



6

X Window System

CERTIFICATION OBJECTIVES

- 6.01 X Server
- 6.02 X Font Server
- 6.03 X Clients
- 6.04 Remote Display of X Applications
- 6.05 Desktops and Window Managers

- 6.06 The X Start Process
- 6.07 Basic Graphical Applications



Two-Minute Drill

Q&A Self Test

One of the most important aspects of getting a Linux system up and running is configuring the user interface. As RHCEs and RHCTs are expected to configure computers for non-administrative users, the Red Hat exams test your ability to configure the X Window System. The *X Window System* is the foundation of the Linux graphical user interface (GUI). While the GUI plays an integral part of other operating systems such as Microsoft Windows, the X Window System on Linux is essentially just another application.

Many administrators don't even bother with the GUI; the command line interface is enough for most administrative purposes. However, regular users on a Linux workstation are more productive with the GUI and the multitude of X Window-based applications. If you are helping users migrate from Microsoft Windows to Linux, the X Window System allows you to provide a less intimidating environment.

Not all Linux computers require the X Window. For example, computers that are used as dedicated DHCP, DNS, or NFS servers generally don't serve as workstations for anyone and therefore don't need any sort of GUI. Many Linux gurus have a negative bias toward the GUI. While Red Hat and others have developed some helpful GUI tools, they are almost always "front ends," or programs that customize one or more commands at the command line interface.

But if you're administering a network of Linux computers for regular users, you'll need to know how to administer the X Window System, a skill that requires a basic understanding of the available desktops and graphical applications.

This chapter starts with the X Server, as configured on the local computer. It continues with X Clients, as generic applications that you can run from the local or a remote network computer. Once everything is configured, you're ready to take a step back to the start process for the X Window. It moves on to the two major Linux graphical desktops, and finally goes into an overview of a very few of the available graphical applications.

CERTIFICATION OBJECTIVE 6.01

X Server

The X Window System is designed as a flexible and powerful client/server-based system. In order to configure and troubleshoot the X Window interface, it is important you understand the client/server nature of the X Window System.

INSIDE THE EXAM

The Linux Graphical User Interface

The Red Hat exams test your ability to configure the X Window, presumably for non-administrative users. As stated in the Red Hat Exam Prep guide, RHCE candidates have to successfully complete *all* RHCT Troubleshooting and System Maintenance requirements, including configuring the X Window System and a desktop environment. Naturally, you need to know how to configure the X Window on a local computer. The X Window may fail if you have problems with the X Font Server.

This is a situation where it may be more efficient to use the Red Hat GUI X Window

Display Settings configuration tool, which you can start with the **redhat-config-xfree86** command. The language of the X Window configuration file is difficult to master. As time is of the essence on the Red Hat exams, you should use the method that does the job most quickly for you.

The X Window System can work over a network. Once properly configured, you can run GUI applications from a remote computer. To make this work, you need to know understand modularity of the X Server and X Clients, as well as the way X Window security is managed on your network.

exam

Watch

I use the Display Settings tool and `redhat-config-xfree86` command interchangeably; the command is the fastest way to start the tool.

As you might have guessed from the terms *client* and *server*, the X Window System is designed to work in a networked environment. The client and server can both reside on your own computer or on separate computers on the network. In other words, not only can you run X applications on your system, you can run X applications on other

computers on your network. The graphical displays from those remote applications are sent to your monitor.

In fact, X Window applications handle this task so well that, providing the network is fast enough, you really can't tell from a performance point of view which applications are running locally and which applications are running remotely.

When you configure the X Server, I'll show you the modularity of the system. In brief, components such as keyboards, mice, and monitors are configured separately and all become modular components of the X Server.

Different Meanings for Client and Server

Normally on a network, the local computer is the client, and the remote computer acts as the server. X Window Clients and Servers work on a different paradigm. The X Server controls the graphics on the local computer. The X Server draws images on your screen and takes input from your keyboard and mouse. In contrast, X Clients are local or remote applications such as **xclock** that you can run on the local X Server.

You can run an X Client locally or remotely. Local X Clients run on your workstation; remote X Clients run on the local X Server. When you run a remote X Window Client application, you start the program on a different computer and send its output to use the X Server on your local computer. Figure 6-1 illustrates a local X Server with one local and one remote X Client.

Before we can talk about running X Client applications, we need to look first at configuring an X Server on our system. While RHEL 3 normally configures most hardware configurations during the installation process, you need to know how to edit the X Window configuration file, directly in a text editor or with the Red Hat configuration tool (**redhat-config-xfree86**).



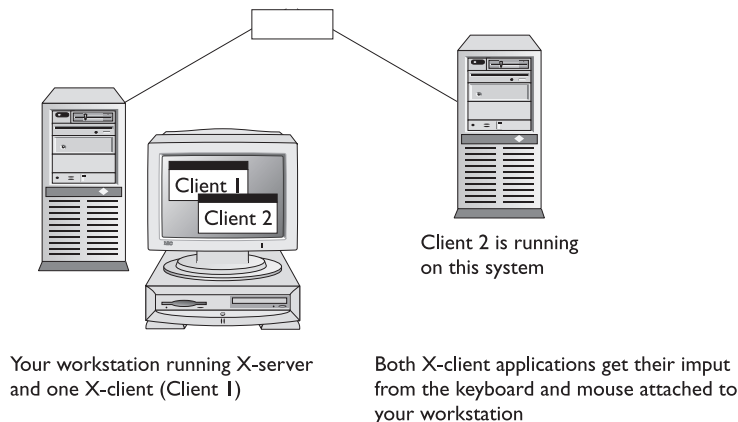
If you're in the GNOME or KDE desktops, you can also start the Red Hat Display Settings tool with the Main Menu | System Settings | Display command.

Supported Hardware

If a Linux installation program does not successfully configure the X Window System, it can be difficult to configure the GUI on a Linux workstation. Fortunately, RHEL 3 comes with tools and drivers that make this job relatively painless and easy.

FIGURE 6-1

Running X Window Clients from remote or local computers





But it all depends on the hardware. Linux may not always work with the latest and greatest video card or monitor. Many video cards and monitors include proprietary software; it may take some time before Linux developers are able to “reverse-engineer” these components. For the latest official information, check the Red Hat Hardware Compatibility List described in Chapter 1.

Linux provides world-class support for graphics. The list of movie studios that use Linux to create feature films is impressive (Disney, DreamWorks, Industrial Light and Magic, Paramount, and so on). If you need motion-picture quality graphics support, you may want to consider some commercial alternatives to XFree86. Two are Metro-X from Metro Link (www.metrolink.com) and Accelerated-X from X-Inside (www.xinside.com).

Hardware: X Server Selection

The X Window Server program shipped with RHEL 3 is an open-source X Server program called XFree86. The XFree86 Server supports hundreds of video cards and monitors. The best place to check to see whether your video card and monitor are supported is the Red Hat Hardware Compatibility List, currently available from hardware.redhat.com/hcl.

Alternatively, navigate to the XFree86 Web site at www.xfree86.org to find the latest support information. As installed, RHEL 3 uses the XFree86-4.3.0 package; you can find more about compatible hardware at www.xfree86.org/4.3.0/Status.html.



As of this writing, RHEL 3 includes only the XFree86 (4.3.0) server. There may be a few video cards that require the older XFree86 server, version 3.3.x. Hopefully, these cases are now rare. If you find that you are unlucky in this way and cannot upgrade your video card, first try a standard VGA or VESA (also known as SVGA) configuration. As a last resort, download the older XFree86 Server from www.xfree86.org.

Servers and XFree86

The latest version of XFree86 includes modules for different video servers. Unlike the XFree86 3.x series, this one doesn’t require you to install different packages for each video server. Hardware support for most video servers is already there. If you learn of updates, changes are easy. Just add the module, and then point to it in the `/etc/X11/XF86Config` configuration file.

If you are using an unsupported video card, support is also included for generic VGA devices. Most video cards and monitors will work with these X Servers.



Recent versions of Red Hat Linux included an `/etc/X11/XF86Config-4` configuration file. When Red Hat removed support for XFree86 version 3.x servers, it changed the name of the default X Window configuration file to `/etc/X11/XF86Config`.

Tools for X Configuration

If you want to configure your X Window System, you have two options. You can edit the X Window configuration file directly, you can use the Red Hat Display Settings tool (`redhat-config-xfree86`), or you can install and configure the X Window during the installation process.



Older configuration tools, including `xf86config`, `Xconfigurator`, and `XF86Setup` are not included with RHEL 3 (or Red Hat Linux 9).

Even if you didn't install any graphics software when you installed RHEL 3, you can still use the `redhat-config-xfree86` command. When you do, it starts its own default graphics mode if it detects a graphics driver.

Red Hat Display Settings Tool

The Red Hat Display Settings tool is a stand-alone program that you can run at any time from the command line. The basic routines which start with the `redhat-config-xfree86` command are also used by the Red Hat installation program if you choose to install and configure the X Window System at that time.

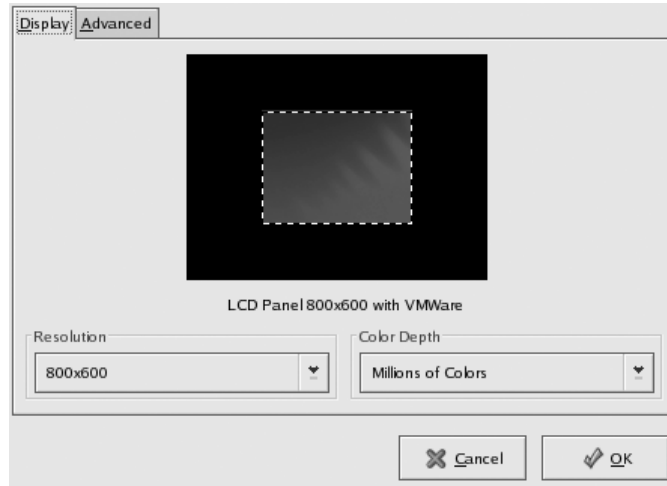
The `redhat-config-xfree86` program is a character-based menu-driven interface that helps you to configure your video hardware. If you're starting from a text console, it automatically probes your video card and selects the appropriate X Server image. If `redhat-config-xfree86` cannot detect your graphics card, it allows you to select them from the list of supported video cards.

Running the Red Hat Display Settings Tool

It's easy to start the Red Hat Display Settings tool. Just type `redhat-config-xfree86` at a command line interface. It provides a simple GUI, even if you start it from a regular text console. When you start `redhat-config-xfree86`, you'll see the Display Settings window similar to that shown in Figure 6-2.

FIGURE 6-2

The Display Settings tool, started from the text console



While the look of the Display Settings tool is slightly different when started from the text console or the GUI, the functionality is identical.

If the Display Settings tool successfully identifies your hardware, you'll see it listed under the Display tab. In this case, it detected a VMWare graphics driver with a LCD monitor. You can now set the default resolution and color depth under the Display tab. You can change these settings by clicking on the associated drop-down text box. If your hardware supports it, you can configure:

- Monitor resolutions between 640×480 and 1920×1440.
- A color depth of 256, thousands, or millions of colors. Thousands corresponds to 16-bit color, and millions corresponds to 24- or 32-bit color, depending on the capability of your hardware.

But if you want to select a different hardware component, you can select it from a list. Click the Advanced tab, as shown in Figure 6-3.

You'll see options to configure your monitor and video card. Click the Configure button in the Video Card section. This should bring up the Video Card Settings dialog box shown in Figure 6-4. If the Probe Videocard option is active, the Display Settings tool may be able to configure your card for you. Otherwise, search through the list of video cards. If you do not see your graphics card here, it may not be supported. In this case, you have several options:

FIGURE 6-3

Display settings

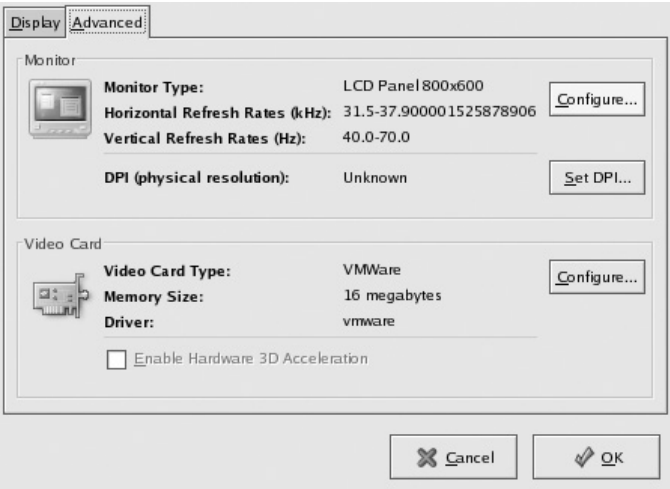
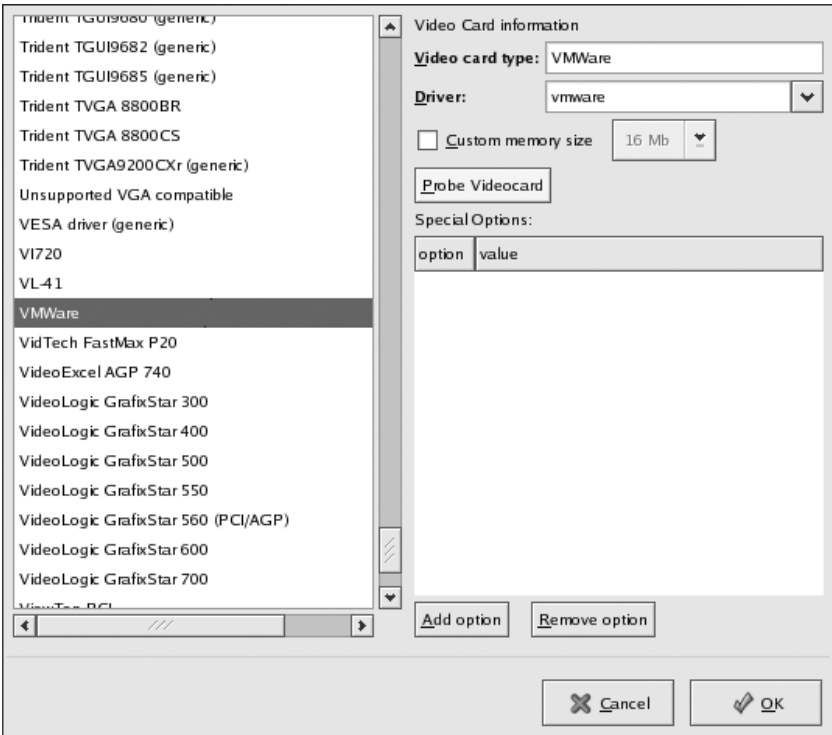


FIGURE 6-4

Selecting a graphics card

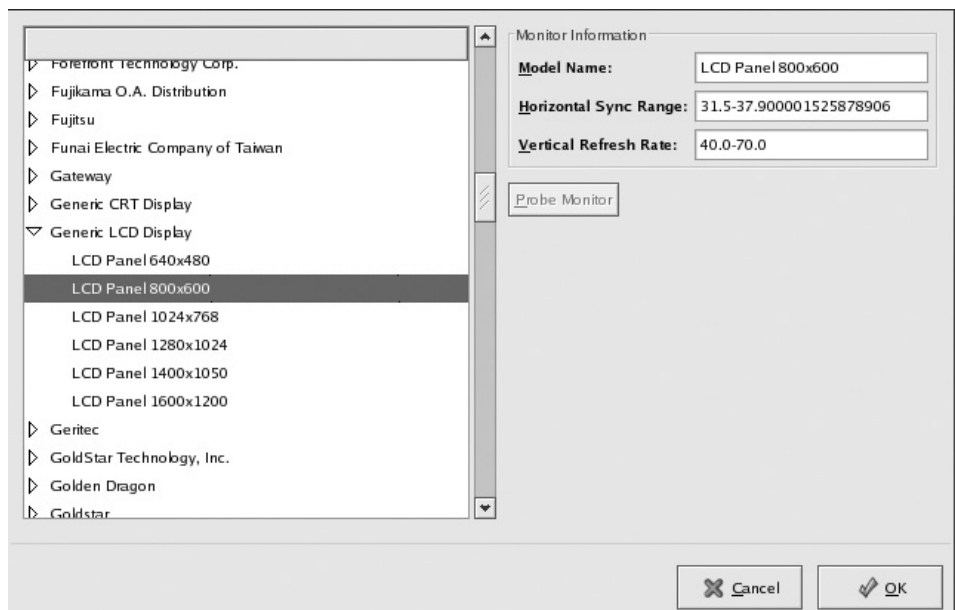


- Select a video card similar to your model. Alternatively, you may find a generic server such as VESA driver (generic) that is compatible with your video card. Test and if necessary edit the `/etc/X11/XF86Config` file to complete your changes.
- Check the Web for other Linux users who are running the X Window System with the same type of hardware. A useful database is available by searching through newsgroup messages at groups.google.com.
- Use the Unsupported VGA compatible X Window Server.
- Select one of the more generic cards. With the Custom Memory Size option, you can configure between 1MB and 128MB of video RAM. You can also add one of the many configuration options associated with the `/etc/X11/XF86Config` file.
- Go to www.xfree86.org and download the latest drivers. You'll need to edit the `/etc/X11/XF86Config` file directly to point to this driver.

Once your selections are complete, click OK. This returns you to the Advanced tab. If your video card is so capable, you'll be able to activate the Enable Hardware 3D Acceleration option. Next, configure the Monitor. Click the Configure button in the Monitor section of the Advanced tab. This brings you to the dialog box shown in Figure 6-5.

FIGURE 6-5

Selecting
a monitor



If the Probe Monitor button is active, the Display Settings tool may be able to probe your hardware. Otherwise, select the option that most closely matches your monitor. If you don't see the make and model for your monitor, select one of the "generic" monitors. You can then customize the Horizontal and Vertical settings per your monitor's documentation.



Don't select Probe Monitor if you have an older monitor. And don't configure a horizontal sync or vertical refresh rate beyond the capability of your monitor. Otherwise, your video card may send signals that exceed the capability of your monitor. Especially on a laptop computer, the consequences can be rather costly.

Laptops

Configuring the X Window System to run on a laptop can be more challenging than configuring it to run on a desktop system. If you are planning to install Red Hat Enterprise Linux on a laptop, a good source for tips and additional information is the Linux on Laptops Web site at www.linux-laptop.net.

Configuring the X Window System

With some older or sensitive hardware, configuring the X Window System may be a risk. If the refresh rate that you set is too high, you could potentially damage your hardware. The refresh rate is the frequency at which your graphics card redraws images on your screen.

Refresh rates are expressed in terms of hertz (Hz). A refresh rate of 60 Hz means that an image is redrawn 60 times in one second. Computer monitors have both vertical and horizontal refresh rates. Modern monitors, known as multisync or multiscan monitors, automatically adjust to the vertical and horizontal refresh rates as driven by the graphics card. In either case, refresh rates vary from monitor to monitor.

If the monitor you configure does not match your actual hardware, it is possible for your video card to overtax your monitor, resulting in a blown monitor. This is a special concern with laptop computers. Laptop display screens are expensive; it is often more cost-effective to replace a laptop computer instead of replacing just the laptop display screen.

It's just as easy to overtax your monitor with any of the Microsoft Windows operating systems.

Using a Two-Button Mouse

The X Window System was designed to work with a three-button mouse. Many PCs come with only a two-button mouse. With the Mouse Configuration tool, you can configure your Linux to emulate the middle button in the GUI by pressing both buttons simultaneously.

Alternatively, if you have a mouse with a scrolling wheel, you may already have a middle mouse button. If you have a mouse of this type, try pressing the scroll wheel. If it clicks, RHEL 3 has probably already configured it as a middle mouse button.

In many cases, the Red Hat installation program automatically configures the scroll wheel. But if you have a problem, you may need to edit the `/etc/X11/XF86Config` file directly. The following is an excerpt from my version of this file. It describes the mouse settings, with the scroll wheel enabled.

```
Section "Input Device"
    Identifier "Mouse0"
    Driver      "mouse"
    Option      "Protocol" "IMPS/2"
    Option      "Device"    "/dev/psaux"
    Option      "ZAxisMapping" "4 5"
    Option      "Emulate3Buttons" "no"
EndSection
```

“Mouse0” helps identify the mouse in other parts of this configuration file. The driver is `mouse`; the device is `/dev/psaux`. The protocol, `IMPS/2`, is associated with a standard PS/2 mouse. And the `ZAxisMapping` variable translates motion from the wheel to standard software mouse buttons 4 and 5, which corresponds to up and down motion in the active window. Mouse buttons 4 and 5 exist in the software even if you have only two physical buttons on your mouse.

X Configuration Files

While there are a wide variety of X Window configuration files in the `/etc/X11` directory, the primary X Window configuration file is `XF86Config`. It’s instructive to read the associated man page carefully. It is well documented and includes a number of commented sample commands that can help you configure your system in a number of special ways. For example, it includes tips on how you can

- Configure different keyboards
- Set up multiple monitors, in what is known as a “multi-head” configuration

- Disable switching from the GUI to other virtual terminals with the following command:
`Option "DontVTSwitch" "on"`
- Disable the CTRL-ALT-BACKSPACE key sequence, which normally exits from the GUI with the following command:
`Option "DontZap" "on"`

This is just a small sample of the things you can do with your X Window configuration. It's possible that you'll see a problem like this on the Troubleshooting and System Maintenance exam. You may also be asked to configure one of these options during the Installation and Configuration Exam.

Starting the X Window

The default server is linked to the **X** command. X is called up by the **startx** command. Both of these commands are located in the `/usr/X11R6/bin` directory. Examine the long listing for the **X** command. You'll see that it's linked to the version 4 all-in-one server package, XFree86:

```
# ls -l /usr/X11R6/bin/X
lrwxrwxrwx    1 root    root          7 Jul 30 07:33 /usr/X11R6/bin/X ->>
XFree86
```

The Display Settings tool normally writes its changes to `/etc/X11/XF86Config`. When the XFree86 X Server starts, it reads this file.

EXERCISE 6-1

X Server

In this exercise, you will start your X Server without a window manager. You'll then start an X Client application known as **xterm**. Some of the commands used in this exercise are covered later in the chapter. If the X Window System is not running, you can skip steps 1 and 3.

1. If the X Window System is running, change to a text console by pressing CTRL-ALT-F1.
2. If you see a login prompt, log in at the text console as root. Otherwise, press CTRL-C to stop the X Window.

3. If you logged in as root at the text console, stop the current X Window Server with the following command:

```
# init 3
```

4. Start the XFree86 X Server by typing the command:

```
# X &
```

Your X Server will start, but all you will see is a blank gray screen, with an “x” that represents your mouse cursor.

5. Switch back to your text console session by pressing CTRL-ALT-F1.
6. Type the following command:

```
# xterm -display localhost:0.0 &
```

(Note: **xterm** starts with a lowercase **x**, and there is only one hyphen before the display switch.)

7. Switch back to your X Window display by pressing ALT-F7.

You should now have an **xterm** terminal window. Select the window and enter commands at the **xterm** command line. Check out the contents of the `/usr/X11R6/bin` directory. Try starting other X Client applications such as **xcalc** and **xclock** from the **xterm** command line. Reboot your system to return things to normal.

One last keystroke hint for X: pressing the CTRL-ALT-BACKSPACE keys sends a termination signal to the X Server. In some situations, the Linux GUI crashes and it's not possible to start a text console session by pressing CTRL-ALT-F1. In this case, the CTRL-ALT-BACKSPACE key combination can keep you from having to reboot your computer.

EXERCISE 6-2

Multiple X Servers

In this exercise, you will start two different X Servers. If the X Window System is not running, you can skip steps 1 and 3.

1. If the X Window System is running, change to a text console by pressing CTRL-ALT-F1.
2. If you see a login prompt, log in at the text console as root. Otherwise, press CTRL-C to stop X Window.

3. If you logged in at the text console, stop the current X Window Server by typing this:

```
init 3
```

4. Start the XFree86 X Server by typing the command:

```
# startx &
```

Your normal GUI will start.

5. Switch back to your text console session by pressing CTRL-ALT-F1. Return to your GUI by pressing ALT-F7. Switch back again to your text console session. (If these commands do not work, check for a DontVTSwitch option in your /etc/X11/XF86Config file.
 6. Type the following command:

```
# startx -- :1 &
```
 7. You should now have two different GUIs. Switch to the first GUI by pressing CTRL-ALT-F7. Return to the second GUI by pressing CTRL-ALT-F8.
 8. Log out from both GUIs.
-

XF86Config in Detail

As you learn the intricacies of configuring the Linux X Window Server, it's worth some trouble to examine the /etc/X11/XF86Config file in detail. It's interesting to note that these files are almost identical with RHEL 3 on my VMWare computer and with Red Hat Linux 9 on my desktop computer. The only differences are associated with the different graphics hardware and monitor.

The first line is a comment that indicates whether your X Window system was last configured by the Red Hat installation program or the Display Settings tool. The two options are:

```
# XFree86 4 configuration created by pyxf86config
# XFree86 4 configuration created by redhat-config-xfree86
```

The first section binds the other sections together. In other words, you'll see sections later in this file, each associated with an Identifier variable: Screen0, Mouse0, Keyboard0, and DevInputMice.

```

Section "ServerLayout"
    Identifier      "Default Layout"
    Screen 0       "Screen0" 0 0
    InputDevice    "Mouse0" "CorePointer"
    InputDevice    "Keyboard0" "CoreKeyboard"
    InputDevice    "DevInputMice" "AlwaysCore"
EndSection

```

The next section identifies the location of RGB (red, green, blue) style colors for display in the X Window. It also identifies the standard TCP/IP port for the X Font Server. That provides a key clue: the X Font Server can communicate over a network. You may have problems if a firewall is blocking this port.

```

Section "Files"
    RgbPath        "/usr/X11R6/lib/X11/rgb"
    FontPath       "unix/:7100"
EndSection

```

The following section describes font and server extension modules. These are fairly standard; you can review a full list of possible modules in the `/usr/X11R6/lib/modules` directory (or subdirectories).

```

Section "Module"
    Load "dbe"
    Load "extmod"
    Load "fbdevhw"
    Load "glx"
    Load "record"
    Load "freetype"
    Load "type1"
    Load "dri"
EndSection

```

Next, we'll look at a couple of different sections, both labeled `InputDevice`. The first configures your keyboard as a regular US keyboard with 105 keys. The second configures a PS/2 mouse, as described earlier. The `Emulate3Buttons` option, when active, configures the X Server to activate middle mouse button functionality when you click the left and right mouse buttons simultaneously.

```

Section "InputDevice"
    Identifier      "Keyboard0"
    Driver          "keyboard"
    Option          "XkbRules" "xfree86"

```

```
Option      "XkbModel"  "pc105"
Option      "XkbLayout"  "us"
EndSection

Section "InputDevice"
Identifier   "Mouse0"
Driver       "mouse"
Option       "Protocol"   "IMPS/2"
Option       "Device"     "/dev/psaux"
Option       "ZAxisMapping" "4 5"
Option       "Emulate3Buttons" "no"
EndSection
```

The following InputDevice is included by default, to support the simultaneous use of a USB mouse:

```
Section "InputDevice"
Identifier   "DevInputMice"
Driver       "mouse"
Option       "Protocol"   "IMPS/2"
Option       "Device"     "/dev/input/mice"
Option       "ZAxisMapping" "4 5"
Option       "Emulate3Buttons" "no"
EndSection
```

The following Monitor and Device sections are associated with the monitor that you configured during the installation process or with the **redhat-config-xfree86** command. Since no Monitor Vendor is specified, this is associated with some generic monitor.

```
Section "Monitor"
Identifier   "Monitor0"
VendorName   "Monitor Vendor"
ModelName    "LCD Panel 800x600"
HorizSync    31.5 - 37.9
VertRefresh  40.0 - 70.0
Option       "dpms"
EndSection

Section "Device"
Identifier   "Videocard0"
Driver       "vmware"
VendorName   "Videocard vendor"
BoardName    "VMWare"
```

```

        VideoRam      16384
    EndSection

```

As you can see, the Screen section associates the “Monitor” and “Device” as configured by Identifiers Videocard0 and Monitor0. If the Red Hat installation program or the Display Settings tool gives you the wrong DefaultDepth or inappropriate screen sizes (Modes), you can add them here.

```

Section "Screen"
    Identifier      "Screen0"
    Device          "Videocard0"
    Monitor         "Monitor0"
    DefaultDepth    24
    SubSection "Display"
        Depth       24
        Modes       "800x600" "640x480"
    EndSubSection
EndSection

```

Many graphics systems support the 3D direct rendering interface, as shown:

```

Section "DRI"
    Group      0
    Mode       0666
EndSection

```

CERTIFICATION OBJECTIVE 6.02

X Font Server

Assuming your X Server is properly configured, you should be able to start a GUI with the **startx** command. One of the things that could go wrong is with the X Font Server. The X Window System needs fonts. Linux manages fonts through the X Font Server. RHEL 3 manages the X Font Server with the **xfs** service script. Generally, you can't start the Linux GUI unless **xfs** is running. Many different fonts are normally available in the `/usr/X1R6/lib/X11/fonts` directory.

The X Font Server can be an Achilles' heel for X Window. A number of things can go wrong with **xfs**. If you can't start your X Window, check the status of your X Font Server with the following command:

```
# service xfs status
```

If the X Font Server is not running, you'll want to go through the following troubleshooting list:

- The **xfs** service could be stopped or dead. In this case, you may need to try restarting **xfs**.
- The **xfs** service might not be set to start in your current runlevel; you can inspect and change this with the appropriate **chkconfig** command.
- The filesystems with **/tmp** or **/home** could be full. The **xfs** service can't start if either of these filesystems is full. There may also be problems if **/tmp** is on a different physical hard disk from other X Window files.

exam

Watch

Do not confuse the X Font Server service script, *xfs*, with the filesystem with the same initials, which was developed for very large files by Silicon Graphics.

- Fonts could be misconfigured in **/etc/X11/XF86Config**.
- Fonts could be missing from the default **/usr/X1R6/lib/X11/fonts** directory. For example, missing 100-dpi or 75-dpi fonts could cause applications in a Linux GUI to look strange.

Any of these problems could make it impossible for you to start a Linux X Window. In addition, if you've set up a default X Window login, these problems could keep you from getting to the graphical login manager.

exam

Watch

The GUI doesn't work unless *xfs* has started in your current runlevel. For example, if *xfs* isn't running

in runlevel 3, the *startx* command won't work from that runlevel.

EXERCISE 6-3

X Font Server

In this exercise, we'll look at the effect of a problem with the X Font Server on starting the GUI in Linux.

1. If you're in the GUI, exit from it. If you've configured RHEL 3 to start from the text login interface by default, that is easy. Just click Main Menu | Log Out, and click OK to confirm your logout from the GUI.

2. If you have configured RHEL 3 to start the GUI by default, you'll have to open up a command line interface. Right-click on the desktop. In the pop-up menu that appears, click New Terminal, and run the following command:

```
# init 3
```

3. Check on the status of your X Font Server. If your GUI works, it should be operational. Next, deactivate your X Font Server:

```
# service xfs status
xfs (pid 2560) is running . . .
# service xfs stop
Shutting down xfs: [OK]
#
```

4. Try starting the GUI again with the **startx** command. Watch what happens. Look for the “Fatal server” error message.
5. What other things could keep the X Font Server from starting on your computer? Think about elements such as the service configuration utility, or a full /tmp or /home filesystem. You may need to go through this process to diagnose problems during the Red Hat exams.
6. Restart the X Font Server.

CERTIFICATION OBJECTIVE 6.03

X Clients

Once you've configured a working X Server, you are (almost) ready to start connecting to various X Window Client applications. An X Window Client is an application program. This client uses your X Server's graphical services to display output. While one X Server process controls the display, you can run as many X Clients as your hardware resources, primarily RAM, will support. If your Linux system is part of a network, you may also start X Clients on other systems on the network and have those clients send their displays to your X Server.

X Clients exist for almost every basic application. There are X Clients for word processing, spreadsheets, games, and more. The Red Hat GUI configuration tools were developed as X Clients. There are even X Client versions of popular utilities such as the emacs editor.

Default X Clients

When you configure a workstation for most users, you'll be configuring their GUI. You may need to specify a default desktop such as GNOME or KDE. I'll describe these desktops later in this chapter. You may want to set up specific icons for that user's desktop. Settings for default X Clients are stored in each user's home directory, in various hidden directories. In RHEL 3, KDE configuration files are stored in each user's `~/kde` directory (where the tilde, `~`, represents a Linux home directory). GNOME configuration files are stored in hidden directories such as `~/gconf`, `~/gnome`, and `~/gnome2`.

Towards the end of this chapter, I'll show you the GUI tools that you can use to help you configure a workstation quickly during the Red Hat exams.

X Clients and Command Line Options

Starting an X Client is very easy. When you start the X Window System for the first time on RHEL 3, it puts X Client icons on the desktop by default. You can start additional X Clients by selecting a program from a menu, or you can start an X Client from a command line terminal screen. Later, I'll show you how to set up default X Clients when you start the Linux GUI.

It's almost as easy to start running X Clients from remote servers. All you need to do is give permission. As discussed later in this chapter, the `xhost +client` command on your Linux computer allows a remote computer named *client* to run X Clients on your local X Server.



You may wish to reread and practice with the commands in this section after reading the next section on the Remote Display of X Applications.

X Clients are standard Linux applications, which you can start with commands such as `xclock`, `xeyes`, `xterm`, `xclipboard`, etc. If you choose to start an X Client from a command line, you can follow the command name with any number of options. Most X Clients understand a common set of options. These options are used to control such things as the size and location of the X Client's window, the font the application uses to display the text, and even the display on which the application should display its output. Table 6-1 lists some of the more useful options you can supply when you start an X Client from the command line.

The behavior of most of the command line options in Table 6-1 is self-descriptive, but we need to take a more detailed look at how some of the options work. We will examine the `-display` option in more detail when we look at running remote X Clients.

TABLE 6-1 Common X Client Command Line Options

Option	Example	Result
<code>-display server:0.0</code>	<code>-display frodo:0.0</code>	Send output to the X Server running on the computer named frodo.
<code>-geometry XSIZE×YSIZE+XOFF+YOFF</code>	<code>-geometry 100×100+10+20</code>	Specify the size and location of the window. In this case, we want a window 100×100 pixels in size, offset from the upper-left corner by 10 pixels horizontally and 20 pixels vertically.
<code>-font fontname</code>	<code>-font lucidasans-14</code>	Display text for this client using a specific font.
<code>-background color</code>	<code>-background blue</code>	Set the window background to blue.
<code>-foreground color</code>	<code>-foreground white</code>	Set the window foreground to white.
<code>-title string</code>	<code>-title "My Window"</code>	Place a title on the client window's title bar.
<code>-bordercolor color</code>	<code>-bordercolor green</code>	Make the window border green.
<code>-borderwidth pixels</code>	<code>-borderwidth 5</code>	Make the window border 5 pixels wide.

The `-geometry` option is used to specify both the size of the window that the X Client starts up in and the location of the window. Notice that the first two numbers, the *XSIZE* and the *YSIZE*, are separated by a lowercase “x.” These two numbers specify the size of the client window in either pixels or characters, depending on the application. For example, if you are opening an **xterm** window, the size represents a terminal screen with *XSIZE* columns and *YSIZE* lines. If you are starting an **xclock**, the size represents a window *XSIZE*×*YSIZE* in pixels.

The next two numbers specify where you want the client window to appear on your display. The numbers are relative to the upper-left and lower-right corners of the desktop: +0+0 represents the upper-left corner, -0-0 represents the lower-right corner. These specs are shown in Table 6-2.

Therefore, for *XOFF* +*YOFF*, if you specify +10+ 10, the client is positioned 10 pixels from the left edge of the screen and 10 pixels from the top of the screen.

TABLE 6-1

X Client
Geometrical
Positioning

<i>XSIZE</i> × <i>YSIZE</i>	Description
-0-0	Lower-right corner
-0+0	Upper-right corner
+0-0	Lower-left corner
+0+0	Upper-left corner

Alternatively, `-10-10` positions the client 10 pixels from the right edge of the screen and 10 pixels up from the bottom of the screen.

The `-font` option specifies the font that the X Client should use to display text. The X Window System comes with a wide variety of both fixed and proportionally spaced fonts. The default list is located in the `/usr/X11R6/lib/X11/fonts` directory. This directory contains a number of subdirectories, each of which contains font files for the various types of fonts installed on your system.

Many of the X Client command line options enable you to specify colors for different parts of the client window. You can specify a simple color such as red, green, white, and black. Alternatively, you can specify a color by indicating the red, green, and blue components of the color:

```
# xclock -background RGB:FF/00/FF
```

xterm

One of the most useful X Clients is a program called **xterm**. As its name implies, **xterm** is an X Client application that creates a terminal window on your X display. So, after all the hard work you've gone through to get a nice windowing display, you're right back where you started, with a command line interface.

The difference is that now you can start up as many of these command line interfaces as you like, and you can switch between them with the click of a mouse. Since **xterm** is an X Client, you can even open terminal windows on other computers on your network and have them display to your desktop. You can start **xterm** either from a menu or from a command line prompt.

While the two major desktops include their own versions of **xterm**, Red Hat has configured **gnome-terminal** as the default command-line interface within the default GNOME desktop. It is what is started when you right-click on the desktop and select **New Terminal** from the pop-up menu.



In RHEL 3, the default GUI command line interface for the KDE desktop is **konsole**, which you can start when you right-click on the desktop and select **Open Terminal** from the pop-up menu. Functionally, you can run commands in **konsole** just as you can run commands within **gnome-terminal**.

CERTIFICATION OBJECTIVE 6.04

Remote Display of X Applications

One of the most powerful features of the X Window System is its networking support. The X Window System was designed to run in a networked environment. If you are a system administrator responsible for a number of RHEL 3 systems, you don't have to run to the server room every time you want to run a GUI administration tool. With the X Window System, you can connect to any number of systems and redirect the output from X Clients running on those systems back to the X Server running on your desktop.

Security

Before we configure clients and servers to transmit graphical applications, let's examine some of the basics of X Window security. When working with remote X applications, all other Linux security features are still in effect. You will still need a user account and password in order to connect to the remote system to start an X Client. You need to make sure that a firewall isn't blocking access.

Part of the job of the X Server running on your local computer is to listen for requests from X Clients that want to send their output to your display. Those requests can come from applications that are running on your local computer, or they can come from applications running on another computer on your network.

You need access control. Otherwise, anyone on your network can send the display from their X Clients to your computer. In a larger or a production network environment, you will want to limit access to your local X Server.

xhost Control

The Linux way to control access to your local X Server is with the **xhost** command. This command controls access by computer. In Table 6-3, I illustrate a number of examples of how you can use the **xhost** command to secure your X Server. You can substitute IP addresses for specific computer hostnames.

TABLE 6-3

Limiting Access
to Your Local
X Server

Command	Description
xhost	Show current security settings
xhost +	Disable security; allow connections from any system
xhost -	Enable security
xhost +apps.xyz.com	Allow connection from apps.xyz.com
xhost -apps.xyz.com	Disable connections from apps.xyz.com

exam

Watch

These xhost commands won't work unless you're running them from a command line terminal inside the Linux GUI. Otherwise, you'll get an error message like "unable to open display."

The **xhost** command allows you to control X Server security on your computer. One more sophisticated alternative is the **xauth** command, which can use encryption to verify not just remote computers, but the permissions that you may grant to remote users. Access information for these other validation methods is stored in the hidden binary file `~/.Xauthority`.

Before you can send X Clients from a remote computer, you need to set the **DISPLAY** variable. You can do so with the **a** command such as **DISPLAY=localhost:0.0**. Then you can export that variable with the **export DISPLAY** command. Alternatively, you can combine the two commands with the following command:

```
# export DISPLAY=xserver:0
```

where *xserver* is the name of the remote computer where you want the X Client to be shown.

The second part of the **DISPLAY** line (0.0) requires further explanation. As described earlier, you can start multiple X Servers. When you are redirecting the output for an X Client, you must tell it not only which system to connect to but also which GUI display and which screen on that display to use. On most systems this will be *servername*, display 0, console 0 (written as *host:0.0*). The server portion of this entry can be a hostname, a fully qualified domain name, or an IP address. The following are all valid specifications for remote displays:

- `desk:0.0`
- `desk.xyz.com:0.0`
- `192.168.1.5:0`

To summarize, there are three steps required before you can run X Clients from remote computers:

- Disable the firewall on the local computer, the X Server, with a command such as **service iptables stop**.
- Authorize connections from the remote computer. For example, for a remote computer named charlie, you'd run the **xhost +charlie** command.
- Log into the remote computer and export the **DISPLAY** variable.

xauth Control with ssh

But there's an easier way. Using the Secure Shell (**ssh**) described in Chapter 11, you can log into the remote client computer. You can then start X Clients such as **xterm** or even a Red Hat GUI tool such as the Security Level Configuration tool described in Chapter 10 with the **redhat-config-securitylevel** command. The GUI application opens automatically on the local computer.

You still need to authorize access from the local computer with an appropriate **xhost** command, as I've just described.

EXERCISE 6-4

Starting a Display from a Remote Client

In this exercise, we'll look at the steps required to run a display from a remote client. This assumes a basic knowledge of the Secure Shell, and its associated command, **ssh**. For more information on running the **ssh** command, see Chapter 11. You'll need two computers running Linux. While RHEL 3 is not required on both computers, you will need the Secure Shell installed on both. The Secure Shell service, **sshd**, should be running on the remote computer. You'll also need the root password for both computers. Assume the local computer is named Enterprise, and the remote computer is named cosmicc (which is short for Cosmic Charlie, my favorite Grateful Dead teddy bear).

1. On the Enterprise computer, start the X Window. If it isn't already open, use the **startx** command.
2. When the Linux GUI is open, right-click on the desktop. In the pop-up menu that appears, click New Terminal.
3. Log into the remote computer using the Secure Shell. To log in as root, use the following command. Enter the root password on the remote computer

when prompted. If you're asked if you want to set up an encryption key, type **yes**. This should log you into the remote computer.

```
# ssh root@cosmicc
root@cosmicc's password:
```

4. Now you can start the GUI applications of your choice. Start with some easy X Clients, such as **xterm**, **xclock**, and **xeyes**. Where do you see these clients displayed? If you have the GUI open on the remote computer, you can walk over there and check the other computer for yourself. Close whatever X Clients you open.
 5. You should be able to run most of the Red Hat GUI utilities from the remote computer. Try some with commands such as **redhat-config-network**, **redhat-config-samba**, and **redhat-config-securitylevel**. You can now edit the configuration on the remote computer. Remember to close the GUI utilities that you open.
 6. If you run any remote GUI configuration utilities, check the results in the appropriate configuration file.
-

Troubleshooting

The X Window System is very robust and stable, but occasionally problems can arise. You can try several things when you troubleshoot X Window problems:

- Session managers create log files in your home directory such as `~/.xsession-errors`. Check these log files as well as `/var/log/messages` and `/var/log/XFree86.0.log` for error messages from your X Server.
- Check the **DISPLAY** environment variable to make sure it is set correctly. If you are running X Clients locally, they still use this variable. You can set it with this command:


```
export DISPLAY=localhost:0.0
```

or

```
export DISPLAY=:0.0
```
- Make sure `/usr/X11R6/bin` is in your `$PATH`.

- Check for underlying system problems or network problems that could be causing problems with the X Window System.
- Check the X Font Server for problems described earlier in this chapter.
- Even if your X Server is not responding or you can't read the display, don't forget that you can switch to a text console to gain access to the system.
- If you are troubleshooting X Server problems on a remote system, try starting an X Client from your workstation using the remote X Server's display. Note that you will need appropriate X security access to run the following command:

```
# xclock -display remotesys:0.0
```

EXERCISE 6-5

Troubleshooting DISPLAY Problems

In this exercise, we will see what happens if you unset the **DISPLAY** variable.

1. Log in to an X Window session. Run the **startx** command if required to do so.
 2. Bring up a terminal window. Right-click on the desktop, and click New Terminal from the menu that appears.
 3. Start the **xclock** X Client using the following command:

```
# xclock &
```
 4. Unset your **DISPLAY** variable:

```
# unset DISPLAY
```
 5. Start another **xclock**:

```
# xclock &
```
 6. You should get an error message saying that the X Client can't find your display. Reset the **DISPLAY** variable with the command:

```
# export DISPLAY=localhost:0.0
```
 7. Try starting the **xclock** X Client again.
-

CERTIFICATION OBJECTIVE 6.05

Desktops and Window Managers

Part of the Linux GUI is a special type of X Client known as a *window manager*. Earlier in this chapter in Exercise 6-1, you started the XFree86 X Server with the **X** command. It turned your display into a blank electronic canvas. This is the default desktop display for XFree86, which is an uninteresting textured gray background. The default mouse pointer for the X Window display is a graphic representation of an “X.”

Once XFree86 starts and you have this canvas on your screen, the X Server is ready to start serving X Clients. You will notice that the windows seem to be missing something. You could use the methods in the preceding section to set up various X Clients; for example, the **xclock -display localhost:0.0** command would lead to the screen shown in Figure 6-6.

Still, you don’t have any of the useful features that you’ve come to expect in a GUI, such as borders, title bars, menu bars, and minimize/maximize buttons. For this purpose, you need a window manager. A window manager is a special type of X Client that can only run with an X Server. The window manager controls how other X Clients appear on your display. This includes everything from placing title bars and drawing borders around the window for each X Client application you start,

FIGURE 6-6

An X Client on a plain X Server



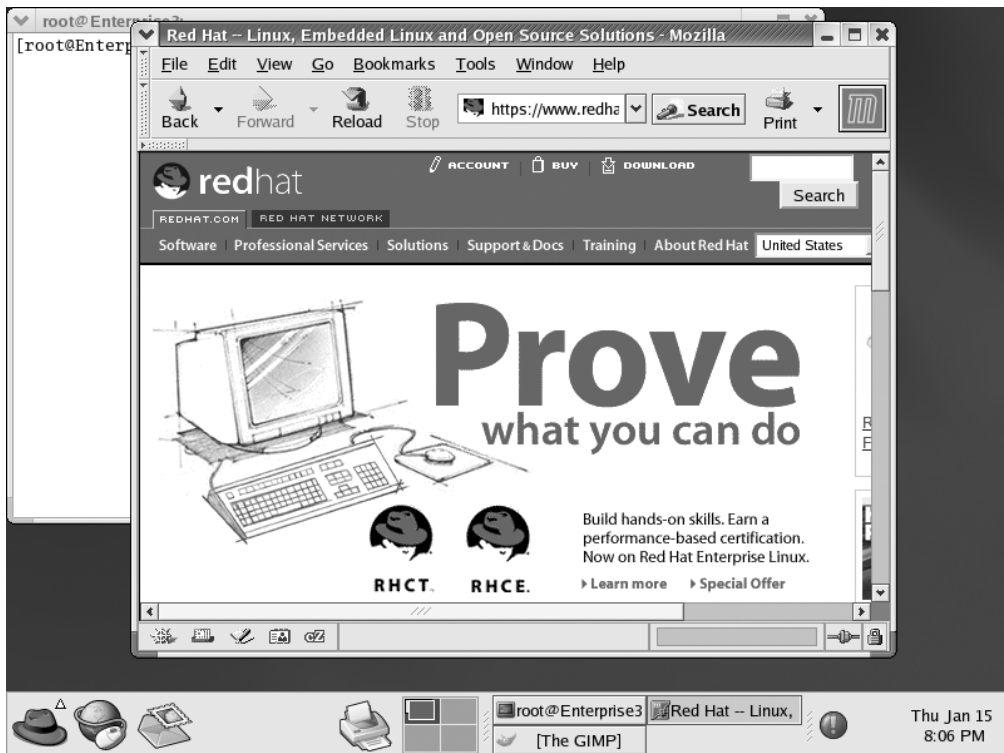
to determining the size of your desktop. In a nutshell, the window manager controls the look and feel of your GUI.

As is usually the case with all things Linux, you have multiple ways to accomplish the same task. RHEL 3 can be installed with several different window managers and desktops. The GNOME and KDE desktops include their own window managers. Your choice of window manager and desktop will drive the look, feel, and functionality of the Linux X Window System.

The GNOME and KDE Desktops

Two powerful virtual desktop environments that come with RHEL 3 are the GNOME (GNU Network Object Model Environment) desktop environment and KDE (the K Desktop Environment). The GNOME desktop, shown in Figure 6-7, is the default desktop for RHEL 3 and is the desktop you first see after installing the X Window System. The

FIGURE 6-7 The GNOME desktop



KDE desktop, shown in Figure 6-8, is the main alternate desktop system. KDE is the default for several other Linux distributions.

GNOME Features

The GNOME desktop includes support for the Common Object Request Broker Architecture (CORBA), which allows GNOME software components written in any language and running on different systems to work together. In addition, the GNOME developer community is also working on an architecture similar to Microsoft's Object Linking and Embedding (OLE) architecture that will allow one GNOME application to call and control another GNOME application.

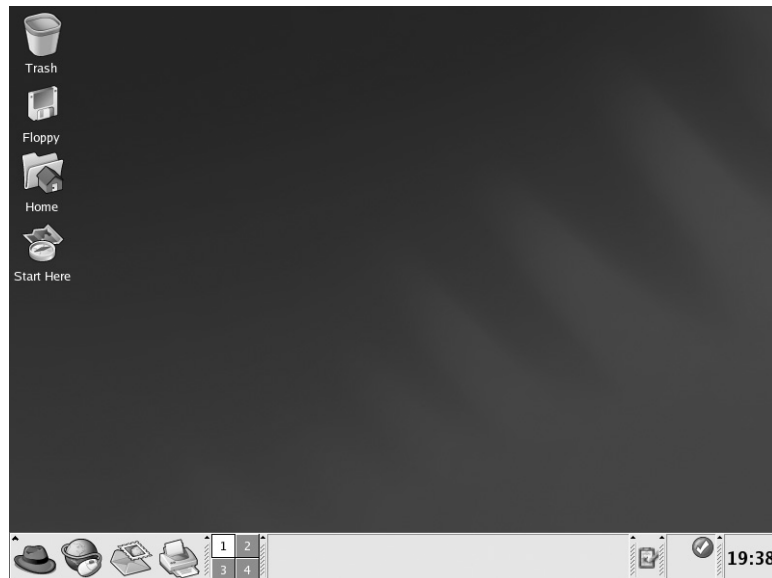
One very nice feature of GNOME-compliant applications is that they are *session aware*; that is, when you quit an application, the application remembers the location in the document where you were last working and will reposition your cursor to that point when you restart the application.

Using GNOME

Many of the features of the GNOME interface will be familiar to you from other desktop environments. On the left side of the screen are icons representing files and applications that you can be open by double-clicking them with the mouse. The GNOME desktop environment also provides you with several virtual desktops. Next to the application

FIGURE 6-8

The KDE desktop



buttons on the right side of the panel is a pager you can use to move from one area of the desktop to another.

One of the key features of GNOME is the *panel*, which you can see at the bottom of the screen in Figure 6-7. The panel is the control center for most of your activities while you use GNOME. The button at the far left of the panel with the imprint of a red hat is the Main Menu button. Click this button, and you will see a list of menus and submenus which start applications. You can also launch configured applications from the panel by clicking the appropriate icon. The default buttons include Mozilla (the Web browser) and Evolution (the mail and schedule manager).

GNOME includes a number of applications, including graphics tools and an office suite, GNOME Office. As the default Red Hat desktop is GNOME, the remainder of this book will be based on this desktop environment. Nevertheless, the Red Hat exam requirements do not specify a preferred desktop; there should be no problems using KDE or the command line console to do everything that is required for the exam. You may be asked to configure either desktop on the Red Hat exams.

If you configure the default GNOME desktop for your users, you may want to configure GNOME in a special way. Normally, GNOME opens with a number of icons and possibly default applications such as nautilus. You can add more default applications such as a new terminal window or the **xcalc** calculator with the Sessions tool, which you can access via the Main Menu | Preferences | More Preferences | Sessions command.

KDE Features

The KDE desktop is built on the Qt C++ cross-platform GUI toolkit. This is another versatile way to create GUI applications for Linux.

Many of the features of KDE should also be familiar to you from other desktop environments. In fact, you can configure KDE to a look and feel that is quite similar to Windows 9x. As shown in Figure 6-8, it includes a Main Menu button, represented by the Red Hat in the lower-left corner of the desktop. Like GNOME, it includes pagers and buttons representing the open programs on the desktop.

Default Desktop

Once you've configured X Window, it's easy to start a Linux GUI. If it isn't already configured to start automatically, just run the **startx** command. This command, in the `/usr/X11R6/bin` directory, calls configuration files from your home directory. If these files don't exist, they are taken from the default directory for GUI configuration, `/etc/X11`.

To manage the default desktop, use the **switchdesk** command. For example, the following commands set the default desktop to KDE and GNOME, respectively:

```
# switchdesk KDE
# switchdesk GNOME
```

The **switchdesk** program creates two hidden files in your home directory, `~/.Xclients` and `~/.Xclients-default`, that are used to start your alternate desktop. You don't need to use **switchdesk**; once you have an `~/.Xclients-default` file, you can edit it directly. It is a simple file; if your default desktop is KDE, this file has one line:

```
exec startkde
```

If your default desktop is GNOME, this file has a different line:

```
exec gnome-session
```

Alternatively, you can use **switchdesk** to set up twm, known as Tom's Window Manager. The version of twm in RHEL 3 includes just the same textured gray screen that you get with the **X** command. The `~/.Xclients-default` file would include the following line:

```
exec /usr/X11R6/bin/twm
```

If you have other desktops or window managers installed, you can use those instead. When run at the command line, the **switchdesk** command can also let you set FVWM, Enlightenment, or WindowMaker as the default window manager. You'll see the new default the next time you run the **startx** command from a console command line interface.

EXERCISE 6-6

Desktops

Let's use the **switchdesk** command to explore the various desktops available in RHEL 3. It's also known as the Desktop Switching Tool.

1. The easiest way to make this work is if you've disabled any automatic login through the GUI. To do so, open a terminal window, and run the **init 3** command.
2. Now enter the GUI with the **startx** command.

3. Open a terminal window in a GUI and run Main Menu | System Settings | More System Settings | Desktop Switching Tool. You can also open it from a GUI command line with the **switchdesk** command.
4. Your current desktop (probably GNOME) is selected. Try one of the other desktops (such as KDE, the other popular desktop for Linux).
5. Log out of your current session with the GNOME Main Menu | Log Out command.
6. Log back in again. This time you should see the KDE desktop.
7. Repeat step 3.
8. In the Desktop Switching Tool, try switching to twm.
9. Log out of KDE, and run the **startx** command again. You should see that twm is a much more basic window manager. You'll need to log out again for your changes to take effect.
10. To exit from twm, left-click on the desktop. In the pop-up menu that appears, click Exit.
11. Run **switchdesk** from the command line. Select your favorite desktop. For example, if you want to make GNOME your default desktop again, run the **switchdesk gnome** command.



Commercial Unix vendors like Sun Microsystems and Hewlett-Packard have officially endorsed GNOME and are working to make their proprietary versions of Unix (Solaris and HP-UX) fully GNOME-compliant. If you get comfortable with GNOME, you will find moving to Solaris or HP-UX much easier. Alternatively, several Linux distributions use KDE as their default desktop.

CERTIFICATION OBJECTIVE 6.06

The X Start Process

You can configure the X Window interface to start automatically when your system boots, or you can choose to start the X Window System manually. You can set up this option in the `/etc/inittab` configuration file. If you select a text login, you'll have to

start the Linux GUI with the **startx** command. If you select a graphical login, you can also select the graphical login manager of your choice.

Text Login Mode

To access a Linux system, you need to log in. In other words, you identify yourself to the system with a username and a password. But this requires a login program. When you log into Linux at a regular command line interface, the **mingetty** command calls up a login program that prompts you for your username and password. Six **mingetty** terminals are configured through `/etc/inittab`.

There are six active runlevels in Linux. The two of concern here are runlevels 3 and 5. If you start Linux in runlevel 3, you get a text login. If you start Linux in runlevel 5, you get a graphical login screen.

It's easy to change the default boot mode from the command line to X Window. Just edit the `/etc/inittab` file. Go to the line with the `initdefault` variable. To make Linux start the X Window System automatically when Linux starts, use your favorite text editor and change the line in `/etc/inittab` that reads

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

to

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

Of course, you can reverse this process. This change does not take effect until you reboot. If you are running as the root user, you can also use the **init** command to switch between runlevel 5 and runlevel 3. Running this command:

```
# init 3
```

switches your display from X Window back to text terminal mode, whereas:

```
# init 5
```

switches you from text terminal mode to X Window. You can find more information on different runlevels and `/etc/inittab` in Chapter 4.

Display Managers: xdm, gdm, kdm

When you configure Linux to start from runlevel 5, you can log in through a special X Client, the *display manager*. The display manager is a fairly simple program; all it does is display a dialog box on the screen asking for your username and password. You can use any of three major display managers. The default display manager is the GNOME display manager, or gdm. To change your display manager, edit the **prefdm** shell script in the /etc/X11 directory, as shown in Figure 6-9.

Take the line that contains `preferred=` and add either `kdm`, `gdm`, or `xdm` for the KDE Display Manager, GNOME Display Manager, or X Display Manager, respectively. In the default RHEL 3 version of this file, it's on line 25.

How you start the X Window System affects its behavior. Assume you start X Window through one of the graphical display managers. When you exit an X Window session, you're returned to the same display manager. To continue, you would need to log in again.

The default behavior for the `gdm` window manager is to launch a GNOME session for you. The `kdm` window manager by default launches a KDE session. However, you can start GNOME or KDE from either login manager.

Both the GNOME and KDE environments use their own startup files. Through the use of option buttons, both the GNOME and KDE display managers allow you to choose the desktop environment you wish to start. From the GNOME display manager, click the Session menu. This starts a window where you can select from several desktop options, as shown in Figure 6-10.

FIGURE 6-9

Set your preferred display manager in /etc/X11/prefdm.

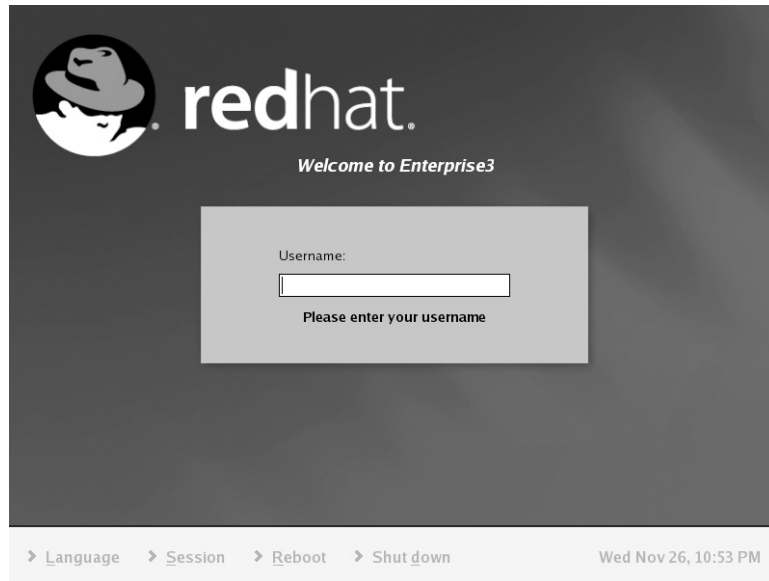
```

5
6 # shut down any graphical boot that might exist
7 if [ -x /usr/bin/rhgb-client ]; then
8     /usr/bin/rhgb-client -quit
9 fi
10
11 # We need to source this so that the login screens get translated
12 [ -f /etc/profile.d/lang.sh ] && . /etc/profile.d/lang.sh
13
14 # Run preferred X display manager
15
16 # Try autologin first, if wanted...
17 if [ -f /etc/sysconfig/autologin -a -x /usr/sbin/autologin ]; then
18     if /usr/sbin/autologin; then
19         exit 0
20     fi
21     # If autologin fails (bad permissions, etc.), we get here -
22     # time to start a real display manager.
23 fi
24
25 preferred=gdm
26 if [ -f /etc/sysconfig/desktop ]; then
27     . /etc/sysconfig/desktop
28     _if [ "$DISPLAYMANAGER" = GNOME ]; then

```

FIGURE 6-10

The GNOME display manager, gdm



From the KDE display manager, click the drop-down text box adjacent to Session Type, as shown in Figure 6-11. You can then select the desired desktop.

You can even set up the generic xdm window manager, which gives you the login screen shown in Figure 6-12.

FIGURE 6-11

The KDE display manager, kdm

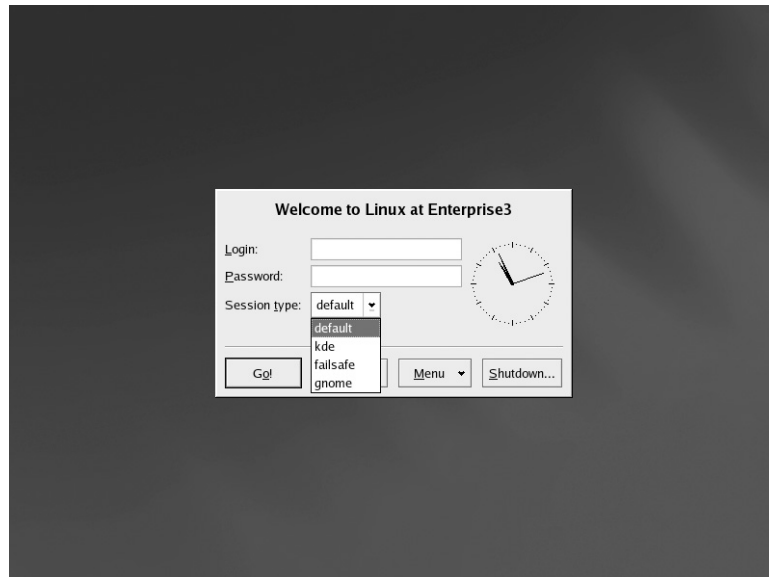


FIGURE 6-12

The xdm
login screen



exam

Watch

If the `startx` command doesn't successfully start the GUI from the command line in runlevel 3, check for a working `xfs` server in that runlevel. Unless you're already in runlevel 5, it doesn't matter if you've set `xfs` to start in runlevel 5.

If you log in using a display manager, X Window starts somewhat differently when compared to using `startx`. When you run `startx`, the X Window runs as a child process of your text-based login shell. Even though the X Window System is running, Linux is still at runlevel 3. After you exit the X Window System, you still have to log out of this shell to terminate your login session.

Analyzing startx

You've already seen that you can start the Linux GUI with the `startx` command. This starts the X Server and switches your display into graphics mode. It also runs through a series of configuration files which prompt Linux to start your selected desktop and other desired applications. In this section, we'll look at how `startx` actually works.

on the job

The `startx` command is actually a shell script that serves as a front end to the `xinit` command. The default location for both the `startx` and `xinit` is `/usr/X11R6/bin`.

There are a number of configuration files which can help you customize the behavior of your X Window session. These are hidden files that you can install into user's home directories. If **startx** doesn't find a particular configuration file in your home directory, it uses a default version of the same file. Let us start our analysis with two of the first lines in the **startx** script:

```
userclientrc=$HOME/.xinitrc
sysclientrc=/etc/X11/xinit/xinitrc
```

When you start the X Window System with the **startx** command, the **xinit** program looks for a file to run named **.xinitrc** in your home directory. If the **startx** command cannot find **\$HOME/.xinitrc**, it will read the **/etc/X11/xinit/xinitrc** configuration script. This file, in turn, runs either the file **\$HOME/.Xclients** or, if that file doesn't exist, **/etc/X11/xinit/Xclients**.

A typical **~/.xinitrc** file includes commands that start various X Clients. (Remember, the tilde (**~**) represents the home directory.)



As described earlier in this chapter, you can also configure X Clients to start when you start the GNOME desktop through the Sessions utility. Settings are stored in each user's home directory, in the **~/.gnome2/session-manual** file.

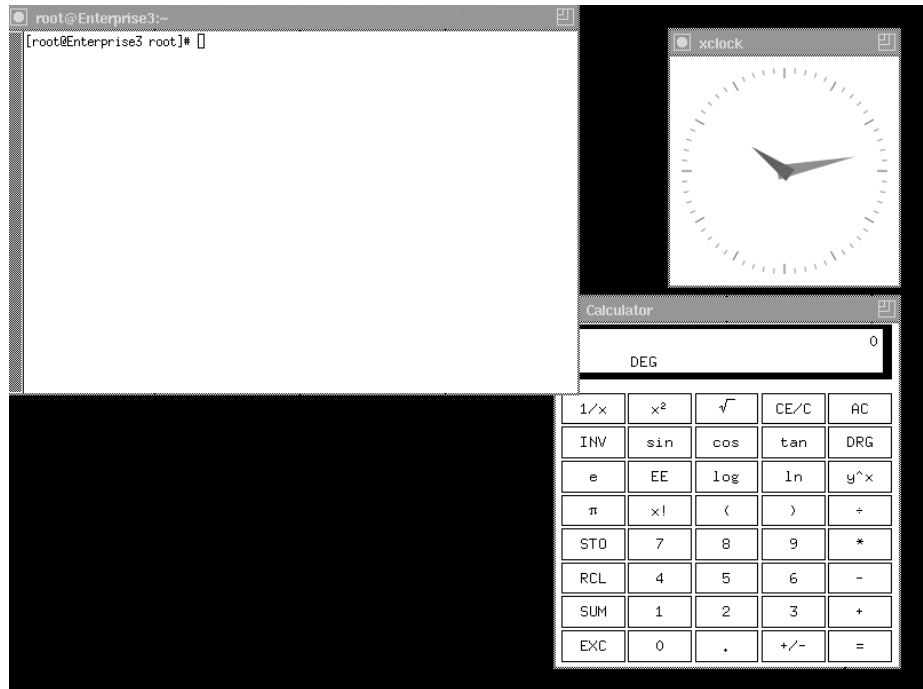
Let me illustrate a simple **~/.xinitrc** file:

```
#!/bin/bash
xterm &
xclock -geometry 200x200-20+20 &
xcalc -geometry 300x300-20-20 &
exec twm
```

The first line sets **bash** as the shell for this script. The next line starts an **xterm** terminal client. The following line starts **xclock**, with a specific size and location. The line after that brings up an X Window calculator, also with a specific size and location. Notice that the first three command lines end with an ampersand (**&**). This is important; it tells the shell to run each command line and return control to the shell for this script. The final line uses the **exec** command to start the **twm** window manager. I illustrate the result in Figure 6-13; this supersedes any default desktop and X Clients which you may have otherwise configured.

FIGURE 6-13

A GUI as configured through `~/xinitrc`



You can start other desktops through this file. You can substitute one of the following commands for `exec twm`:

```
exec gnome-session
exec startkde
```

You can create an `~/xinitrc` file with a text editor of your choice. Once you've saved the file, `startx` executes the commands in this file, with the results shown in Figure 6-13.

exam

Watch

There is more than one way to do many things in Linux, including the setup of default X Clients. Knowing these

options can help you troubleshoot problems during the Red Hat exams.

EXERCISE 6-7

Customizing the startx Process

In this exercise, we will configure the X Window System to use a customized `.xinitrc` file.

1. If the X Window System is running, change to a text console by pressing CTRL-ALT-F1.
2. If you see a login prompt, log in at the text console as root. Otherwise, press CTRL-C to stop the current X Window.
3. If you logged in at the text console, stop the current X Window Server with the following command:

```
# init 3
```

4. Make sure you're in your home directory. Use your favorite text editor to create a `.xinitrc` file. If you're not sure what to do, use the sample file shown in the previous section.
5. Start the X Window System with the following command:

```
startx
```

Your X Window session should automatically start the applications in your `.xinitrc` file.

6. Exit from the twm desktop. Left-click on the desktop to bring up the twm main menu and click Exit.
7. If you want to restore your original desktop settings, to allow you to start GNOME or KDE with the **startx** command, remove the `.xinitrc` file from your home directory.

Now that you have seen how X Window Clients and the X Window Server work together, refer to the following Scenario & Solution for some common situations you may encounter, along with their solutions.

SCENARIO & SOLUTION	
I'm having problems getting XFree86 to run on my hardware.	Check the Red Hat hardware support site. Run redhat-config-xfree86 .
I want to use a different desktop environment.	Use the switchdesk command to change your desktop environment.
I want to stop the X Window System without rebooting Linux. When I try logging out, all I see is a graphical login manager.	Use the init 3 command to change the system runlevel to runlevel 3.
I'm having problems starting an X Client.	Check that the DISPLAY variable is properly set with the following command: # export DISPLAY=localhost:0.0 Check for underlying network problems. Check X security problems.
My X Window display is acting strangely and I can't log in.	Switch to a virtual console and log in. Check the error logs. See if your X Font Server (xfs) is running. If it isn't, check for errors in <code>~/.xsession-errors</code> and <code>/var/log/messages</code> . Make sure your <code>/tmp</code> and <code>/home</code> directory partitions are not full. As a last resort, rerun redhat-config-xfree86 .

CERTIFICATION OBJECTIVE 6.07

Basic Graphical Applications

At the last revision to the Red Hat Exam Prep guide, Red Hat added a couple of prerequisite skills: configuring e-mail clients and working with Web browsers. Therefore, it is reasonable to expect to have to install browsers, configure e-mail clients, and similar applications on a desktop computer during the Red Hat exams.

Overview

There is a wide variety of graphical applications available for RHEL 3. Most are accessible through the Red Hat Main Menu button in the lower-left corner of the GNOME and KDE desktop environments.

Red Hat has modified its version of the GNOME and KDE desktops with the “Bluecurve” theme. It has partially standardized the look and feel of both desktops. The Main Menu button in the lower-left corner of these desktops has a different look in other Linux distributions. In GNOME, the Main Menu button is a foot; in KDE, it’s a big “K.”

Red Hat has modified the Main Menu options that you see in both desktops. The results are similar, though not identical. Many of the menu options are the same in the Red Hat version of the GNOME and KDE desktops.

Browsers

The default Web browser in the RHEL 3 GNOME and KDE desktops is Mozilla. You can start it in either desktop environment by clicking on the icon with the mouse and globe adjacent to the Main Menu button.

One of the prerequisites in the Red Hat Exam Prep guide is that you need to know how to use the Mozilla Web browser to access regular and secure sites on the Internet. While this is an elementary skill, you don’t have to know how to use a Web browser to administer a Linux operating system.

There are a number of other Web browsers available in RHEL 3, including KDE’s Konqueror. If you don’t know how to use a Web browser in Linux, I suggest that you refer to a basic guide to Linux, such *Red Hat Linux Administration: A Beginner’s Guide* by Michael Turner and Steve Shah (McGraw-Hill/Osborne, 2003).

Configuring E-mail Clients

There are graphical and text e-mail clients available on RHEL 3. The default graphical e-mail client in RHEL 3 is Evolution, which is functionally similar to Microsoft’s Outlook. You can start this program by clicking on the icon with an envelope and stamp. By default, it’s next to the Mozilla icon in the taskbar.

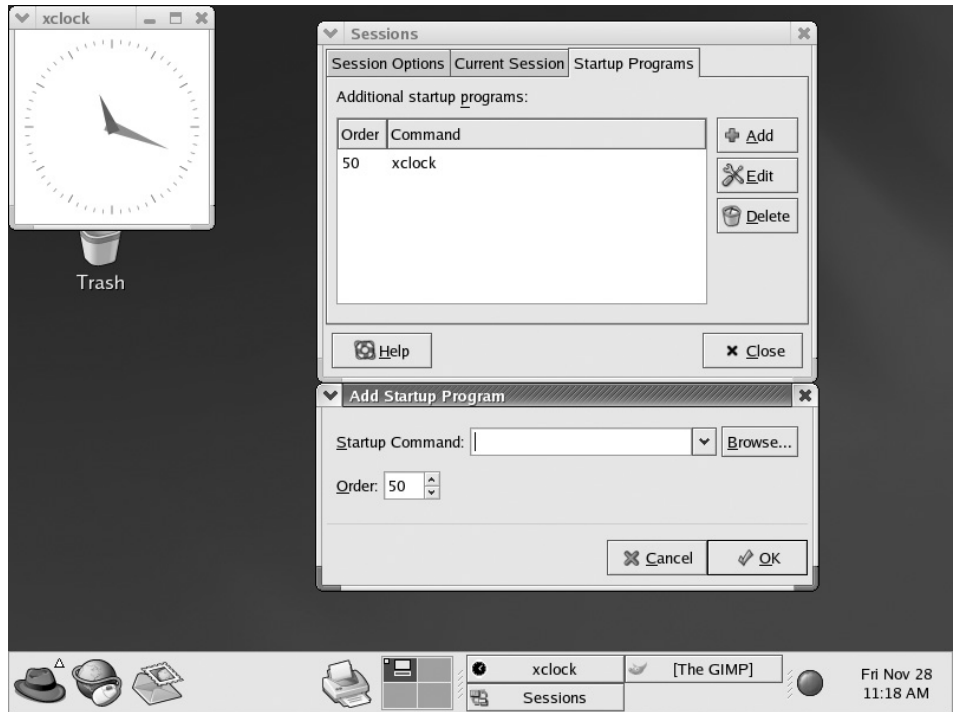
When you first start Evolution, it prompts you for your e-mail information, including your name, e-mail address, and mail server type. If you’re configuring a POP (Post Office Protocol) or IMAP (Internet Message Access Protocol), you’re also prompted for incoming and outgoing mail server addresses, as well as login information as required for each server. Finally, you’re asked to set your time zone.

Configuring Default Clients

It’s easy to add default clients for a user. If you’re configuring a GNOME system, you can do so with the Session utility. As described earlier, you can open it by clicking

FIGURE 6-14

Configuring
default GNOME
clients



Main Menu | Preferences | More Preferences | Sessions. In the Sessions window, click the Startup Programs tab. As shown in Figure 6-14, the **xclock** program is already in the list. If you click Add, that starts the Add Startup Program window, where you can add the additional startup programs of your choice.

Alternatively, you can use the `.xinitrc` file described earlier. As the Sessions utility does not work on KDE, that is the only option if you're configuring X Clients to start when you run the KDE desktop.

CERTIFICATION SUMMARY

The X Window System provides a state-of-the-art graphical user interface and offers features not found in other GUI environments. Although the X Window System can be complicated, you should be able to configure it during the RHEL 3 installation process. Alternatively, the Red Hat Display Settings tool simplifies the setup or reconfiguration process.

One of the key parts of the X Window system is the X Font Server. If it isn't running, or the `/tmp` or `/home` directory partitions are full, you can't run the Linux GUI.

The X Window system works as client and server. X Clients and X Servers can be located on different computers on a network. You can set it up with appropriate **xhost**, **DISPLAY**, and firewall commands. If you prefer, you don't have to worry about your firewall, as you can bypass it through the Secure Shell.

The look and feel of the X Window interface is determined by your choice of desktop. RHEL 3 comes with several desktop environments, including GNOME, KDE, and twm. The GNOME desktop interface is the default for RHEL 3.

You can customize the GUI start process in a number of ways. You can configure a default login manager. With an appropriate `.xinitrc` file, you can set up X Clients to start when you run the **startx** command. Plus, if you create an `.xinitrc` file and then run the **switchdesk** command, Linux modifies the `.xinitrc` file. Alternatively, you can run the Sessions utility to configure X Clients in the GNOME desktop.



TWO-MINUTE DRILL

Here are some of the key points from the certification objectives in Chapter 6.

X Server

- ☐ The X Server is the software that manages the graphics display on the local computer, which includes your monitor/graphics adapter, keyboard, and mouse.
- ☐ X Servers work with local and remote X Clients to keep your display up to date. X Servers send inputs from the keyboard and mouse to X Clients. Then X Clients send instructions to the X Server to update the graphics on the monitor.
- ☐ You can configure the X Server during the RHEL 3 installation process. You can also configure or modify the configuration using the Red Hat Display Settings tool.
- ☐ The X Server configuration is stored in the `/etc/X11/XF86Config` file.

X Font Server

- ☐ If your X Font Server is not running, you won't be able to start the Linux GUI. There are a number of problems which can keep the X Font Server from running.

X Clients

- ☐ RHEL 3 supports a great number of X Client programs. These include terminal emulators, desktop accessories (such as the File Manager), and more.
- ☐ You can send the display of an X Client program to a remote system by using the `-display` option.

Remote Display of X Applications

- ☐ By default, X Clients sends display output to the local computer.
- ☐ You can send display output to other computers with the `-display` command line option.

- ☐ The local system must be willing to accept your remote X Client. You can run the **xhosts +remotepc** command to accept remote X Clients from the computer named remotepc.
- ☐ You can log into the remote client using the Secure Shell.
- ☐ You can set the **DISPLAY** environment variable on the X Client computer to direct all display output to a remote X Server.

Desktops and Window Managers

- ☐ The X Window System just gives you a blank electronic canvas. The look and feel of a GUI is provided by the window manager and desktop.
- ☐ The two main desktop environments are GNOME and KDE.
- ☐ You can use **switchdesk** from a terminal window or the command line interface console to select your default desktop.

The X Start Process

- ☐ You can configure the X Window interface to start automatically, or you can start it manually from the command line console with the **startx** command.
- ☐ You can customize the **startx** process by configuring X Clients in the `~/.xinitrc` file.
- ☐ You can set up graphical logins with the X display manager, the KDE display manager, or the GNOME display manager.

Basic Graphical Applications

- ☐ The default RHEL 3 Web browser is Mozilla. The default graphical e-mail client is Evolution.

SELF TEST

The following questions will help you measure your understanding of the material presented in this chapter. Read all the choices carefully, as there may be more than one correct answer. Don't focus exclusively on these questions. There are no longer any multiple choice questions on the Red Hat exams. These questions exclusively test your understanding of the chapter. While the topics in this chapter are "prerequisites," it is okay if you have another way of performing a task. Getting results, not memorizing trivia, is what counts on the Red Hat exams.

X Server

1. Which of the following is true about the X Window System?
 - A. The X Server runs on your workstation; X Clients run on your workstation or on other computers on the network.
 - B. If an X Server is running on your workstation, then X Client applications running on your computer cannot send their output to an X Server running on another system on the network.
 - C. An X Client application gets its input from the keyboard and mouse attached to the same X Server where the client is sending its output.
 - D. Aside from the steps necessary to start the remote application, there is no difference between running an X Client locally and running one remotely.
2. You performed a new install of RHEL 3 and chose not to configure the X Window System. Your system is now active, and you can log in to a command prompt. How do you go about configuring the X Window System?
 - A. Use the vi editor and modify the `/etc/X11/XF86Config` file.
 - B. Run the `redhat-config-xfree86` command.
 - C. Run the `xf86config` command.
 - D. Reinstall RHEL 3.

X Font Server

3. You're having trouble starting the GNOME desktop environment, and you suspect the problem is with the X Font Server. Which of the following commands will tell you if it's running properly?
 - A. `service xfont status`
 - B. `service xfs status`
 - C. `service xfonts status`
 - D. `service gnome status`

X Clients

4. You want to start an **xterm** X Client that is 80 columns wide by 30 lines high from the command line and position it in the upper-right corner of your display when it starts. What command would you use?
- A. `xterm -geometry +0-0 -font 80x30`
 - B. `xterm -geometry 80x30-0+0`
 - C. `xterm -geometry 80+30+0+0`
 - D. `xterm -display 80x30-0+0`

Remote Display of X Applications

5. You log in to a remote system with the Secure Shell with the intention of starting several remote X Clients to send their output to your local display (`admin1.xyz.com`). What can you do to make this easier?
- A. Use the `-display admin1.xyz.com` option.
 - B. Create a **DISPLAY** variable with this command:
`DISPLAY=admin1.xyz.com:0.0`
 - C. Create a **DISPLAY** variable with this command:
`export DISPLAY=admin1.xyz.com`
 - D. Create a **DISPLAY** variable with this command:
`export DISPLAY=admin1.xyz.com:0.0`
6. You're trying to gain Secure Shell access to another computer on your LAN so you can run X Clients from the remote system. That computer is refusing access. Neither computer has a firewall. What can you do on that remote computer so it will accept your **ssh** command inputs?
- A. Restart the Secure Shell with the `/etc/rc.d/init.d/sshd restart` command.
 - B. Enable the Secure Shell in the `/etc/ssh.conf` configuration file
 - C. Disable firewalls on the network.
 - D. Restart xinetd with the `/etc/rc.d/inet.d/xinetd restart` command.

Desktops and Window Managers

7. When you installed RHEL 3, you configured the X Window System but chose to log on in text mode. You have been starting the X Window System with the **startx** command and everything is working fine. In the future, you want to boot Linux directly into the GUI. How would you make this happen?

- A. Edit `/etc/inittab` and replace the `id:3:initdefault` command with `id:5:initdefault`.
 - B. Run `redhat-config-xfree86`.
 - C. Change the `/etc/X11/prefdm` link.
 - D. Execute the `runlevel 5` command.
8. Your system is using the `xdm` display manager. You want to use the GNOME display manager (`gdm`). How can you do this?
- A. Change the `/etc/X11/prefdm` shell script and define a value for the variable named **preferred**.
 - B. Run `redhat-config-xfree86`.
 - C. Edit `/etc/X11/XF86Config`.
 - D. Use the GNOME Control Panel to change the display manager.

The X Start Process

9. You would like to automatically start up the `xclock` application whenever you start the GNOME desktop with the `startx` command. Which of the following methods can you use?
- A. Create or edit the file `~/.Xdefault` and add this command:

```
xclock &
```
 - B. Use the Sessions utility and add the `xclock` command to the Startup tab.
 - C. Edit `/etc/X11/XF86Config` and add this command:

```
xclock -geometry 200x200-0+0 &
```
 - D. Create or edit the file `.xinitrc` in your home directory and add the following command. Make sure the last command in this file is **exec twm**.

```
xclock &
```

Basic Graphical Applications

10. Which of the following is true about the default GNOME and KDE desktop environments in RHEL 3?
- A. The basic Red Hat utilities can only be found through the GNOME desktop.
 - B. The basic Red Hat utilities can only be found through the KDE desktop.
 - C. You can configure GNOME and KDE with the Red Hat Display Settings tool.
 - D. You can use the same menus in GNOME and KDE to start many Red Hat configuration utilities.

LAB QUESTIONS

Lab 1

You want to upgrade the video card in your Linux system. Your old video card is slow and doesn't have enough display memory to provide you with the resolution and color depth you require. You have obtained a new ATI 32MB Radeon card (I'm using this product for example purposes only). What steps might you follow to replace your old card with your new card?

Lab 2

You want to see what happens when there are problems starting the Linux GUI. With RHEL 3, the XFree86 Version 4.x server is configured by default. The configuration of the X Server is stored in the `/etc/X11/XF86Config` configuration file. Before Linux starts the X Server, it reads this file. To do so, you'll want to back up your current `/etc/X11/XF86Config` file, delete it, and then reboot your computer into runlevel 5. You can restore it after the lab is complete.

Lab 3

For this lab, you'll need two Linux computers connected over a network. You'll need a shared NFS directory from the local computer. You can use the same directory that you may have used in Chapter 2 to share the RHEL 3 installation files. Start a Secure Shell connection between the two computers. Start the GUI on the local computer, and use the Secure Shell to log in remotely to the other computer. Finally, set and export an appropriate **DISPLAY** variable. You can then see what happens when you start X Clients from the remote computer.

Once you do, run the Red Hat root password program from the remote computer. Make changes to the password. When you log out and try to log back into the remote computer, you should be able to confirm that the root password on the remote computer has changed.

Lab 4

In this lab, you'll set up a GUI workstation. It'll start with the `xdm` login manager, and automatically start GNOME, open the Mozilla Web browser, and start a `gnome-terminal` session when you boot this Linux computer.

SELF TEST ANSWERS

X Server

1. ☒ **A, C, and D.** **A** is correct because the X Server runs on the local computer with the graphics hardware, while X Clients may run locally or remotely. **C** is correct because the X Server sends keyboard and mouse inputs to local or remote X Clients. **D** is also correct because remote X Clients interact with the X Server in the same way as a local X Client.
☒ **B** is incorrect because there simply is no such restriction.
2. ☒ **B.** You should use the Display Settings tool any time you need to make changes to your X Window configuration.
☒ **A** is possibly correct assuming you understand enough about the XF86Config file to make the necessary changes. From experience, I would suggest you only edit this file if you have an (almost) working configuration you only wish to fine-tune. **C** refers to an X configuration tool that is not available in RHEL 3. **D** is extreme. The only justification for this might be if you performed a custom install and perhaps neglected to include all the components of the X Window System. Even if this is the case, you can still install the needed RPMs from the command line.

X Font Server

3. ☒ **B.** The X Font Server service is **xfs**. You can check the status of this service with the **service xfs status** command.
☒ **A, C, and D** are incorrect because these are not valid services for RHEL 3.

X Clients

4. ☒ **B.** The correct option is **-geometry 80x30-0+0**. Since you are creating a terminal window, the size specification 80x30 refers to the number of columns and lines. The offset specification -0+0 specifies that the right border of the **xterm** window should be offset 0 pixels from the right edge of the display and that the top border of the **xterm** window should be offset 0 pixels from the top of the display.
☒ **A** is incorrect because the **-font** option is used to specify the default font for the window, not the window location. **C** is incorrect because the window size is specified by the 80x30 format. **D** is incorrect because **-display** is the wrong switch for sizing and positioning the **xterm** window.

Remote Display of X Applications

5. ☒ **D.** X Client applications use the **DISPLAY** environment variable to determine where to send their output. It must be an environment variable, so you must use the **export** command to create it.
☒ **A** suggests using the **-display** option to every remote X Client you wish to start. While this is not the easiest way to do it, it could work. To make the option effective, though, you would have to write it as **-display admin1.xyz.com:0.0**. **B** is incorrect because it makes a variable called **DISPLAY** with the correct value, but it neglects to export the variable. **C** is not quite correct. While you do export **DISPLAY**, this answer does not identify the X graphics display and the screen (0.0).
6. ☒ **A.** The Secure Shell is normally enabled by default in RHEL 3. However, there are a number of reasons why it may not be on; this command restarts it.
☒ **B** is incorrect because there is no **/etc/ssh.conf** configuration file. **C** is incorrect because standard firewalls should not block the Secure Shell. **D** is incorrect because the Secure Shell in RHEL 3 is not a xinetd “Super Server” process. For more information on the Secure Shell, see Chapter 11.

Desktops and Window Managers

7. ☒ **A.** The easiest way to change the default runlevel for your Linux system is to change the **initdefault** setting in **/etc/inittab** to runlevel 5. You could then enter the **init 5** command to change to runlevel 5 immediately.
☒ **B** is incorrect because the Red Hat Display Settings tool cannot change **/etc/inittab**. **C** is incorrect because this just configures the preferred display manager. **D** is incorrect because the **runlevel** command is used to report on the previous and current runlevels, not to set the new default runlevel.
8. ☒ **A.** The file **/etc/X11/prefdm** is a script that starts your preferred display manager. By default, the variable named **preferred** is not set, so if your default runlevel is 5, Red Hat launches **gdm** (the GNOME display manager). To force a different display manager, set the **preferred** variable to either **kdm** (for the KDE display manager) or **xdm** (for the X display manager).
☒ **B** is incorrect because the Red Hat Display Settings tool does not set the preferred display manager. **C** is incorrect because the preferred display manager is not set in **XF86Config**. **D** is incorrect because there is no such option in the GNOME Control Panel.

The X Start Process

9. ☒ **B.** You can configure X Clients to start automatically in GNOME with the Sessions utility.
☒ **A** is incorrect because `.Xdefault` is not used to start X Clients when you start the GUI. **C** is incorrect because the `XF86Config` file is used to configure the X Server to the correct hardware settings, not launch a user's favorite applications. **D** is incorrect because the `exec twm` command starts Tom's Window Manager, not the GNOME desktop.

Basic Graphical Applications

10. ☒ **D.** Red Hat has configured both GNOME and KDE so you can find many Red Hat configuration utilities through the same GUI menus.
☒ **A** and **B** are incorrect as you can find basic Red Hat configuration utilities, using the Main Menu button, in both the GNOME and KDE desktops. **C** is not correct as you can only configure the X Server with the Red Hat Display Settings tool.

LAB ANSWERS

Lab I

- Before you stop Linux on your computer, you should configure it so it no longer attempts to start the X Server when Linux boots. This is controlled by the `initdefault` line in the `/etc/inittab` file. Edit this file and change the second field from 5 (Multi-User With X Support) to 3 (Multi-User With No X Support). You could use `vi`, `pico`, or any other suitable text editor to do this job.
- Perform an orderly shutdown on your system at a safe time. Use the `shutdown -h now` command.
- Now that the system is off, replace your video card.
- Start your computer and boot into RHEL 3. During the boot process, the Red Hat **kudzu** command automatically probes for new hardware. If this probe finds your new video card, you can configure it when prompted.
- If **kudzu** fails to find your new hardware, you should log in as root and run the Red Hat Display Settings tool.

6. The Red Hat Display Settings tool should correctly identify your new hardware. You should select the correct amount of display memory (32MB) and the graphics resolutions and color depths you desire. If it does not, you can configure it manually using the available settings.
7. Test the result. Run the **startx** or **init 5** command to start the Linux GUI.

Lab 2

1. Back up `/etc/X11/XF86Config` to a safe location such as your home directory.
2. As the root user, delete or rename the `/etc/X11/XF86Config` file.
3. Open `/etc/inittab` in your favorite text editor. Look at the line with `initdefault`. Change the number right before this variable from a 3 to a 5 if required.
4. When you reboot your computer, observe what happens when Linux tries to find the default login display manager.
5. Restore your original settings.

If you are interested in more experiments, try deactivating the X Font Server. Run the **chkconfig xfs off** command. Change your `initdefault` in `/etc/inittab` from 3 to 5 again. Restart your computer and observe what happens.

Lab 3

For this lab, you'll need two Linux computers connected over a network. You'll need a shared NFS directory from the local computer. You can use the same directory that you may have used in Chapter 2 to share the RHEL 3 installation files. You'll start a Secure Shell connection between the two computers. You'll start the GUI on the local computer, and use the Secure Shell to log in remotely to the other computer. Finally, you'll set and export an appropriate **DISPLAY** variable. You can then see what happens when you start X Clients from the remote computer.

Once you do, run the Red Hat GUI firewall program from the remote computer. Make changes to the firewall, and see what happens. Finally,

1. On the local computer, start the GUI. If you're currently at the text interface, you can do so with the **startx** command.
2. Open a command line interface. Assuming you're using the default GNOME desktop, right-click on the desktop and click New Terminal in the pop-up menu that appears.

3. In the new terminal, confirm any currently exported directories with the **showmount -e** command. Based on `/etc/exports`, select a directory that is set as writable. Use the techniques described in Chapter 9 if required to make it so. You'll be connecting back to one of these directories from your remote computer.
4. Authorize access from the remote computer. For example, if you're connecting to a computer named `desktop2`, run the following command (you can substitute the IP address):

```
# xhost +desktop2
```

5. Connect to the remote computer using the Secure Shell. Assuming the remote computer is named `desktop2`, run the following command:

```
# ssh root@desktop2
```

(If you have a problem making the connection, you may need to go to the remote computer and activate the Secure Shell service with the **service sshd start** command. You can also substitute the IP address for the computer name.)

6. Enter the root password on the remote computer when prompted.
7. Set and export the **DISPLAY** variable on the remote computer. Make sure it points to the local computer. If the local computer is named `desktop1`, you would use the following command:
8. Now try running the **redhat-config-rootpassword** command. If successful, you'll be changing the root password on the remote computer.
9. Log out of the remote computer. Log back in using the **ssh** command from step 5. Did the root password change?
10. Restore the original root password on the remote computer. Reset the display variable on the remote computer with the following command:

```
# export DISPLAY=localhost:0.0
```

Lab 4

1. Since you're setting up this workstation for a user, you'll want it to start automatically in the GUI. To do so, open the `/etc/inittab` file in a text editor, and make sure the `initdefault` variable is set to `runlevel 5` as follows:

```
id:5:initdefault
```

2. Make sure you don't have other settings defined in the local home directory, in the `~/.xinitrc` file.
3. As you want to start with the `xdm` login manager, you'll want to set it as the preferred login manager in the `/etc/X11/prefdm` file. You can do it by setting the **preferred** variable as shown:

```
preferred=xdm
```

4. Make sure that GNOME is the default desktop. If you see an `.Xclients-default` file, it should contain the following line:

```
exec gnome-session
```

5. If you don't see this line, or the file does not exist, you can set it up and make GNOME the default desktop with the following command:

```
# switchdesk gnome
```

6. Now reboot your computer. From the command line, you can just run the **reboot** command. Alternatively, if you're already in GNOME, click Main Menu | Log Out and select the Reboot option.
7. If you've taken the steps described, you should now see the `xdm` login manager. Log in through that interface.
8. Now in GNOME, click Main Menu | Preferences | More Preferences | Sessions. This opens the Sessions utility.
9. Click the Startup Programs tab. Click Add. This opens the Add A New Session window.
10. Enter the **gnome-terminal** command and click OK.
11. Repeat step 9.
12. Enter the **mozilla** command in Add A New Session window and click OK.
13. Click Close in the Session window.
14. Log out of GNOME, and log back in.
15. You should now see the GNOME desktop with the **gnome-terminal** command line interface and Mozilla Web browser.