



O06

Linux X Window System Part 1 of 2

Matilde L. Valdez

IBM @server xSeries
Technical Conference

Aug. 9 - 13, 2004

Chicago, IL

Notes:

X Window System

- Graphical User Interface of UNIX
- Initially developed at MIT
- Currently licensed by the X Consortium, Inc.
- In Linux implemented as a separate program that runs in user space
- Uses client-server architecture

Notes:

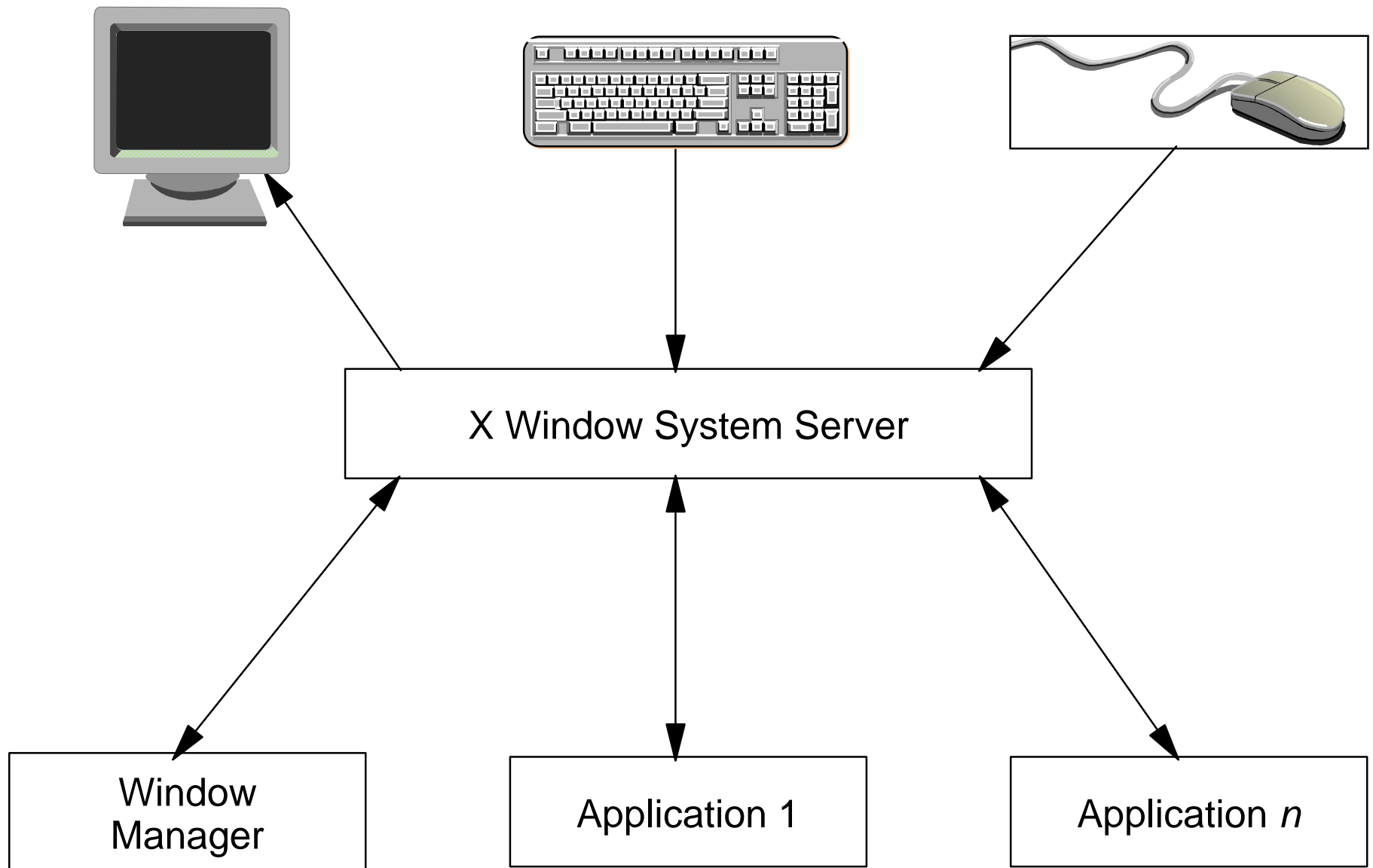
The X Window System uses a client/server architecture, which makes it very flexible. The central piece of software is the X server, which runs on the X station. This server traps all keyboard and mouse events, and sends them to the appropriate application. If an application wants to put something on the screen, it sends that data to the server, which then performs the necessary hardware calls to the graphical adapter.

Any application can connect to the X server, but there should always be one special application active: the window manager. This window manager basically puts a border around each application window, and allows you, for instance, to drag windows around the screen. There are numerous window managers available, each with their own style. Other applications also connect to the X server, and have their data displayed through it. Common examples are:

- xterm, which emulates a terminal screen, allowing you to enter Linux commands
 - xeyes, which displays a pair of eyes on your screen, looking at the mouse pointer
 - xbanner, which displays a background image
 - xcalc, a mathematical calculator
 - xedit, a GUI-based editor
- and many, many more.

The connection between the X server and the X clients (including the Window manager) is a TCP/IP connection. It is therefore possible to run the X client on another system.

Client/Server Architecture



Notes:

X Components

- An X Server
 - Controls keyboard, mouse and one or more screens
 - Controls resolution, refresh rate and color depth
 - Allows simultaneous access by several clients
 - Performs basic graphic operations
 - Forwards keyboard and mouse events to the correct clients
- An X Client
 - Is for instance an application
 - Receives keyboard and mouse inputs from server
 - Sends output to be displayed to server
- A Window Manager
 - Is a special X Client
 - Performs "windows dressing" on other clients
 - Allows other client windows to be moved, iconified and so forth

Notes:

X Servers in Linux

- Default X server in Linux: XFree86
 - Open Source
 - <http://www.xfree86.org>
- Other X servers are available for Linux
 - Metro-X: <http://www.metrolink.com>
 - Xi Graphics: <http://www.xig.com>

Notes:

XFree86

- XFree86 version 3.x (being phased out): Different binaries available for different brands of graphical adapters
 - XF86_Mono (Monochrome adapters)
 - XF86_VGA16 (Standard 16-bit VGA adapters)
 - XF86_SVGA (Super VGA adapters)
 - XF86_S3 (Adapters with S3 chip)
 - XF86_P9000 (Adapters with Weitek P9000 chip)
 - ...and so forth
- XFree86 version 4.x: One binary, that dynamically loads modules to support different graphical adapters

Notes:

About a year ago the XFree86 project released XFree86 version 4. Some distributions are currently already using this version, and other distributions are holding off a little because of some reported problems. That means that there are currently two different versions of XFree86 in production use.

XFree86 version 3 has been used for a number of years and is considered stable. It supports a large number of graphical adapters, and therein lies its biggest problem: Because of the support for all these adapters, a single binary image would be too large. That's why the XFree86 project releases multiple binaries, each with support for a number of related adapters. You need to install the binary that has support for your adapter before you can do anything.

This approach became more and more difficult to support. That's why the XFree86 project decided to use another approach for version 4. In this version, XFree86 consists of a single binary which is able to detect the adapter that is being used, and that can load the modularized support for that adapter in real-time. This makes installation and configuration easier.

XFree86 Configuration

- Can only be done as **root**
- You have to configure XFree86 for your
 - Graphical adapter, Monitor
 - Mouse
 - Keyboard
- Stored in `/etc/X11/XF86Config` or `XF86Config-4`
- Configuration aids to create this file
 - **XFree86 -configure** Integrated in XFree86
 - **redhat-config-xfree86** Red Hat tool
 - **SaX2, YaST** SuSE tools

Notes:

On every system which will run the XFree86 X server, the configuration file `/etc/X11/XF86Config` (or `/etc/X11/XF86Config-42`) file will have to be created. This file contains the hardware characteristics of the system running the server: graphical adapter type and characteristics, monitor characteristics, mouse type and keyboard type and language.

The correct setup of the configuration file is pretty complicated and very tricky, since incorrect monitor settings may damage your monitor. Let's repeat that: Incorrect monitor settings in `/etc/X11/XF86Config` or `/etc/X11/XF86Config-4` may damage your monitor! Don't say you weren't warned!

Most monitors today are Multi-Sync monitors, meaning that they accept a wide range of driving frequencies, and are protected against driving frequencies that would damage it. One exception is an LCD panel, which in a lot of cases only accepts a refresh rate of exactly 60 Hz. With all other refresh rates, the LCD panel simply will not show anything. Because of this, most configuration tools (see below) include a description for a "Generic LCD panel", for each of the most commonly used resolutions. If you've got an LCD panel, use one of these descriptions.

It used to be that you had to set up this file all by yourself, but nowadays there are several programs available that can help you out in about 99% of the situations.

Starting X

- To start XFree86 and your favorite window manager use **startx**
 - Will start XFree86 on a free virtual display (usually number 7)
 - Will start your favorite window manager
 - To start a second X session, use **startx -- :1**
- If you want to enable the graphical login screen, bring the system into **runlevel 5**
 - To switch manually use **init 5** command
 - To make change permanent, edit /etc/inittab:
id:5:initdefault:

```
$ startx
```

```
...
```

Notes:

Since Linux has a large number of virtual terminals, there is nothing keeping you from starting a second X session on another virtual terminal. This is accomplished by starting an X server on display ":1". When you start X via startx you need to make sure that startx understands that this is an option not for itself, but for X, so the full startup line will become `startx -- :1`.

Once you have started multiple X sessions, you can toggle between them with `<Ctrl-Alt-F7>` and `<Ctrl-Alt-F8>`.

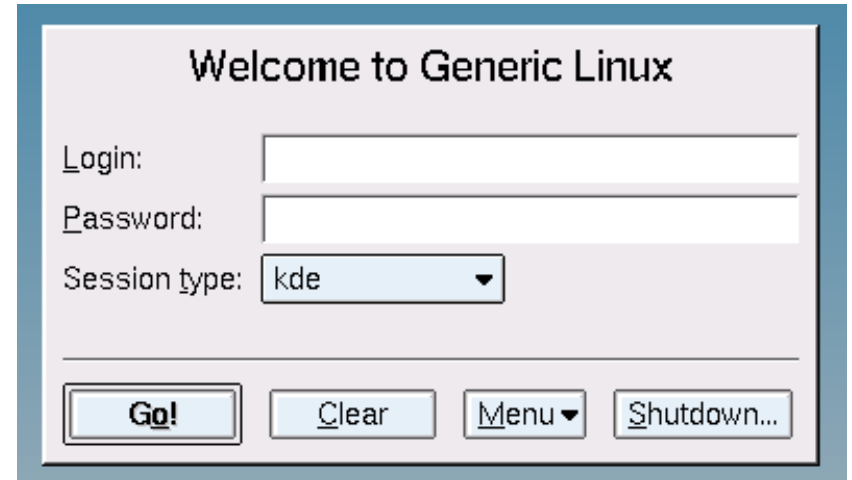
Stopping X

- Use **menu screens** from your window manager
 - Stops processes then stops XFree86
 - Usually saves current desktop layout
- If nothing works, use Ctrl-Alt-BackSpace
 - Stops XFree86 directly, other processes lose connection and die
 - Can be disabled in `/etc/X11/XF86Config[-4]`

Notes:

Session Managers

- Manage X-Sessions
 - Start XFree86
 - Offer a graphical login
 - Authenticate a user
 - Start the user's window mgr
 - Wait until user logs out
 - Restart XFree86
 - Offer a graphical login screen for the next user
 - and so forth
- Different session managers:
 - **x**dm
 - **k**dm
 - **g**dm
- Started from **init** in runlevel 5 (Red Hat) or as a regular System V service (SuSE)



Notes:

The K Desktop Environment (KDE)

The screenshot displays the KDE desktop environment. On the left is a blue sidebar with icons for Trash, Control Panel, and Home. The main workspace contains three windows: a Konqueror browser window showing the KDE website, a Shell terminal window, and a GIMP image editor window. The Konqueror window displays the KDE homepage with navigation links, a search bar, and a list of links under 'Learn', 'Download', and 'Develop'. The terminal window shows a root prompt. The GIMP window shows its standard toolbar. The taskbar at the bottom includes icons for the KDE logo, file manager, shell, and other applications, along with system tray icons for network, volume, and clock.

K Desktop Environment Home (kde.org) - Konqueror

Location Edit View Go Bookmarks Tools Settings Window Help

Location: <http://www.kde.org/>

kde.org (mirrors, i18n) | FAQ | Download | Screenshots | News

the **K** desktop environment **Conquer your Desktop**

KDE Home

KDE at Linuxtag

LINUX TAG 2002 June 6 - 9

Search

All Sites

Learn

- What is KDE?
- KDE 3.0 Info
- FAQ
- Documentation
- Screenshots
- Awards
- People

Download

- Stable Version
- Source Code
- KDE on CD-ROM
- Applications

Develop

- Developer Information
- Bug Reports
- Getting Involved
- Open Jobs

KDE is a powerful Open Source graphical desktop environment for Unix workstations. It combines ease of use, contemporary functionality, and outstanding graphical design with the technological superiority of the Unix operating system.

KDE is an Internet project that is truly open in every sense. Development takes place on the Internet and is discussed on our [mailing lists](#), USENET news groups, and IRC channels to which we invite and welcome everyone.

KDE is a mature desktop suite providing a solid basis to an ever growing number of [applications](#) for Unix workstations. KDE has developed a high quality development framework for Unix, which allows for the rapid and efficient creation of applications.

Announcements

The first beta of KOffice 1.2 is now available. Released on April 26th 2002, KOffice-1.2 Beta1 offers many new features and improvements, ... and certainly a few bugs to be found and reported.

KDE 3.0 Released!

On April 3rd 2002, the KDE Project released KDE 3.0, the Third-Generation of the Leading Desktop for Linux/UNIX, Offering Enterprises, Governments, Schools, and Businesses an Outstanding Free and Open Desktop Solution. A **changelog** has been made available. **Read more in the announcement**

KOffice 1.1.1 Released!

On December 17th, 2001, the KDE Project released KOffice 1.1.1, an update to KOffice 1.1. KOffice is a free, Open Source, integrated office suite demonstrating the richness and power of the KDE development environment. The **full announcement** contains links to the source and binary packages as well as a good deal of information about the current features of the KOffice packages. There is also a **list of changes** with respect to the previous version of KOffice.

[more announcements...](#)

root@kwik:~ - Shell - Kons... K Desktop Environment Ho... The GIMP

9:46 05/21/02

Notes:

The GNOME Desktop Environment



Notes:

X Networked

- Connections between different X Clients and the X Server are all TCP/IP connections
 - Can be run over a TCP/IP network
- Three levels:
 - Individual applications
 - Whole Session
 - Session Chooser

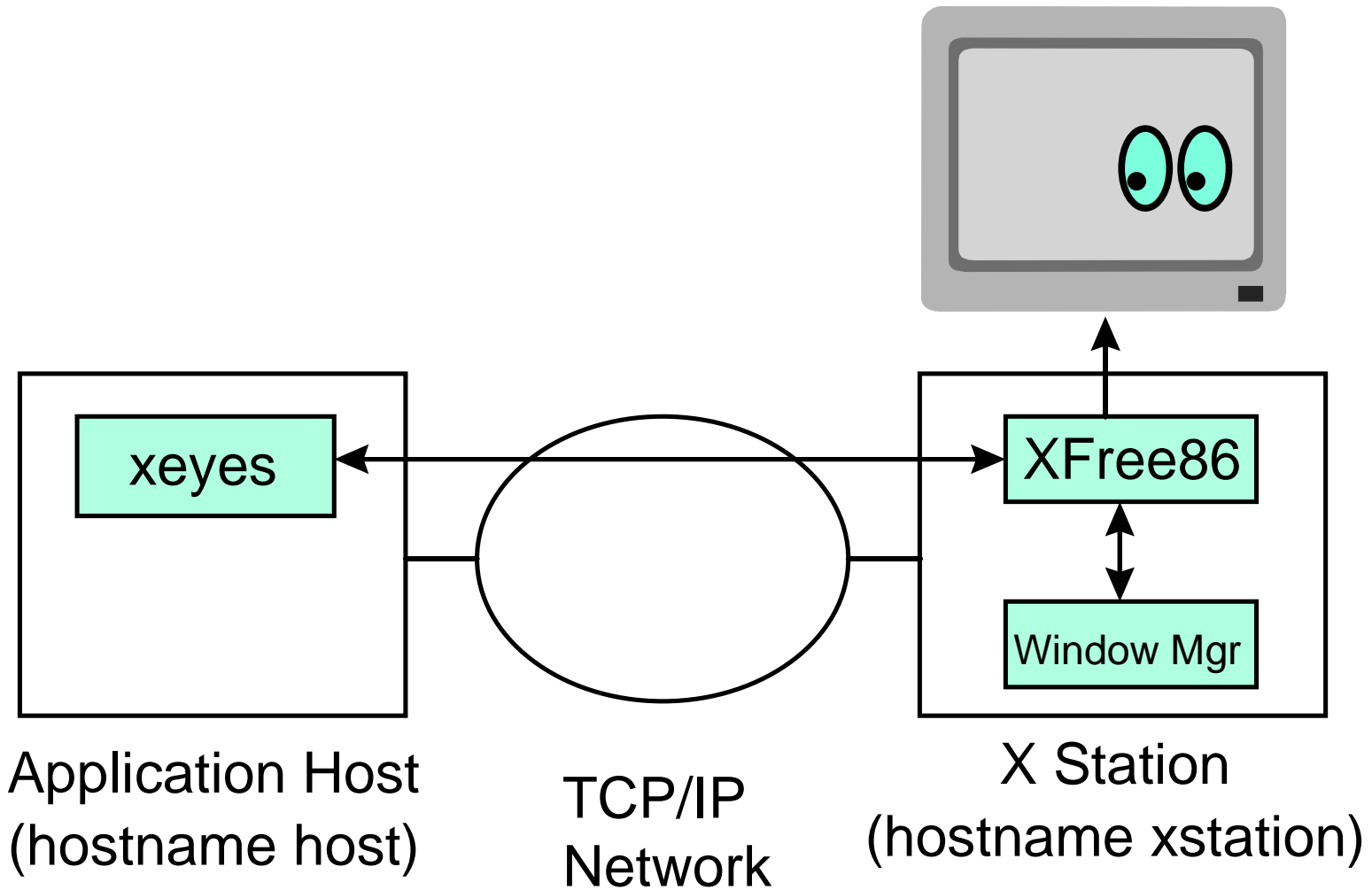
Notes:

All connections between the different X components (server, window manager, applications) are TCP/IP connections. This means that we can run them over a network too. And that opens up some interesting possibilities.

There are three levels of networking with X:

- The first level is by just running a single application over the network. This allows you to run an application on another system, but redirect the display to your local screen. This is very useful if that application is not supported or present on your local system.
- The next level is by running your whole X session over the network. In this case, all applications and your window manager are all running on a remote system. This is useful if you have disk- or dataless clients: clients that do not have any disk space to store data on, or do not have any disk at all. All user data and programs can be stored on a single server, and are run from this single server.
- The last level is by using a session chooser. In this case, before logging in, you get a list of servers that are willing to manage your session. This is very useful if you have multiple servers, and users need to be able to run their sessions from their local system on each of these servers.

X Applications Networked



Notes:

The visual shows the first level of networking X-applications. Both the XFree86 server and the window manager (and possibly other applications as well) are running on the local system. Only a single application is running on the remote host (the application server).

Applications over TCP/IP

- On the host (where X clients run):

```
host$ xterm -display xstation:0.0
  or:
host$ export DISPLAY=xstation:0.0
host$ xterm
```

- Displaying applications on a remote host is by default disabled for security reasons
- To enable this, two methods possible: **xauth** and **xhost**
- **xauth**: Uses cryptographic authentication method

```
xstation$ xauth extract xauthfile xstation:0.0
host$ xauth merge xauthfile
```

- **xhost**: Allows all connections from a given host

```
xstation$ xhost +host
```

Notes:

If you want to run an application from another server, then the only thing you basically need to do is start the application with a special option telling the application what X server to use.

This can be done using two methods:

- First, every X application will accept the `-display` option.
- Second, every X application will look at the `$DISPLAY` environment variable if no `-display` option is given, to determine the X server to contact.

The X server to contact is written as `<hostname>:<servername>[.<displaynumber>]`, with `<hostname>` being the IP address or hostname of the system where the X server is running, `<servername>` the instance of the X server to contact, and `<displaynumber>` the screen to use.

You can imagine that it is not desirable that the whole internet can redirect the graphical output of their commands to your screen. Therefore, doing this is by default disabled but can be enabled.

The first, safest method is by using the `xauth` mechanism. This works roughly as follows:

- When your X server is started, the startup scripts ensure that a random number, called the "authorization record" is generated. These records are stored in the `$HOME/.Xauthority` file.

- Any client who wants to connect to the X server needs to present this authorization record. If no or an invalid authorization is presented, then access is disabled.

Since normally all applications are started by the same person who started the X server, they all use the same `.Xauthority` file and present the right record.

- A client on a remote host obviously cannot access the `.Xauthority` file directly, so the authorization record needs to be transferred manually to that other host. This is a two-part process.

First, on the host where the X server is running, you need to extract the correct record from the `.Xauthority` file and store it in a file. This is done with the following command:
`xauth extract xauthfile client:0.0`

This means that the authorization record to connect to `client:0.0` needs to be stored in the file `xauthfile`.

You then transfer the file to the other system (using FTP, scp, rcp or any other means), and add it to the `.Xauthority` file there, with the following command:
`xauth merge xauthfile`

Any application started on this host, with the correct `-display` option or `$DISPLAY` environment variable set will now use this authorization record to connect to the X server.

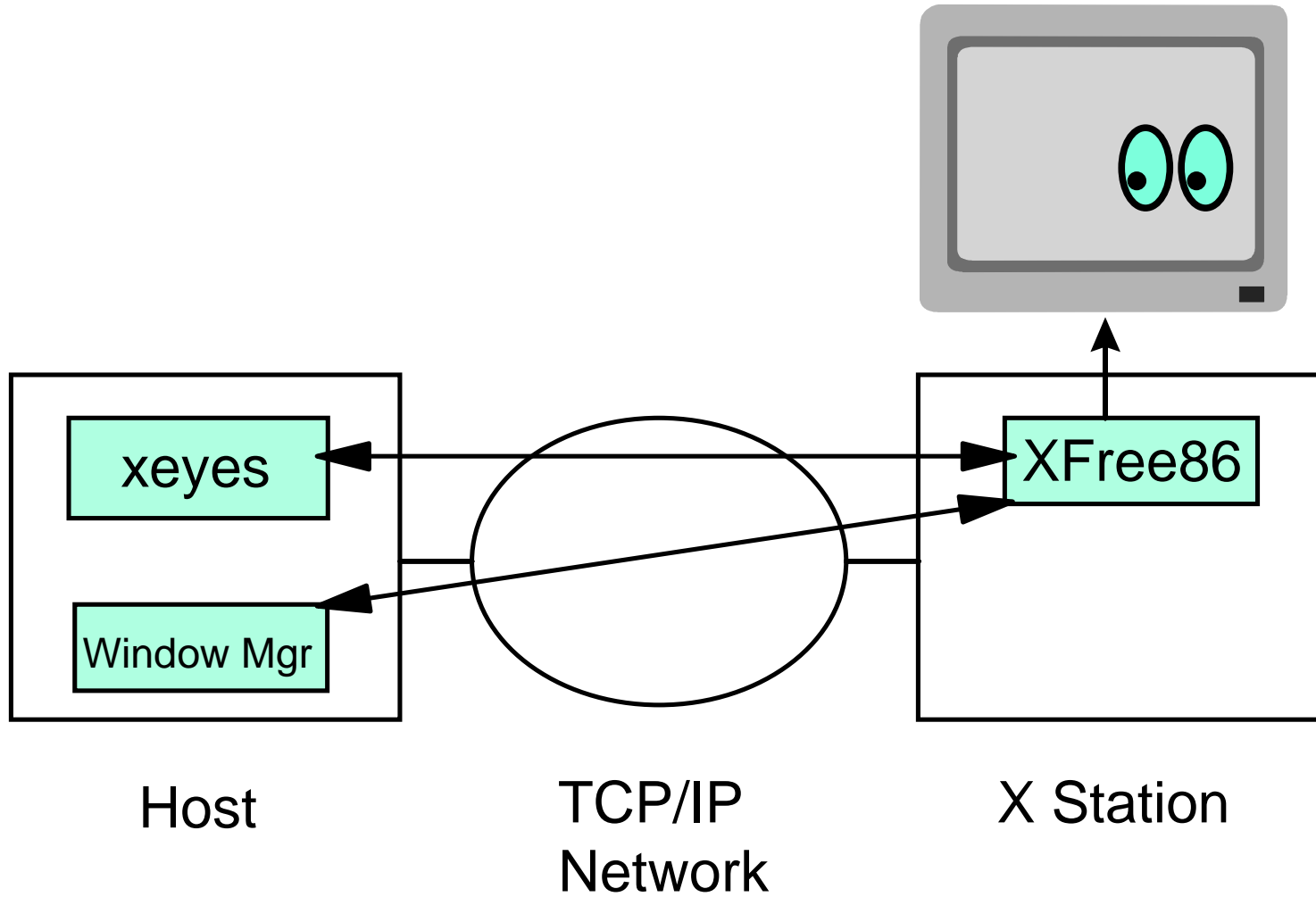
Of course, smarter ways of doing this are also possible. How about, for instance:
`xauth extract - client:0.0 | rsh host xauth merge -`
`rsh host xeyes -display client:0.0`

The smartest way however is to use ssh, since this protocol has the ability to automatically transfer the xauth record to the host, and set the `$DISPLAY` variable so that all data is transmitted via a secure session. This means that the only thing you need to do is: `ssh host xeyes`

The second method is less safe but more convenient. In this case, the user who has already started the X server issues the `xhost +<hostname>` command. This command allows all connections originating from `<hostname>` to succeed. This is obviously less secure, since every user on that particular host is now able to make a connection, not just the intended user. And this method is vulnerable to IP address spoofing and DNS poisoning.

Notes:

X Sessions Networked



Notes:

X Sessions over TCP/IP

- On most Linux distributions, X sessions are normally disabled for security reasons. To enable:
 - **xdm**:
 - Edit Xaccess and xdm-config
 - **kdm**:
 - Edit kdmrc and Xaccess
 - **gdm**:
 - Edit gdm.conf
- On the X Station:
X -query <hostname>

Notes:

In order to run your X-session over a network, you need to set up your display manager so that it accepts session requests over a network. How this is done depends on your session manager.

For xdm, there are two things you need to do:

- You need to edit the `/etc/X11/xdm/Xaccess` file so that it allows any host to get a login window. The line that specifies this is usually already there, but is commented out. So you just need to uncomment this line.
- You also need to edit the `/etc/X11/xdm/xdm-config` file because most distributions have set the XDMCP port to zero (meaning: invalid port) as a safety feature. This is usually done at the last line of this file, so if you comment out this line (with an exclamation mark), you've disabled this safety feature.

For kdm, there are again two things you need to do:

- You need to edit `/etc/kde/kdm/Xaccess` (Red Hat) or `/opt/kde3/share/config/kdm/Xaccess` (SuSE) so that it allows any host to get a login window. The line that specifies this is usually already there, but is commented out. So you just need to uncomment this line.
- You need to edit `/etc/kde/kdm/kdmrc` (Red Hat) or `/opt/kde3/share/config/kdm/kdmrc` and enable `xdmcp` direct and indirect requests.

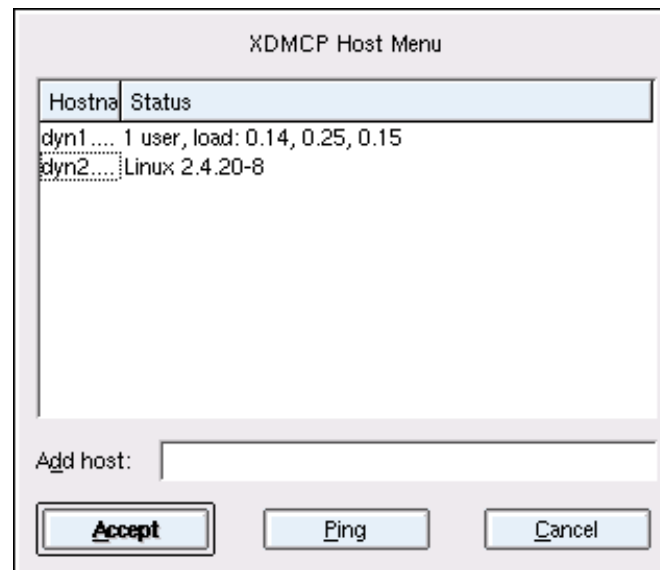
For gdm, the procedure is again different. Here, you only need to edit `/etc/X11/gdm/gdm.conf` (Red Hat) or `/etc/opt/gnome2/gdm/gdm.conf` (SuSE) to enable `xdmcp` direct and indirect requests.

When you're done setting up your display manager, you need to restart it. This is for instance done by switching to runlevel 3, and then back to 5 (`init 3; sleep 10; init 5`).

Then, you need to start the X server on the X station. Since the only program running here is XFree86, we can start it with the `X` command. We only need to tell it that it has to query the display manager to get a login prompt and a session. So the complete command becomes `X -query <hostname>`

Chooser Sessions

- If enabled, Display Managers do broadcasts to discover each other
- An "indirect" query shows a list of all Display Managers willing to manage your session
- To start an indirect session:
X -indirect <hostname>



Notes:

You can imagine having multiple display managers in your environment. In that case, it is very useful to be able to choose the display manager you are going to use. This is done using a chooser. Usually, this functionality is built into the session manager so we don't need to configure a separate program. We just call the session manager a little differently. If the session manager receives a so-called indirect query, it does a broadcast over the network to discover all systems that are willing to manage displays, and displays a list of these hosts. You can choose one of these hosts, and this host will then manage an X-session for you.

To start X and receive a chooser, the command line is `X -indirect <hostname>`

Font Server

- All fonts needed should be available to the X server
 - Thus, on any client workstation
- To save disk space, XFree86 can work with a font server
 - Central server, accessed via:
 - TCP port 7100
 - UNIX socket /tmp/.font-unix/fs7100
 - Configuration file /etc/X11/fs/config
- To use a font server, specify in XF86Config or XF86Config-4:

```
Section "Files"  
    FontPath "tcp/hostname:7100"  
    FontPath "unix/:7100"  
EndSection
```
- Red Hat uses a font server on *localhost* by default

Notes:

In general, X applications do not ask the X server (XFree86) to display individual pixels, but ask it to display complex structures like rectangles, circles, lines and so on. Furthermore, they can also ask the X server to display a certain character out of a fontset. This saves a tremendous amount of bandwidth.

For this to work, the X server needs to have available all the fonts an application would possibly use. Obviously this leads to a large management problem if multiple custom fonts are installed and used beyond the basic set.

To cope with this problem you can use a font server (xfs). This is a central server which holds all the fonts that are used in your organization. When XFree86 needs to display a font, it downloads it in real-time from the font server. This saves you from needing a large set of font files on each client workstation. SuSE, by default, does not use a font server. Red Hat configures and uses a font server on the local system by default, which is accessible through a UNIX socket only.

The specification in `/etc/X11/XF86Config` on a Red Hat system will thus look like this:

```
Section "Files"  
FontPath "unix/:7100"  
EndSection
```

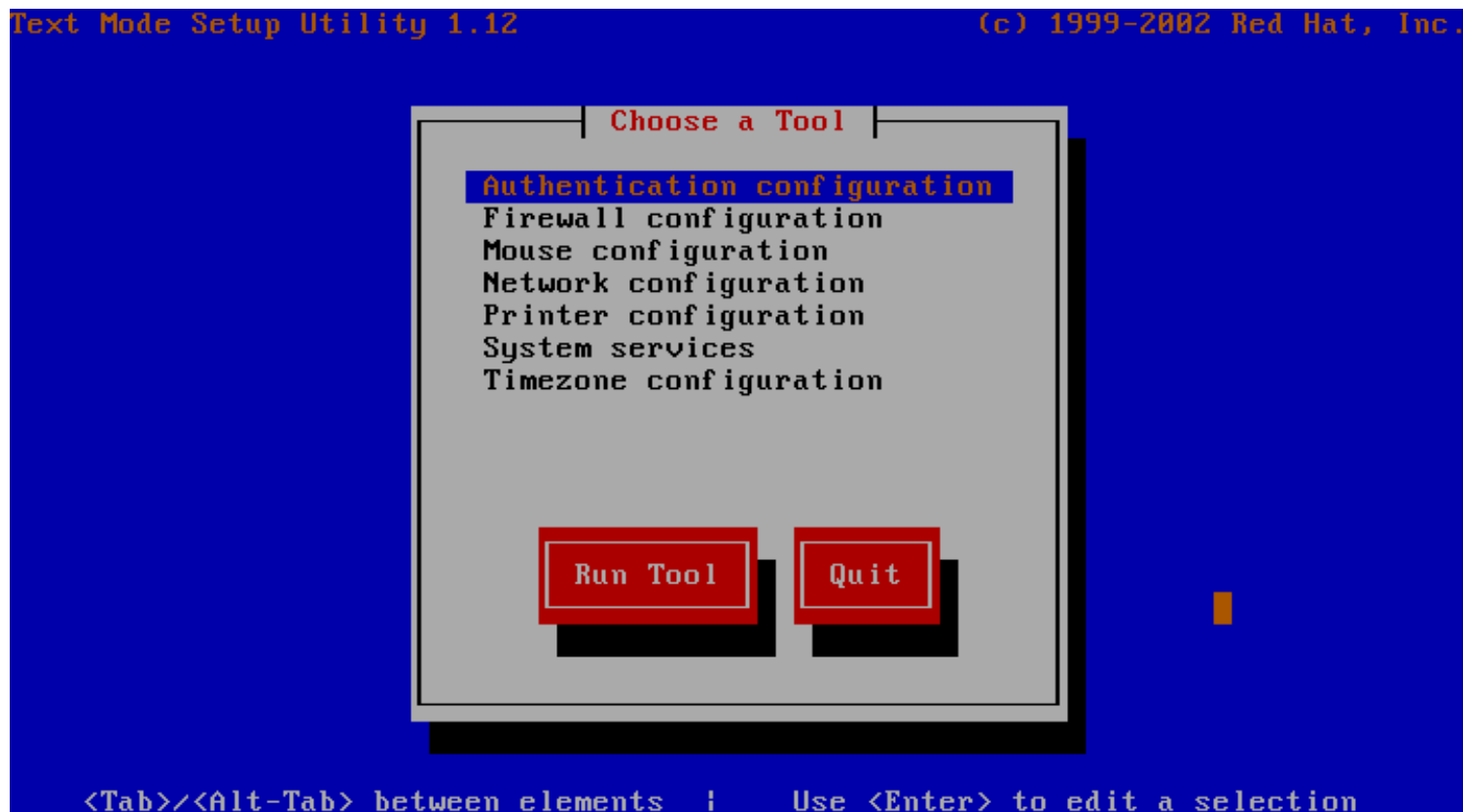
In order to use a font server over the network, you specify it using the following syntax in the `/etc/X11/XF86Config` file:

```
Section "Files"  
FontPath "tcp/hostname:7100"  
EndSection
```

Depending on your distribution, you also might need to enable the font server to serve network requests. Some distributions disable this by default in the xfs configuration file `/usr/X11R6/lib/X11/fs/config`.

Red Hat "setup"

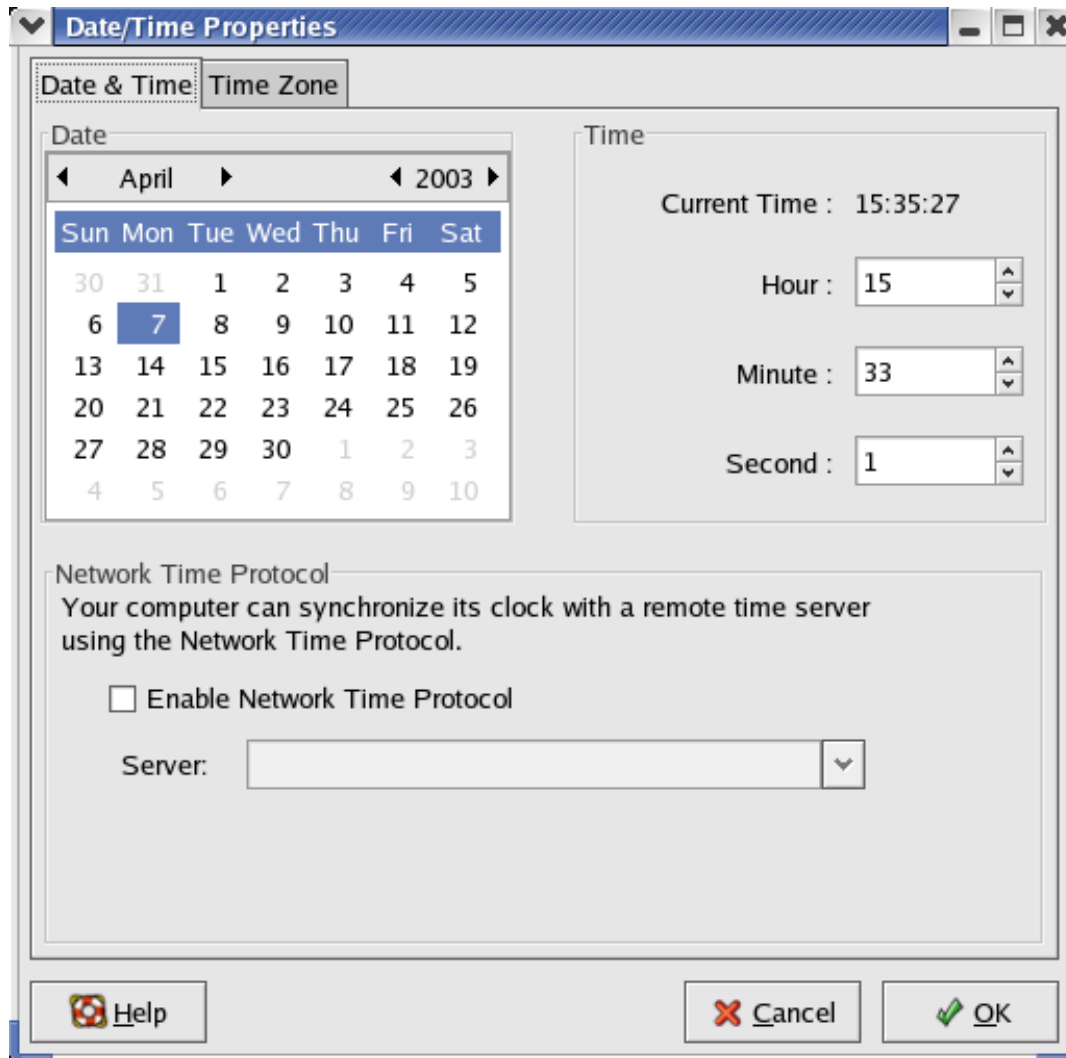
- Menu-based front-end for various tools that are part of the text-based installation



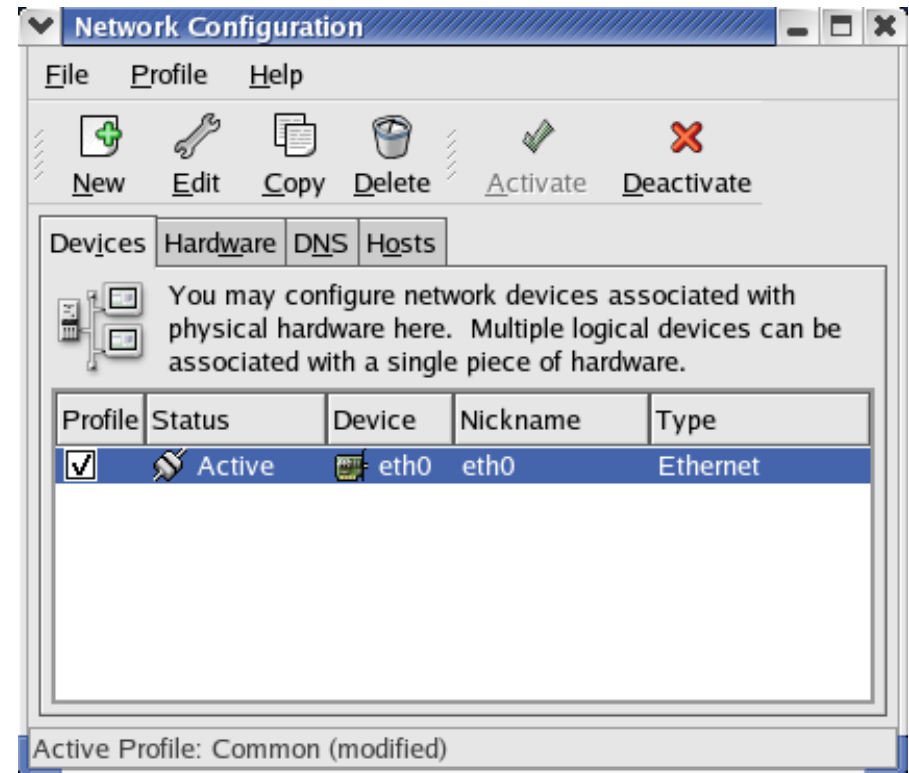
Notes:

Red Hat "redhat-config-*)"

- GUI-based tools for system management



redhat-config-time



redhat-config-network

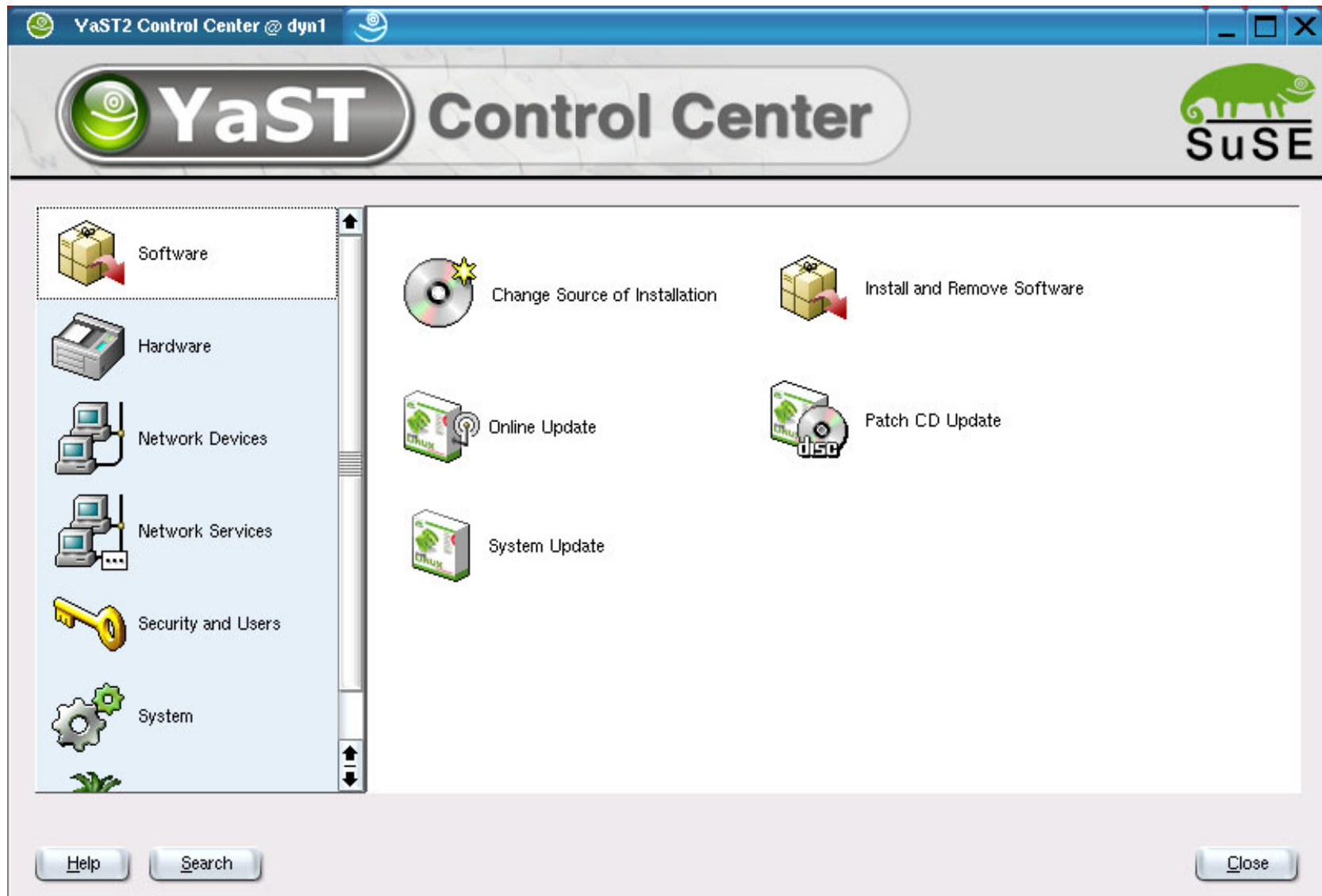


redhat-config-soundcard

Notes:

SuSE "YaST"

- Yet another Setup Tool



Notes:

Webmin

- <http://www.webmin.com>
- Open Source initiative to create an independent configuration framework
- BSD Open Source License
- Uses modules to configure specific items
 - Modules can be created by anybody, using any license
- Support for all major Unix versions, not just Linux
- Web-based interface only
- Not installed on all distributions by default
 - May need to install yourself

Notes:

Webmin Installation

- Download `webmin-version.rpm` from <http://www.webmin.com>
- **`rpm -ivh webmin-version.rpm`**
- Start web browser and connect to port 10000
- Login with **root** password

Notes:

Webmin Screenshot

Webmin Feedback Log Out

Webmin System Servers Networking Hardware Cluster Others

Help.. Module Config Search Docs..

Users and Groups

Local Users

[Create a new user](#) [Create, modify and delete users from batch file](#)

Username	User ID	Real name	Home directory	Shell
root	0	root	/root	/bin/bash
bin	1	bin	/bin	/bin/bash
daemon	2	Daemon	/sbin	/bin/bash
lp	4	Printing daemon	/var/spool/lpd	/bin/bash
mail	8	Mailer daemon	/var/spool/clientmqueue	/bin/false
news	9	News system	/etc/news	/bin/bash
uucp	10	Unix-to-Unix CoPy system	/etc/uucp	/bin/bash
games	12	Games account	/var/games	/bin/bash

Notes: