users to logon and run applications from that server.

There are 2 main ways that XDM can interact with an X Server:

- XDM accepts queries from X server
- XDM manages X server

3.4.1. XDM accepts queries from X Server

The communications between XDM and the actual 'X server' (the machines with the physical screen/keyboard/mouse/etc) are handled via XDMCP the 'X Display Manager Control Protocol'.

This permits X servers to send out queries to servers running XDM. Effectively, the X server has to say 'I have someone wanting to login − please give me a login prompt'. In this mode of operation, XDM will not do anything unless it is asked to by your X server.

The query from the X server can take one of 3 forms:

- Direct query: the X server contacts a named host, requesting that XDM on that host presents a login prompt on its display.
- Broadcast: the X server sends out a broadcast message to the network, and the first server running XDM that responds to the broadcast will be the one to present the login prompt on its display.
- Indirect query: the X server contacts a named host, but asks it which other hosts it knows about on the network. The named host will then present the user with a list of hosts to choose from, and will then go on to initiate communications with the selected host resulting in the selected host presenting a login prompt on the X servers display.

There are several other options, but these will not be described here − refer to the XDM and XDMCP documentation in Section 7 for more details.

3.4.2. XDM Manages X Server

If you have a set of machines (e.g. diskless or dumb X terminals) that just end up running an X server, all designed to provide a login prompt to a single application server, then it is possible to configure XDM on the application server to connect back to each X server and present its login prompt on each display automatically.

In this mode of operation, XDM will actively 'push out' a login prompt to any listed X server that it knows about, without waiting for a query from the X servers themselves.

In this case, the configuration file 'Xservers' (see later) lists each machine (including the local display, if required) to which XDM should connect, to display its login prompt.

This configuration, when used with no remote X servers listed in the configuration, is the typical configuration used for a X workstation, in order to present a user with a graphical login to the local machine he is working on. As stated earlier, most distributions will support this configuration 'out of the box' in order to present the user with a local graphical login prompt.

Note: XDM must be permitted to connect to the X servers in question − so the access control on the X servers must be configured accordingly.

4. Configuring XDM

This section covers what needs to be configured for XDM to perform the functions described so far in this document.

In each case, the configuration described is the minimum necessary to accomplish each goal. In most cases this means that the configuration is also the least secure. Please refer to some of the additional documentation listed in Section 7 for information about securing XDM and X terminals (in particular the 'Running Remote X Applications Howto' from the LDP).

4.1. Configuration Files

This describes the following scheme of XDM configuration files:

- xdm−config
- Xaccess
- Xservers
- Xresources

These must be setup for the machine actually running XDM itself. They will typically be found in (Debian 2.1. Mandrake 7.0.2, RedHat 6.2):

```
      /etc/X11/xdm
```

or (SuSE 6.4):

```
      /usr/X11R6/lib/X11/xdm
```

xdm−config

Defines the names and locations of the other configuration files and the basic access permissions. For all distributions considered for this document, the file names were as listed here (but sometimes the locations varied).

This also defines the scripts to be run for the various state transitions for an X session, i.e. on startup, etc. You should not need to change these, as most distributions would appear to come with this pre−configured for you.

Note that XDM managed X sessions have a different set of startup and configuration scripts to X sessions started via xinit or startx (i.e. non−XDM managed X sessions).

Some distributions (e.g. Redhat 7.1) include the following line in this configuration file, which will prevent XDM from listening for queries:

```
        DisplayManager.requestPort: 0
```

This must be commented out as follows:

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

Xaccess

Determines which machines can connect to XDM − i.e. from which other machines on the network we are accepting XDMCP queries. If a machine is not listed in this file, then it will not be able to request a login prompt from XDM.

Xservers

Contains a list of machines that XDM will connect to, to provide a login prompt, automatically − i.e. those machines already running an X server, but would like this machine to provide the login prompt.

This is only required for 'XDM Managed X Servers'. You do not need any entries in this file if you will be relying on remote X servers to query XDM.

When running as a stand−alone 'X Workstation', there is usually a single entry in this file, listing just the localhost.

Xresources

Details of the X properties used by the XDM widgets (e.g. size of the login 'box', colours, bitmap backgrounds, etc).

4.2. Configuring XDM to Manage X Servers

An entry must be placed in the Xservers file for each X server that XDM will be presenting a login prompt on. This could include the local machine and/or a list of remote machines.

Example

```
# First the local host
:0 local /usr/bin/X11/X vt7
#
# Then the remote hosts
emma:0 foreign
alex:0 foreign
```

This will start XDM on the local machine and also present a login screen on the X servers running on the hosts 'emma' and 'alex' (assuming that permissions have been setup on 'emma' and 'alex' such that this machine is permitted to connect to the running X servers).

Note that it is possible to specify the host and display (:0, :1, etc) if required. This is useful, for example, if you are running multiple X servers on a single machine, etc.

4.3. Configuring XDM to Receive Queries

The file Xaccess determines which hosts may query XDM on this machine, in order to request a login prompt.

Example

```
# First line for direct queries
*
# Following line for indirect queries
* CHOOSER BROADCAST
```

This means that any host may request a login prompt via XDM (the first '*') using a direct query.

The 'CHOOSER' line specifies which hosts can connect to XDM using indirect queries − in this case, any host may query this machine for a list of potential hosts to connect to (the second '*' line).

'BROADCAST' means that the 'chooser' application on this machine will obtain its list of available servers (that will also be running XDM) via network broadcast queries. I will talk about the 'chooser' later.

It is possible to place specific host names or specifications of network IP addresses (e.g. a whole IP network or specific hosts) in these entries (and there are also other indirect queries possible, without using the chooser) but this is not described here (refer to Section 7 for some links to more information).

4.4. Starting X

The way you start the X server itself, will depend upon how you want it to interact with XDM locally and remotely.

X Workstation: XDM and local X server

XDM will normally start X automatically for you and XDM will usually be configured to run as part of the startup process (via the init scripts). Most distributions have a specific 'run−level' for starting the system up with a graphical login prompt.

The Xservers file would typically contain a single entry − that of the local host, and the Xaccess file would only need to permit access from the local host.

X Terminal: Remote XDM

Just start X with no clients, with access permissions such that the remote XDM is able to connect when it starts up. The following will start X with no access control:

```
/usr/X11R6/bin/X −ac
```

When the remote XDM is started, it will 'push out' a login prompt to all such configured X servers (as listed in its Xservers file).

X Terminal: Query a remote XDM

Recall there are 3 modes for queries: direct, indirect and broadcast (direct for a single host, indirect for a list of hosts or broadcast for the first host that replies):

```
/usr/X11R6/bin/X -query the.remote.host
/usr/X11R6/bin/X -indirect the.remote.host
/usr/X11R6/bin/X -broadcast
```

In each case, X will probably have to be started as root.

It is possible to have a machine automatically start X and perform a query for a running XDM on the network. One way is to 'hijack' the inittab setting for running as a graphical login (this is runlevel 5 on Debian and Redhat based systems, and 3 for SuSE − this example assumes runlevel 5 throughout). This is often the line beginning x:5 towards the end of /etc/inittab. Set this to (or add it if it doesn't exist):

```
x:5:respawn:/usr/X11R6/bin/X -broadcast
```

Replacing −broadcast with −direct or −indirect, etc. as required. You may have to change your default runlevel to be 5 too, (and then reboot), as follows:

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

4.5. Starting XDM

In a standard X workstation configuration, XDM would typically be started up by specifying the default initial run−level to be that corresponding to a full graphical login. On Redhat and Debian based systems this is usually runlevel 5. On SuSE it is run−level 3.

It is possible to run XDM automatically on startup by changing the default runlevel to that described above. This is configured in /etc/inittab as follows:

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

Alternatively it is possible to add a startup script for XDM itself to the rc scripts in the startup directories (/etc/rc.d/ for Redhat/Debian), to start and stop XDM in a similar manner to other services on a Linux machine.

The following script is suitable for a Redhat (and probably Mandrake) based system, and should be saved as /etc/rc.d/init.d/xdm. You will have to enable it using 'chkconfig −−add xdm'.

```
#!/bin/sh
# xdm start/stop script for RedHat based systems
#
# chkconfig: 234 60 60
# description: xdm permits remote users to logon to this X display
# processname: /usr/X11R6/bin/xdm
# config: /etc/X11/xdm/xdm-config

# source function library
. /etc/rc.d/init.d/functions
```

```
[ -x /usr/X11R6/bin/xdm ] || exit 0

prog=/usr/X11R6/bin/xdm

RETVAL=0

start () {
    echo -n $"Starting $prog: "
    # start daemon
    daemon $prog
    RETVAL=$?
    echo
    [ $RETVAL = 0 ] && touch /var/lock/subsys/xdm
    return $RETVAL
}

stop () {
    echo -n $"Stopping $prog: "
    killproc $prog
    RETVAL=$?
    echo
    [ $RETVAL = 0 ] && rm -f /var/lock/subsys/xdm
    return $RETVAL
}

restart () {
    stop
    start
    RETVAL=$?
    return $RETVAL
}

# See how we were called.
case "$1" in
     start)
    start
    ;;
     stop)
    stop
    ;;
     status)
    status $prog
    RETVAL=$?
    ;;
     restart)
    restart
    ;;
     condrestart)
    # only restart if it is already running
    [ -f /var/lock/subsys/xdm ] && restart || :
    ;;
     reload)
    echo -n $"Reloading $prog: "
    killproc $prog -HUP
    RETVAL=$?
    echo
    ;;
     *)
        echo $"Usage: $0 (start|stop|restart|condrestart|reload|status)"
        RETVAL=1
esac
```

```
exit $RETVAL
```

4.6. The Chooser Application

When XDM receives an indirect query, and assuming that the appropriate options have been specified in Xaccess for the 'chooser' application, it can provide the user with a list of other XDM managed servers that it knows about.

In this mode of operation, instead of the normal XDM login prompt, the user will be presented with a 'chooser' application, which will provide a list of detected hosts on the network that are currently accepting XDM connections.

When I first tried the use the chooser, I found that the Xresources files that came with my SuSE and Debian systems, specified a size for the chooser widget that was too big for the screens ... The following line from the Xresources file fixed that:

```
Chooser*geometry:      700x500+300+200
```

The chooser will obtain its lists of hosts by one of two methods:

- Broadcast Query: In this mode a request is broadcast over the network, and a list is built up from the replies received from other application servers running XDM.
- Explicit Listing: It is possible to provide a list of hosts for the chooser in the Xaccess file, as follows:

```
%hostlist       emma alex liam abigail
*               CHOOSER %hostlist
```

This will mean that the hosts emma, alex, liam and abigail will all be listed as candidates − even if one of the machines is down (there is often a button to 'ping' the host to see if it is running, before trying to connect to it).

Not that it is possible to include the localhost in the list of machines known to the chooser as well. XDM should be configured not to startup on the local console display though. Login should always be performed via an indirect query to the local chooser application, then the localhost should appear alongside any other hosts on the network.

4.7. Alternatives to XDM

Both KDE and Gnome have their own application to replace the standard XDM. They do similar things and are well documented. As far as providing remote X access, there is a single setting in the configuration file for the application to enable XDMCP support.

KDM: KDE Display Manager

The following must be set in the KDM configuration file (/usr/share/config/kdm/kdmrc for a Mandrake 8.1 system):

```
[Xdmcp]
Enable=True
```

GDM: Gnome Display Manager

The following must be set in the GDM configuration file (/etc/X11/gdm/gdm.conf):

```
[Xdmcp]
Enable=True
```

To have GDM running without starting the local X server, comment out the line

```
0=Standard
```

in the 'servers' section of the configuration file too.

5. Advanced Configuration Options

5.1. Configuration Sets

The xdm−config file provides a rich set of options, when it comes to defined scripts and other configuration files. In many cases, the defaults provided with your distribution should be fine, but for those of you who want more ...

The names of the startup scripts and configuration files used by XDM are determined by a series of statements in the top−level xdm−config file. This permits you to configure a different set of files for different X servers and X terminals, with different abilities.

For example, say you are using XDM to manage your local display, but also want it to accept queries from other X terminals on the network. It is possible to specify a different Xresources file for each of these cases, by using the following 2 lines in xdm−config:

```
DisplayManager._0.resources            /etc/X11/xdm/Xres_0
DisplayManager*resources               /etc/X11/xdm/Xresources
```

This will use Xres_0 for the local display (_0 is the XDM way of saying :0) and Xresources for everything else (the '*').

Similarly, if you wanted a particular resource file for a specific host, you would use an entry like the following:

```
DisplayManager.host_0.resources       /etc/X11/xdm/Xres_host_0
```

Note that XDM configuration files use the terminology host_0, where you would normally use host:0, to designate 'display 0 on host'.

If you look over your default xdm−config file, you will probably find that it has been setup so that your local X server has different files to the remote ones anyway, as different things must be performed on startup and reset of the server. My Debian file has the following for local servers:

```
DisplayManager._0.resources:    /etc/X11/xdm/Xresources_0
DisplayManager._0.setup:        /etc/X11/xdm/Xsetup_0
DisplayManager._0.startup:      /etc/X11/xdm/Xstartup_0
DisplayManager._0.reset:        /etc/X11/xdm/Xreset_0
```

and the following for remote servers:

```
DisplayManager*resources:       /etc/X11/xdm/Xresources
DisplayManager*setup:           /etc/X11/xdm/Xsetup
DisplayManager*startup:         /etc/X11/xdm/Xstartup
DisplayManager*reset:           /etc/X11/xdm/Xreset
```

5.2. X Resources

This document has only briefly touched on the available X resources, but I should mention that it is possible to fully configure XDM via the Xresources file.

The following may all be changed if required:

- Fonts, login prompt sizes
- Background graphics
- Window Titles, etc

There is a more detailed discussion of XDM resources, on Richard Kaszeta's web site (see Section 7)

6. Common Configurations

6.1. Linux to Linux

6.1.1. Stand−alone X Workstation

This has already been mentioned several times in this document, but in summary, it is a PC running XDM, with a single entry in the Xservers file, relating to the localhost (i.e. the local machine).

This is usually the default configuration provided by most distributions 'out of the box'.

6.1.2. X Terminal and Application Server

XDM runs on the application server:

- Xserver: Contains no entries
- Xaccess: Must permit the X terminal to connect

X terminal runs X using a direct query to the application server:

```
        /usr/X11R6/bin/X −query the.application.server
```

6.1.3. Group of Managed X Terminals

XDM runs on an application server:

- Xserver: Lists each X terminal to be managed
- Xaccess: Must permit each X terminal to connect

Each X terminal, just runs X, with suitable access control to permit XDM to connect to it.

```
        /usr/X11R6/bin/X −ac
```

6.2. Linux to Other Systems

It is possible to use a Linux X terminal to connect to another system running XDM. The same principles as above apply, but the specifics of configuring XDM (or its equivalent) will be specific to that system.

6.2.1. Linux and Solaris

You can run X on a Linux box, instructing it to query a Solaris machine as previously described:

```
/usr/X11R6/bin/X −query the.solaris.server
```

Note that you may have to configure X on the Linux machine to use the font server from the Solaris box. Although my Linux box connected and logged in fine without doing this, the fonts used by CDE were not displayed correctly.

I have not got this to work yet, as I don't have a Solaris box that I have any control over − but I am told that a font entry in /etc/XF86config similar to the following should work − you may have to change the port number from 7200 to something else (7100 has been quoted at me before). Can anyone confirm that this works?

```
FontPath "tcp/solaris.box:7200/all"
```

6.2.2. Linux and Windows

It is not possible to use X to remotely display Windows applications on a Windows box. It is possible to use X to display Windows versions of X applications on a Linux box, using a Windows X Server and Windows X applications (for example the XFree86 Win32 port − see Section 7)

It is possible to view Windows applications remotely on a Linux box using one of the following applications (which don't rely on X or XDM):

- Windows Terminal Services (WTS). RDesktop is a Linux application that understands the 'RDP' protocol used by WTS. This enables Linux to act as a client to WTS (see Section 7).
- Vitual Network Computing (VNC). This is an excellent platform independent remote desktop system that provides a bi−directional 'Windows or Linux' to 'Windows or Linux' networked desktop. It can be a bit slow, but works well (see Section 7).

  You can actually do quite strange things with VNC, such has having multiple machines connect and 'control' the desktop (and consequently 'fight' over control of the mouse :). It also doesn't maintain any state in the client, so you can leave your client, shutdown, bootup again, reconnect and carry on from where you left off. There is even a version of the viewer implemented as a Java applet, usable from any Java−enabled web browser.

6.3. Other Systems to Linux

If you have an X server for your system, it should be able to connect to a Linux XDM application server.

6.3.1. Solaris and Linux

I assume that you would run XDM on the Linux box as usual, and instruct X on Solaris to query the Linux box − does anyone have any information on this?

6.3.2. Windows and Linux

If you have an X server for windows that supports XDMCP queries, then it should be possible to configure it to query the Linux box. You should just run XDM on the Linux box as usual.

There are many commercial X Server implementations for Windows, and I will not list them all here. There is also a port of XFree86 to Windows, that makes use of the cygwin libraries (used to port many GNU/Linux tools to Windows − see Section 7). This works well.

The following batch file would start the cygwin XFree86 X server on Windows and connect to a Linux box (or any OS/machine running XDM), assuming a default installation of cygwin and XFree86 in c:\cygwin (save it as xdm.bat):

```
@echo off
if "%1"=="" goto noserver
goto allok
:noserver
echo Usage: xdm servername
goto end

:allok
set path=%PATH%;\cygwin\bin;\cygwin\usr\X11R6\bin
c:
chdir \cygwin\usr\X11R6\bin
XWin -query %1

:end
```

---

7. Resources

This section lists some resources that have been consulted in order to construct this document and which provide further details to the concepts described.

Many of the references listed below form part of the Linux Documentation Project (LDP): http://www.tldp.org/

The X Window System

- X User Howto (from the LDP)
- Running Remote X Applications Mini Howto (from the LDP)
- Man pages: X (main concepts), Xserver (X server concepts)
- X FAQ (on http://www.x.org/)

Thin−clients/X terminals

- Thin−client Howto Document (from the LDP)
- Linux Terminal Server Project: http://www.ltsp.org/
- Setting up LTSP systems (using quiet hardware): http://development.gbdirect.co.uk/xterminal.html
- Diskless Howto Document (from the LDP)
- X terminal pages: http://dns.solucorp.qc.ca/xterminals
- Richard Kaszeta: Converting old PCs into Standalone X Terminals: http://www.menet.umn.edu/~kaszeta/unix/xterminal/

XDM, XDMCP, etc

- Man pages: xdm
- XDMCP Howto Document (from the LDP)

Software

- Cygwin/XFree86 (Win32): http://www.cygwin.com/
- VNC: http://www.uk.research.att.com/vnc/
- rdesktop: http://www.rdesktop.org/
- X Servers for Windows (and other systems): http://www.rahul.net/kenton/xsites.html
- GPL Java X Server: http://www.jcraft.com/weirdx/
- X Terminal/Diskless specific distributions: http://lwn.net/Distributions/