

Preface

You've probably heard about Linux from a magazine, radio or TV program, or a friend. You're wondering what Linux is about and whether you should give it a try. If so, particularly if you currently use Microsoft Windows, this book was written for you.

When the first edition of this book was being written, Linux was a much talked about novelty. Today, Linux has invaded corporate information technology departments, becoming a popular technology used by hobbyists and professionals alike.

As predicted in the first edition of this book, Linux is becoming easier to use. If you work with Microsoft Windows and have dabbled a bit in MS-DOS or are curious about what happens inside Windows, you already have the skills to install and configure Linux. Thousands of people from all walks of life have already done so.

Recently, Red Hat—the distributor of the most popular Linux release, Red Hat Linux—split its product line. This book explains both resulting Linux distributions: Red Hat Enterprise Linux and Fedora Core. It includes the two-CD publisher's edition of Fedora Core, containing everything you need to install and configure your own Linux system. This book will make your Linux journey easier, by giving you the big picture, providing you with step-by-step procedures, and getting you started doing useful or fun activities, such as word processing or games. This book focuses on the needs of the new Linux user and on desktop Linux applications. You'll learn about networks and servers, but the details of those topics are left for more advanced books.

Organization of This Book

Chapter 1

This chapter is designed to introduce you to Linux and help you determine whether Linux is appropriate for you.

Chapter 2

This chapter helps you understand what's involved in installing Linux and guides you through the process of gathering information about your system necessary to successfully install Linux.

Chapter 3

This chapter takes you step-by-step through the installation of Linux.

Chapter 4

Before you can effectively use a desktop environment, you need to know some Linux fundamentals. This chapter explains basic Linux concepts that underlie graphical and nongraphical system use.

Chapter 5

This chapter explains how to configure and use the GNOME and KDE desktop environments.

Chapter 6

This chapter introduces you to OpenOffice.org, Linux's desktop suite, which includes a word processor, spreadsheet, presentation graphics, and more. The chapter also introduces you to Evolution, a popular Linux email client and personal scheduler.

Chapter 7

This chapter digs deeper into the *bash* shell, the Linux command-line interface. You'll learn how to use the Linux command-line interface, which resembles MS-DOS but is much more powerful and sophisticated. Here you'll see firsthand just how powerful and easy to use Linux can be.

Chapter 8

This chapter explains the RPM Package Manager, which helps you manage program and applications. It also explains how to use Red Hat Network to keep your system up to date.

Chapter 9

Administering a multiuser operating system such as Linux is somewhat more complicated than administering a single-user operating system, but Linux includes tools that simplify the work. This chapter shows you how to configure your Linux system, including how to configure sound and printers.

Chapter 10

This chapter shows you how to connect via your Internet Service Provider (ISP) to the Internet. Once connected, you can use your Linux system to surf the Web and access other familiar Internet services.

Chapter 11

This chapter shows you how to connect your Linux system to other systems on your Local-Area Network, and how to install and configure servers such as Apache, the world's most popular web server.

Chapter 12

This chapter shows how to use advanced shell features and how to create shell scripts that extend the capabilities of Linux.

Four appendixes are also provided to help you locate important files and directories, manage the way your system boots, and use common Linux commands.

Sources of Information

If you are new to the world of Linux, there are a number of resources to explore and become familiar with. Having access to the Internet is helpful, but not essential.

Red Hat's Web Site

Your primary resource for information on Red Hat Linux is Red Hat's web site, <http://www.redhat.com>. The Red Hat web site includes more resources than can be mentioned here. The final section of [Chapter 3](#) describes several of the most useful resources. Two particularly important resources are:

The redhat-install-list mailing list

<http://www.redhat.com/mailling-lists>

Here, you can obtain installation assistance from members of the Red Hat Linux community.

Bugzilla

<http://bugzilla.redhat.com>

Bugzilla is a database that lists possible bugs affecting Red Hat Linux. The database often gives fixes or workarounds for bugs.

Linux Documentation Project Guides

The Linux Documentation Project (LDP) is a group of volunteers who have worked to produce books (guides), HOWTO documents, and manual pages on topics ranging from installation to kernel programming. More works are in development. For more information about the LDP, consult their web page at <http://www.tldp.org>, or one of its many mirrors. The LDP works include:

Linux Installation and Getting Started

By Matt Welsh et al. This book describes how to obtain, install, and use Linux. It includes an introductory Unix tutorial and information on systems administration, the X Window System, and networking.

Linux System Administrators Guide

By Lars Wirzenius and Joanna Oja. This book is a guide to general Linux system administration and covers topics such as creating and configuring users, performing system backups, configuring major software packages, and installing and upgrading software.

Linux System Administration Made Easy

By Steve Frampton. This book describes day-to-day administration and maintenance issues of relevance to Linux users.

Linux Programmers Guide

By B. Scott Burkett, Sven Goldt, John D. Harper, Sven van der Meer, and Matt Welsh. This book covers topics of interest to people who wish to develop application software for Linux.

The Linux Kernel

By David A. Rusling. This book provides an introduction to the Linux kernel, how it is constructed, and how it works. Take a tour of your kernel.

The Linux Kernel Module Programming Guide

By Ori Pomerantz. This guide explains how to write Linux kernel modules. More manuals are in development. For more information about the LDP you should consult their web server at <http://www.tldp.org> or one of its many mirrors.

HOWTO documents

The Linux HOWTOs are a comprehensive series of papers detailing various aspects of the system—such as installation and configuration of the X Window System software or how to write in assembly language programming under Linux. These are available on the Web at one of the many Linux Documentation Project mirror sites. See the file *HOWTO-INDEX* for a list of what's available.

You might want to obtain the *Installation HOWTO*, which describes how to install Linux on your system; the *Hardware Compatibility HOWTO*, which contains a list of hardware known to work with Linux; and the *Distribution HOWTO*, which lists software vendors selling Linux on diskette and CD-ROM.

Linux Frequently Asked Questions

The *Linux Frequently Asked Questions with Answers* (FAQ) contains a wide assortment of questions and answers about the system. It is a must-read for all newcomers.

Documentation Available Commercially

O'Reilly & Associates publishes a series of Linux books. They include:

Running Linux

This installation and user guide to the system describes how to get the most out of personal computing with Linux.

Linux in a Nutshell

One of the successful "in a Nutshell" series, this book focuses on providing a broad reference text for Linux.

LPI Linux Certification in a Nutshell

While this book is geared toward junior-level system administrators who want to take the Linux Professional Institute's exams for Level 1 Certification (LPIC-1), this book is also a great resource for new users, such as yourself.

Linux Journal and Linux Magazine

Linux Journal and *Linux Magazine* are monthly magazines for the Linux community, written and published by a number of Linux activists. They contain articles ranging from novice questions and answers to kernel programming internals. Even if you have Usenet access, these magazines are a good way to stay in touch with the Linux community.

Linux Journal is the older magazine and is published by SSC, Inc., for which details were listed previously. You can also find the magazine on the Web at

<http://www.linuxjournal.com>.

Linux Magazine is a newer, independent publication. The magazine's homepage is

<http://www.linuxmagazine.com>.

Online Linux Support

There are many ways of obtaining help online, where volunteers from around the world offer expertise and services to assist users with questions and problems.

The Freenode is an IRC network devoted entirely to open projects—open source and open hardware alike. Some of its channels are designed to provide online Linux support services. IRC stands for Internet Relay Chat and is a network service that allows you to type interactively on the Internet to other users. IRC networks support multiple channels on which groups of people communicate. Whatever you type in a channel is seen by all other users of that channel.

There are a number of active channels on the Freenode IRC network where you will find users 24 hours a day, 7 days a week who are willing and able to help you solve any Linux problems you may have, or just chat about their favorite OS. You can use this service by installing an IRC client like *irc-II*, connecting to *servername irc.freenode.net:6667*, and joining the *#linpeople* channel.

Linux User Groups (LUGs)

Many Linux user groups around the world offer direct support to users, and many engage in activities such as installation days, talks and seminars, demonstration nights, and other completely social events. Linux user groups are a great way of meeting other Linux users in your area. There are a number of published lists of Linux user groups. Some of the better-known ones are:

Groups of Linux Users Everywhere

<http://www.ssc.com/glue/groups>

LUG registry

<http://www.linux.org/users>

Web Sites

The following are useful Linux-related web sites. Check them out to get the latest information about Linux. Perhaps the most useful is the home page of the Linux

Documentation Project (LDP). There, you can find almost anything you want to know about Linux. The Linux Documentation Project web site includes a search engine that makes it easy to find what you need.

Linux Documentation Project

<http://www.tldp.org>

Linux Gazette

<http://www.linuxgazette.com>

Linux Today

<http://www.linuxtoday.com>

Linux Web Ring

<http://nll.interl.net/lwr>

The Linux Web Ring offers a convenient way to explore a variety of Linux-related web sites. Participating web sites present links to one another; by following these links, you can circumnavigate the entire ring or you can use the Web Ring's home page to seek exactly the sort of page you're interested in.

Linux Weekly News

<http://www.lwn.net>

O'Reilly & Associates Linux/Unix Center

<http://linux.oreilly.com>

Slashdot

<http://www.slashdot.org>

The motto of the Slashdot web site is "News for nerds. Stuff that matters." You'll find a great deal of interesting news and information there, concerning not only Linux, but the open source community and computing generally.

Conventions Used in This Book

The following typographical conventions are used in this book:

Bold

Used for commands, programs, and options. All terms shown in bold are typed literally.

Italic

Used to show arguments and variables that should be replaced with user-supplied values. Italic is also used to indicate new terms and example URLs, filenames and file extensions, and directories and to highlight comments in examples.

Constant Width

Used to show the contents of files or the output from commands.

Constant Width Bold

Used in examples and tables to show commands or other text that should be typed literally by the user.

Constant Width Italic

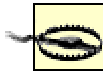
Used in examples and tables to show text that should be replaced with user-supplied values.

#, \$

Used in some examples as the root shell prompt (#) and as the user prompt (\$) under the Bourne or *bash* shell.



This icon signifies a tip, suggestion, or general note.



This icon indicates a warning or caution.

A final word about syntax: in many cases, the space between an option and its argument can be omitted. In other cases, the spacing (or lack of spacing) must be followed strictly. For example, **-wn** (no intervening space) might be interpreted differently from **-w n**. It's important to notice the spacing used in option syntax.

Path Notation

I use a shorthand notation to indicate paths. Instead of writing "From the Start menu, choose Find, then Files or Folders," I write: Start → Find → Files or Folders. I distinguish menus, dialog boxes, buttons, or other GUI elements only when the context would otherwise be unclear. Simply look for the GUI element whose label matches an element of the path.

Keyboard Accelerators

In a keyboard accelerator (such as **Ctrl-Alt-Del**), a dash indicates that the keys should be held down simultaneously, whereas a space means that the keys should be pressed sequentially. For example, **Ctrl-Esc** indicates that the Control and Escape keys should be held down simultaneously; **Ctrl Esc** means that the Control and Escape keys should be pressed sequentially.

Where a keyboard accelerator contains an uppercase letter, you should not type the Shift key unless it's given explicitly. For example, **Ctrl-C** indicates that you should press the Control and C keys; **Ctrl-Shift-C** indicates that you should press the Control, Shift, and C keys.

Chapter 1. Why Run Linux?

Welcome to Linux, the operating system everyone's talking about. Unlike the weather—which proverbial wisdom says you can't do anything about—you *can* do something about Linux. You can run it on your own PC, so that you can see firsthand what the talk is about and perhaps contribute suggestions to its future development.

This chapter is the first leg of your journey into the land of Linux. Here, you'll learn whether this particular journey is right for you and what you can expect down the road. If you're impatient to get started, you can jump ahead to the next chapter, which helps you prepare your PC for installing Linux. But, if you'd like to know more about the history and capabilities of Linux, read on.

1.1 Why Red Hat Enterprise Linux and Fedora?

This book explains Release 3 of Red Hat Enterprise Linux WS and Release 1 of Fedora Core. As explained in the following sections, Red Hat offers several Linux products, or *distributions* as they're called. Moreover, other companies sell or freely provide Linux distributions. Why, then, does this book focus on Red Hat Enterprise Linux WS and Fedora Core?

From the standpoint of market share, Red Hat is the leading provider of Linux distributions in the U.S. and worldwide. To many people, Red Hat Linux *is* Linux. And, among the various Linux distributions provided by Red Hat, Red Hat Enterprise Linux WS and Fedora Core stand out as the most appropriate distributions for desktop users, especially those in corporate environments. To understand why this is so, it's necessary to understand more about Linux, operating systems, and open source (<http://www.opensource.org>) software.

1.2 What Is Linux?

Linux is an *operating system*, a software program that controls your computer. Most PC vendors load an operating system—generally, Microsoft Windows—onto the hard drive of a PC before delivering the PC; so, unless the hard drive of your PC has failed or you've upgraded your operating system, you may not understand the function of an operating system.

An operating system handles user interaction with a system and provides a comfortable view of the system. In particular, it solves several problems arising from variation among hardware. As you're aware, no two PC models have identical hardware. For example, some PCs have an IDE hard drive, while others have a SCSI hard drive. Some PCs have one hard drive; others have two or more. Most PCs have a CD-ROM drive, but some do not. Some PCs have an Intel Pentium CPU, while others have an AMD Athlon, and so on.

Suppose that, in a world without operating systems, you're programming a new PC application—perhaps a new multimedia word processor. Your application must cope with

all the possible variations of PC hardware. As a result, it becomes bulky and complex. Users don't like it because it consumes too much hard drive space, takes a long time to load, and—because of its size and complexity—has more bugs than it should. Operating systems solve this problem by providing a standard way for applications to access hardware devices. Thanks to the operating system, applications can be more compact, because they share the commonly used code for accessing the hardware. Applications can also be more reliable, because common code is written only once—and by expert systems programmers rather than by application programmers.

As you'll soon learn, operating systems do many other things as well; for example, they generally provide a filesystem so you can store and retrieve data and a user interface so you can control your computer. However, if you think of a computer's operating system as its subconscious mind, you won't be far off the mark. It's the computer's conscious mind—applications such as word processors and spreadsheets—that do useful work. But, without the subconscious—the operating system—the computer would cease breathing and applications would not function.

Pronouncing Linux

Internet newsgroup participants have long debated the proper pronunciation of *Linux*. Because Linus Torvalds originated the Linux kernel, it seems reasonable that his pronunciation of the word should reign as the standard. However, Linus is Finnish and his pronunciation of *Linux* is difficult for English speakers to approximate. Consequently, many variations in pronunciation have arisen. The most popular pronunciation sounds as though the word were spelled *Linnucks*, with the stress on the first syllable and a short *i*.

If your computer has a sound card, you can hear how Linus Torvalds pronounces Linux: <http://www.linux.org/info/sounds/english.au>. Linus's personal opinion is that how you pronounce *Linux* matters much less than whether you use it.

1.2.1 Desktop and Server Operating Systems

Now that you know what an operating system is, you may be wondering what operating systems other PC users are using. According to the market research firm IDC, Microsoft products account for over 90 percent of sales of desktop operating systems. Because Linux is a free operating system, Linux sales are a mere fraction of actual Linux installations. Unlike most commercial operating systems, Linux is not sold under terms of a per-seat license; a company is free to purchase a single Linux CD-ROM and install Linux on as many systems as they like.^[1] So, sales figures understate the popularity of Linux. Moreover, it's important to consider who uses a product and what they use it for, rather than merely the number of people using it. Linux is particularly popular among power users who run web sites and databases and write their own code. Hence, though Linux *is* popular, its influence is even greater than its popularity suggests.

^[1] Recently, some Linux vendors, including Red Hat, have begun bundling services with their Linux distributions. Your service agreement with such a vendor may forbid you to

install unlicensed copies of the vendor's distribution or may impose penalties for doing so. See [Section 1.2.7](#) for more information.

Later in this chapter you'll learn how Linux is distributed, but notice that Linux was termed a *free* operating system. If you have a high-speed Internet connection, you can download, install, and use Linux without paying anyone for anything (except perhaps your Internet Service Provider, who may impose a connection fee). It's anyone's guess how many people have downloaded Linux, but it appears that about 10 million computers now run Linux. This book focuses on how Linux can be used on the desktop. However, if you plan to set up a Linux server and are unfamiliar with Linux and Unix, this book is a great starting point. This book will take you through the basics of setting up and using Linux as a desktop system. After you've mastered what this book offers, you should consult *Running Linux*, by Matt Welsh, Matthias Kalle Dalheimer Terry Dawson, and Lar Kaufman (O'Reilly & Associates, Inc.), a more advanced book that focuses on setting up and using Linux servers. You might also enjoy *Linux in a Nutshell*, by Ellen Siever, Stephen Figgins, and Aaron Weber (O'Reilly); this book puts useful Linux reference information at your fingertips.

GNU/Linux

Properly speaking, the name *Linux* applies to the Linux kernel, the most basic and fundamental part of a computer operating system, rather than the entire operating system. Some people prefer to refer to the Linux operating system as GNU/Linux. Doing so emphasizes the contribution of the GNU project—which is described later in this chapter—to the development of the Linux operating system. However, Red Hat calls its operating system Red Hat Enterprise Linux, not Red Hat Enterprise GNU/Linux. Therefore, in this book we refer to both the kernel and operating system as Linux. Context will help you understand whether the entire operating system or only the kernel is meant.

1.2.2 How Linux Is Different

Linux is distinguished from other popular operating systems in three important ways:

- Linux is a cross-platform operating system that runs on many computer models. Only Unix, an ancestor of Linux, rivals Linux in this respect. In comparison, Windows 2000 and XP run only on CPUs having the Intel architecture.
- Linux is free, in two senses. First, as mentioned earlier, you can obtain and use Linux without paying anything to anybody. On the other hand, you may choose to purchase Linux from a company that bundles Linux with special documentation or applications or provides technical support.

Second, and more important, Linux and many Linux applications are distributed in source form. Thus, if you have the necessary skill, you're free to modify or improve them. You're *not* free to do this with most operating systems, which are distributed in binary form. For example, you can't make changes to Windows or Office—only

Microsoft can do that. Because of this freedom, Linux is being constantly improved and updated, far outpacing the rate of progress of any other operating system. For example, Linux was the first operating system to support Intel's Itanium 64-bit CPU.

- Linux has more attractive features and performance. Free access to Linux source code lets programmers around the world implement new features and tweak Linux to improve its performance and reliability. The best of these features and tweaks are incorporated in the Linux kernel or made available as kernel patches or applications. Not even Microsoft can mobilize and support a software development team as large and dedicated as the volunteer Linux software development team, which numbers in the hundreds of thousands, including programmers, code reviewers, and testers.

1.2.3 The Origins of Linux

Linux traces its ancestry back to a mainframe operating system known as Multics (Multiplexed Information and Computing Service). Multics was one of the first multiuser computer systems and is still in use today. Participating in its development, which began in 1965, was Bell Telephone Labs, along with the Massachusetts Institute of Technology (MIT) and General Electric.

Two Bell Labs software engineers, Ken Thompson and Dennis Ritchie, worked on Multics until Bell Labs withdrew from the project in 1969. One of their favorite pastimes during the project was playing a multiuser game called Space Travel. Without access to a Multics computer, they found themselves unable to indulge their fantasies of flying around the galaxy. Resolving to remedy this, they decided to port the Space Travel game to run on an otherwise unused PDP-7 computer. Eventually, they implemented a rudimentary operating system they named *Unics*, as a pun on *Multics*. Somehow, the spelling of the name became *Unix*.

Their operating system was novel in several respects, most notably its portability. Most previous operating systems had been written for a specific target computer. Just as a tailor-made suit fits only its owner, such an operating system could not be easily adapted to run on an unfamiliar computer. In order to create a portable operating system, Ritchie and Thompson first created a programming language called *C*. Like assembly language, *C* let a programmer access low-level hardware facilities not available to programmers writing in a high-level language such as FORTRAN or COBOL. But, like FORTRAN and COBOL, a *C* program was not bound to a particular computer. Just as a ready-made suit can be altered here and there to fit a purchaser, writing Unix in *C* made it possible to easily adapt Unix to run on computers other than the PDP-7.

As word of their work spread and interest grew, Ritchie and Thompson made copies of Unix freely available to programmers around the world. These programmers revised and improved Unix, sending word of their changes back to Ritchie and Thompson, who incorporated the best improvements in their version of Unix. Eventually, several Unix variants arose. Prominent among these was BSD (Berkeley Systems Division) Unix, written at the University of California, Berkeley, in 1978. Bill Joy—one of the principals of the BSD project—later became a founder of Sun Microsystems, which sold another Unix variant (originally called SunOS and later called Solaris) to power its workstations. In

1984, AT&T, the parent company of Bell Labs, began selling its own version of Unix, known as System V.

1.2.4 Free Software

What Ritchie and Thompson began in a distinctly noncommercial fashion ended up spawning several legal squabbles. When AT&T grasped the commercial potential of Unix, it claimed Unix as its intellectual property and began charging a hefty licensing fee to those who wanted to use it. Soon, others who had implemented Unix-like operating systems were distributing licenses only for a fee. Understandably, those who had contributed improvements to Unix considered it unfair for AT&T and others to appropriate the fruits of their labors. This concern for profit was at odds with the democratic, share-and-share-alike spirit of the early days of Unix.

Some, including MIT scientist Richard M. Stallman, yearned for the return of those happier times and the mutual cooperation of programmers that had existed. So, in 1983, Stallman launched the GNU (*GNU's not Unix*) project, which aimed at creating a free Unix-like operating system. Like early Unix, the GNU operating system was to be distributed in source form so that programmers could read, modify, and redistribute it without restriction. Stallman's work at MIT had taught him that, by using the Internet as a means of communication, programmers could improve and adapt software at incredible speed, far outpacing the fastest rate possible using traditional software development models, in which few programmers actually see one another's source code.

As a means of organizing work on the GNU project, Stallman and others created the Free Software Foundation (FSF), a nonprofit corporation that seeks to promote free software and eliminate restrictions on the copying, redistribution, understanding, and modification of software. Among other activities, the FSF accepts tax-deductible charitable contributions and distributes copies of software and documentation for a small fee, using this revenue to fund its operations and support development activities.

Copyleft

Commercial software vendors protect their proprietary rights to software by copyrighting the software. In contrast, the FSF protects software freedom by *copylefting* its software under the GNU General Public License (GPL). To copyleft software, the FSF uses the same legal instrument used by proprietary software vendors—the copyright—but the GPL includes special license terms that guarantee freedom to users of the software. These terms give everyone the right to use, modify, and redistribute the software (or any software derived from it), but only if the distribution terms are unchanged. Thus, someone who attempts to transform FSF software into a proprietary product thereby loses the right to use, modify, or distribute the product.

If, rather than copyright its software, the FSF placed its software in the public domain, others would be free to transform it into a proprietary product, denying users the freedom intended by the original author of the software. For example, a company might distribute the software in binary rather than source form and require payment of a license fee for the privilege of making additional copies.

As the FSF puts it: "Proprietary software developers use copyright to take away the users' freedom; we use copyright to guarantee their freedom. That's why we reverse the name, changing *copyright* into *copyleft*."

If you find it peculiar that the FSF charges a fee—even a small fee—for "free" software, you should understand that the FSF intends the word *free* to refer primarily to freedom, not price. The FSF believes in three fundamental software freedoms:

- You can copy GNU software and give it away to anyone you choose.
- If you're a programmer, you can modify GNU software any way you like, because you have access to the source code. In return, your modified code should be available for others so they can enjoy the privileges of learning from and modifying it.
- You must provide a free copy of the source so that you cannot unfairly profit by changing the original.

1.2.5 The Linux Kernel

By the early 1990s, the FSF had obtained or written all the major components of the GNU operating system except for one: the kernel. About that time, Linus Torvalds, a Finnish computer science student, began work on a kernel for a Unix-like system. Linus had been working with Minix, a Unix-like operating system written by Andrew Tannenbaum primarily for pedagogical use. Linus was disappointed by the performance of the Minix kernel and believed that he could do better. He shared his preliminary work with others on Internet newsgroups. Soon, programmers around the world were working together to extend and improve his kernel, which became known as Linux (for *Linus's Minix*). As [Table 1-1](#) shows, Linux grew rapidly. Linux was initially released on October 5, 1991, and as early as 1992, Linux had been integrated with GNU software and other open source software (www.opensource.org) to produce a fully functional operating system, which became known as Linux after the name of its kernel.

Table 1-1. The history of Linux

Version	Year	Estimated users	Kernel size (Kb)	Milestone(s)
0.01	1991	100	63	Linus Torvalds writes the Linux kernel.
0.99	1992	1000	431	GNU software is integrated with the Linux kernel, producing a fully functional operating system.
0.99	1993	20,000	938	High rate of code contributions prompts Linus to delegate code review responsibility.
1.0	1994	100,000	1017	First production kernel is released.
1.2	1995	500,000	1850	Linux is ported to non-Intel processors.

Table 1-1. The history of Linux

Version	Year	Estimated users	Kernel size (Kb)	Milestone(s)
2.0	1996	1,500,000	4718	Linux supports multiple processors, IP masquerading, and Java.
2.2	1999	7,500,000	10,593	Linux growth rate exceeds that of Windows NT, according to market research firm Dataquest.
2.4	2001	10,000,000	19,789	Linux invades the enterprise as major companies begin using it.
2.6	2003	20-50,000,000	32,476	Linux improved to include additional enterprise-level features, such as improved support for threads.

Work on Linux has not ceased. Since the initial production release, the pace of development has accelerated as Linux has been adapted to include support for non-Intel processors and even multiple processors, sophisticated TCP/IP networking facilities such as firewalling, network address translation (NAT), and more. Versions of Linux are now available for such computer models and architectures as the PowerPC, the Compaq/DEC Alpha, the Motorola 68k, the Sun SPARC, the MIPS, IBM mainframes, and many others. Moreover, Linux does not implement an obscure Unix variant: it generally complies with the POSIX (Portable Operating System Interface) standard that forms the basis of the X/Open specifications of The Open Group.

1.2.6 The X Window System

Another important component of Linux is its *graphical user interface* (GUI; pronounced *goeey*), the *X Window System*. Unix was originally a mouse-less, text-based system that used noisy teletype machines rather than modern video monitors. The Unix command interface is very sophisticated and, even today, some power users prefer it to a point-and-click graphical environment, using their video monitors as though they are noiseless teletypes. Consequently, some remain unaware that Unix long ago outgrew its text-based childhood and now provides users a choice of graphical or command interfaces.

The X Window System (or simply *X*) was developed as part of MIT's Project Athena, which it began in 1984. By 1988, MIT released X to the public. Responsibility for X has since been transferred to The Open Group. The XFree86 Project, Inc. provides a freely redistributable version of X that runs on Intel-architecture PCs.

X is a unique graphical user interface in three major respects:

- X integrates with a computer network, letting users access local and remote applications. For example, X lets you open a window from which you can interact with an application running on a remote host: the remote host does the heavy-duty

computing; all your computer need do is pass the host your input and display the resulting output.

- X lets you configure its look and feel to an amazing degree. To do so, you run a special application—called a *window manager*—on top of X. A variety of window managers is available, including some that closely mimic the look and feel of Microsoft Windows. Desktop managers further extend X by providing common applications such as file browsers, menus, and control panels. GNOME and KDE are the most popular Linux desktop managers and are discussed in this book.
- X is optional. Systems used as servers are often configured without a GUI, saving resources to serve client requests.

1.2.7 Linux Distributions

Because Linux can be freely redistributed, you can obtain it in a variety of ways. Various individuals and organizations package Linux, often combining it with free or proprietary applications. Such a package that includes all the software you need to install and run Linux is called a *Linux distribution*. Table 1-2 shows some of the most popular Linux distributions.

Table 1-2. Popular Linux distributions and their home pages	
Distribution	Home page
Debian GNU/Linux	http://www.debian.org
Fedora Core	http://fedora.redhat.com
Gentoo Linux	http://www.gentoo.org
Mandrake Linux	http://www.mandrakelinux.com
Red Hat Enterprise Linux	http://www.redhat.com
Slackware Linux	http://www.slackware.com
SuSE Linux	http://www.suse.com

Red Hat Enterprise Linux, Mandrake Linux, SuSE, and Slackware are packaged by commercial companies, which seek to profit by selling Linux-related products and services. However, because Linux is distributed under the GNU GPL, you can download the source code related to these distributions from the respective companies' web sites and make additional copies. (Note, however, that you cannot necessarily make additional copies of proprietary software that these companies may distribute with their Linux distribution.) Debian GNU/Linux is the product of volunteer effort conducted under the auspices of Software in the Public Interest, Inc. (<http://www.spi-inc.org>), a nonprofit corporation. Fedora Core and Gentoo are also the products of volunteer efforts. However, Red Hat provides significant support to the team responsible for Fedora Core.

1.2.8 Red Hat's Linux Distributions

Red Hat formerly provided a single series of Linux distributions known as Red Hat Linux. Red Hat Linux included distinct offerings for various uses (for instance, workstations versus servers), but the offerings were all based on one stable core. Now Red Hat provides several Linux distributions, and it's worth understanding the strengths and weaknesses of each. At the same time, remember that the distributions all include essentially the same software.

Red Hat came to recognize that its distribution interests two very different types of people. One type is business clients who want stable software that comes with support and are willing to pay significant money for that support. If the software changes slowly, that is fine with such users because it means fewer wrinkles and less time spent on upgrades. The other type of people who use Red Hat love to experiment and make use of the newest, most advanced features of Linux. These people don't mind upgrading frequently, but they are not running mission-critical systems and don't want to pay for support. In fact, they're used to getting Linux at no cost.

Red Hat decided that by offering multiple distributions, it could make both types of users happy—and get revenue from its efforts—while benefiting from the testing and experimentation of the user community. Thus, any software that bears the trademark "Red Hat" is licensed software. It is backed by Red Hat in some manner, often through support contracts that include upgrades. It is expected to change only once every two or three years. By contrast, the freely downloadable version of the distribution is called "Fedora Core" or simply "Fedora." Fedora comes without Red Hat support. New versions of Fedora are released every few months and contain a lot of new features.

Right now, the Red Hat Enterprise Linux distribution and Fedora are very similar. To readers of this book, they essentially work the same way, and only some details of installation differ. We include the publisher's edition of Fedora Core on two CDs, but we have tested the material thoroughly with both Fedora Core and Red Hat Enterprise Linux, so the book applies to both.

Naturally, if you install another version of either system, you may find minor differences between the menus or software versions in this book and the ones on your system. However, most of the things described in this book have been stable for some time and are not likely to change quickly.

Red Hat Enterprise Linux, Red Hat Professional Workstation, and Fedora Core Linux

Red Hat formerly provided a Linux distribution known as Red Hat Linux. In late 2003, Red Hat discontinued Red Hat Linux and now provides two Linux distributions:

- Red Hat Enterprise Linux
- Red Hat Professional Workstation

The Red Hat Enterprise Linux product line includes three products, which are

available exclusively through Red Hat rather than sold by retailers:

- Red Hat Enterprise Linux WS, intended primarily for desktop use.
- Red Hat Enterprise Linux ES, intended for small- to mid-range servers with one or two CPUs.
- Red Hat Enterprise Linux AS, intended for larger and mission-critical servers, and users who require high levels of support.

Red Hat Professional Workstation includes the same software sold as Red Hat Enterprise Linux WS. However, Red Hat Professional Workstation is sold at retail and has a license agreement different from that associated with Red Hat Enterprise Linux WS.

More up-to-date information on the Red Hat Enterprise Linux product line, including current prices for annual licenses, is available at

<http://www.redhat.com/software/rhel>.

In addition, Red Hat supports the volunteer team responsible for Fedora Core and uses that distribution as a testing ground for features being considered for incorporation into Red Hat Enterprise Linux. The publisher's edition of Fedora Core is included with this book. The publisher's edition of Fedora Core includes two CD-ROM diskettes rather than the three diskettes comprising the standard edition of Fedora Core.

Despite the outward similarity between Red Hat Enterprise Linux and Fedora Core, these distributions reflect divergent goals and methods. The Fedora Core project team promises 2-3 releases per year of their distribution, a rapid development pace that will enable them to incorporate the most recent available versions of Linux software in a timely manner.

However, Fedora Core's rapid release cycle does not afford opportunity to perform quality assurance appropriate for enterprise environments, where reliability and security are crucial. Moreover, updates to a given release of Fedora Core are promised to be available only until 2-3 months after a subsequent release. Thus, users of Fedora Core can expect to have to install a new release of Fedora Core every 6-9 months or leave their systems vulnerable to security flaws for which updates are not available. Consequently, Fedora Core is best suited for use by hobbyists and others interested in sampling the bleeding edge of Linux technology and features.

Red Hat Enterprise Linux has a longer release cycle of 12-18 months. And, updates to releases of Red Hat Enterprise Linux are promised to be available for five years after each release. Consequently, users of Red Hat Enterprise Linux will not need to re-install their operating system as often as users of Fedora Core. Moreover, the longer release cycle of Red Hat Enterprise Linux permits third-party certification of Red Hat Enterprise Linux applications and hardware vendor support of systems running Red Hat Enterprise Linux, considerations which are important, even crucial, to enterprise users. However, the long release cycle is not without its drawbacks. For instance, at any given time, several releases of Red Hat Enterprise Linux may be in use. So, if you're using a release of Red Hat's Enterprise Linux other than Red Hat Enterprise Linux WS 3, you'll probably find some differences between what's installed on your system and what's shown in this book.

1.2.9 Linux Features and Performance

The origins of Linux and the availability of its source code set it apart from other operating systems. But most users choose an operating system based on features and performance—and Linux delivers these in spades.

Linux runs on a wider range of hardware platforms and runs adequately on less costly and powerful systems than other operating systems. Moreover, Linux systems are generally highly reliable.

But this impressive inventory of selling points doesn't end the matter. Let's consider some other technical characteristics of Linux that distinguish it from the pack:

Cost

Foremost in the minds of many is the low cost of Linux. Comparable server operating systems can cost more than \$100,000. On the other hand, the low cost of Linux makes it practical for use even as a desktop operating system. In that mode, it truly eclipses the competition.

Power

Many desktop systems are employed as servers. Because of its design and heritage, the features and performance of Linux readily outshine those of desktop operating systems used as makeshift servers. For example, Microsoft's software license for Windows NT/2000/XP restricts the number of authenticated client connections; if you want your Windows NT/2000/XP server to be able to handle 100 authenticated clients, you must pay Microsoft a hefty license fee. However, Linux imposes no such restriction; your Linux desktop or server system is free to accept as many client connections as you think it can handle.

Availability

Properly configured Linux systems are quite stable—almost as stable as the hardware on which they run. Moreover, installing Linux software or updates doesn't generally require you to reboot your system. So, Linux systems tend to be highly available.

Reliability

Again, because of its design and heritage, Linux provides more reliable data storage than competing desktop operating systems. Most Linux users store their disk data using the *ext3* filesystem, which is superior in performance and reliability to filesystems (partition types) provided by Microsoft operating systems, including FAT, FAT32, and NTFS. Of course, if you're worried about losing a whole disk to hardware failure, you can outfit Linux with the powerful facility known as Redundant Array of Independent Disks (RAID).

1.2.9.1 Filesystem reliability

Microsoft claims that its NTFS filesystem is so reliable that you'll probably never need special software tools to recover lost data—truth is, Microsoft provides no such tools.

Despite Microsoft's ambitious claims, some Windows NT users report that NTFS reliability is less than satisfactory.

Here's a case in point: When my Windows NT workstation crashed a little over a year ago I discovered that its NTFS filesystem was damaged. I searched the Microsoft web site for recovery instructions and tools and found nothing that helped. So I went to my local software store and purchased a third-party disk recovery tool for Windows NT. When I opened the box, I was angered to discover that it supported recovery of FAT and FAT32 data, but not NTFS data.

Eventually, I recovered 95 percent of my data by using a free Linux utility that was able to open the damaged NTFS partition and copy its files. If I'd been without Linux, I'd be without my data.

1.2.9.2 The command-line interface

If you're an old computer dog who remembers the days of MS-DOS, you may have a fondness for what's now called the MS-DOS Prompt window or the Command Line Interface (CLI). However, if you've worked exclusively within the Windows point-and-click environment, you may not fully understand what the MS-DOS Prompt window is about. By typing commands in the MS-DOS Prompt window, you can direct the computer to perform a variety of tasks.

For most users, the MS-DOS Prompt is not as convenient as the GUI offered by Windows. That's because you must know the commands the operating system understands and must type them correctly if you expect the operating system to do your bidding.

However, the MS-DOS Prompt window lets you accomplish tasks that would be cumbersome and time-consuming if performed by pointing and clicking. Linux comes with a similar command interface, known as the *shell*. But, the word *similar* fails to do justice to the Linux shell's capabilities, because the MS-DOS Prompt provides only a fraction of the capabilities provided by the Linux shell.

You may have used the MS-DOS Prompt and, finding it distastefully cumbersome, forever rejected it in favor of pointing and clicking. If so, you'll be pleasantly surprised to see how easy it is to use the Linux shell. You'll certainly be pleased—perhaps amazed—by the enormous power it offers. Moreover, you can customize the operation of the Linux shell in an almost limitless number of ways and even choose from among a variety of shells, and automate your work by combining commands into files called *shell scripts*. You'll learn more about the Linux shell in [Chapter 7](#).

1.2.9.3 Developing portable code

If you're a programmer, you'll also admire the ease with which it's possible to develop portable, Unix-compliant software. Linux comes with a suite of software development tools, including an assembler, C/C++ compilers, a *make* application, and a source code librarian. All of these are freely distributable programs made available under the terms of the GNU GPL.

1.3 Reasons to Choose or Not Choose Linux

Notwithstanding its high points, Linux is not for everyone. You should approach your decision to use Linux as you'd approach any decision, by evaluating the pros and cons. Here are several reasons to run Linux:

- **You want a stable and reliable computing platform.** No popular operating system is more stable and reliable than Linux. If you're tired of crashes and hangs and the lost time and data they entail, you're a candidate for Linux.
- **You want a high-performance computing platform.** Linux can coax blazingly fast performance out of hardware below the minimum required to load and run other popular operating systems. And with ample memory and a fast CPU, Linux goes toe-to-toe with anything Microsoft or other vendors offer. If speed is your thing, Linux is your hot rod.
- **You need a low-cost or free operating system.** If you're on a budget or if you need to set up many systems, the low cost of Linux will let you reserve your hard-earned capital for hardware or other resources. Linux is the best operating system value on the planet.
- **You're a heavy network or Internet user.** If you use networks, especially the Internet, Linux's advanced support for TCP/IP may light up your life. Linux makes it easy to construct firewalls that protect your system against hackers or routers that let several computers share a single network connection.
- **You want to learn Unix or TCP/IP networking.** The best way—perhaps the only way—to learn more about Unix or TCP/IP networking (or computers generally) is through hands-on experience. Whether you're interested in such experience owing to personal curiosity or career ambition (system administrators are often handsomely paid), Linux affords you the opportunity to gain such experience at low cost, without leaving the comfort of your home.
- **You seek an alternative to Microsoft's vision of computing's future.** If you're tired of marching to the relentless drumbeat of the Redmond juggernaut, Linux offers a viable way to cut the umbilical cord and set about creating a new, open source computing destiny for yourself and others.
- **You want to have fun.** Hopefully, you've discovered that one of the best reasons for doing anything is that it's fun. Many Linux users report that they've never had so much fun with a computer. There's no better reason for running Linux than that.

To be frank, some folks shouldn't run Linux. If one or more of the following are true of you, you should run Linux *only* if you have a good friend who's knowledgeable about Linux, available by phone at odd hours, and works cheap:

- **You're scared of computers.** If you're scared of computers, you should spend more time working with Windows 2000/XP or Mac OS X before venturing into the Linux world. Linux may indeed be right for you, but it's not right just yet.
- **You don't like, or don't have the time and inclination, to tinker and learn.** Setting up and running Linux will require you to learn new concepts and skills. None of these is especially difficult, but unless you're enthusiastic about learning

and playing around, setting up and running Linux will stress you out. Instead, you should stick with the familiar.

- **You're married to certain Windows applications.** You can run some Windows applications under Linux's WINE emulation, but this isn't true of every Windows application. Before putting your toe in the Linux waters, you should obtain up-to-date information on the status of WINE emulation of your favorite Windows applications (see <http://www.winehq.com>). Alternatively, you can purchase the commercial products VMware (see <http://www.vmware.com>), Win4Lin (see <http://www.netraverse.com>), or CrossOver Office (<http://www.codeweavers.com>) that enable you to run Windows applications or Windows itself under Linux.

Rather than convert your desktop system to run Linux, you may prefer to install Linux on a second system or to set up your computer as a dual-boot system, running both Windows and Linux on separate partitions of a single hard drive. That way, you have your choice of running your favorite Windows applications or Linux whenever you desire.

On the other hand, the quality of Linux applications continues to improve. Many Linux users are quite satisfied with Linux applications and desktop suites such as OpenOffice.org, which provides word processing, spreadsheet, and presentation programs. So, unless your marriage to Windows applications includes a formal vow, you may want to reconsider your marital status.

- **You're addicted to 3D gaming.** 3D gaming is a resource-intensive form of computing. To provide the best possible gaming experience, designers of 3D games strive to take full benefit of advanced capabilities of sophisticated graphics adapters. However, manufacturers of graphic adapters do not always provide Linux-compatible drivers. And, even if a Linux-compatible driver is available for your graphics adapter, it may not fully support the adapter's capabilities. Moreover, relatively few game publishers release Linux versions of their games. So, if you're a 3D game aficionado, you'll likely prefer to keep one foot in the Windows world. But, that doesn't mean that you shouldn't run Linux; only that you probably shouldn't run *only* Linux.

Still game? Excellent! Please proceed to [Chapter 2](#) and prepare to install Linux.

Chapter 2. Preparing to Install Linux

Before installing Linux, you must first gather some data about your system. This chapter presents information you need to know and tasks you need to perform before the installation of Linux. It helps you make certain that your IBM-compatible PC meets the minimum hardware requirements for Linux. It shows you how to document your Windows

operating system configuration so that you can respond to questions presented by the Linux install procedure. And, finally, it shows you how to prepare your hard disk for Linux.



Various Linux distributions have somewhat different hardware requirements. To avoid tedious repetition, I refer to Red Hat Enterprise WS 3 and Fedora as simply "Linux" throughout this chapter. But, please bear in mind that the information in this chapter may not accurately apply to Linux distributions other than Red Hat Enterprise Linux WS 3 and Fedora.

2.1 Minimum Hardware Requirements

Linux supports a wide range of PC hardware, but not even Linux supports every known device and system. Your PC must meet certain minimum requirements in order to run Linux, which I describe in the following sections. For the latest and most complete information, you should check Red Hat's hardware compatibility web site, <http://hardware.redhat.com/hcl>. This site will also help you determine whether Linux supports the devices installed in your system. If you're not familiar with PC hardware, check out Robert and Barbara Thompson's *PC Hardware in a Nutshell: A Desktop Quick Reference* (O'Reilly), an excellent introduction and reference to PC hardware.

2.1.1 Central Processing Unit (CPU)

Linux does not support the Intel i386 and earlier processors. However, it fully supports the Intel i486, Celeron, Pentium, Pentium Pro, Pentium II, Pentium III, and Pentium IV processors and compatible processors manufactured by others, such as AMD's Athlon, Athlon XP, and Athlon MP. Such processors are members of what is known as the *x86 family* of processors. Linux also supports non-Intel processors such as AMD's AMD64 processor, IBM's PowerPC processor, and certain processors used in IBM mainframes. And, it supports Intel's Itanium processor.



Although Linux supports processors other than members of Intel's x86 processor family, special measures are required to install it on such processors. This book describes the installation, configuration, and use of Linux on only the x86 family of processors.

Apart from the processor model, there's the issue of processor speed. You should generally have a 400 MHz Pentium II or faster processor for satisfactory results using Linux desktop applications.

2.1.2 Motherboard

The motherboard is the main part of a PC. It holds the CPU, RAM, and other internal computer components, linked by several buses. Linux supports the standard ISA, EISA,

PCI, and VESA (VLB) system buses used on most IBM-compatible PCs, as well as the AGP, and USB auxiliary buses. Fedora Core includes limited support for the IEEE 1394 (FireWire) bus. However, the related software components are largely untested. Your motherboard should include at least 256 MB of RAM for optimum Linux performance. However, this figure is merely a guideline. Many people manage to install Linux on systems having only 192-256 MB of RAM. Some especially determined and skilled users have managed to coax Linux into working on systems with as little as 4 MB of RAM by using their own installation programs or methods. However, unless you're a skilled system administrator and programmer, it's not recommended that you attempt such an exotic installation.



Red Hat does not support systems having less than the required amount of RAM.

A handful of motherboards present special problems when installing Linux. Generally, problems stem from bad BIOS, for which a fix is often available. Check the Red Hat web site for details; the best way to do so is via the knowledgebase search page at <http://www.redhat.com/apps/support>.

2.1.3 Drives

An anonymous wag once quipped that one can never be too thin, be too rich, or have too much hard disk space. Fortunately, Linux is not extremely hungry for disk space. To install and use Linux, you should have a minimum of 4000 MB (4 GB) of free hard disk space. More realistically, you should have at least 10 GB.



The "free hard disk space" required by Linux must be truly free, unpartitioned space. you cannot install Linux into a partition owned by another operating system, such as Windows. However, Linux and Windows can co-exist on a single system, so long as each operation system has its own partition or partitions. For more information on partitions and partitioning, see [Section 2.3](#).

For convenient installation using the Linux installation CDs, your system needs a CD-ROM drive (both common types, IDE and SCSI, are acceptable). Most recently manufactured PCs can boot from a CD-ROM. If your PC can't do so, your system should include a 3.5-inch floppy drive. You'll use the floppy drive to boot your system from a special Linux diskette you create. Instructions for how to create the boot floppy can be found in [Chapter 3](#).



It's also possible to install Linux from a PCMCIA CD-ROM drive; an FTP, Web, or NFS server; or a hard drive. See the *Red Hat Enterprise Linux Installation Guide* and *Red Hat Enterprise Linux Reference Guide*, included on the Red Hat Enterprise Linux installation media. These documents are also available at <http://www.redhat.com/docs>. Because Fedora was only recently inaugurated, it does not yet have

documentation of a quality comparable to that included with Red Hat Enterprise Linux. However, the Red Hat Enterprise Linux documentation contains much information that is also applicable to Fedora.

2.2 Collecting Information About Your System

Before you launch into the installation process, you should collect some basic information about your system. Generally, Red Hat's installer will successfully probe your system and discover its configuration, but when it fails to do so, you must be prepared to supply the required information. Otherwise, you'll be forced to terminate the installation procedure, obtain the information, and then start all over again.

2.2.1 Information You Need

[Table 2-1](#) specifies the configuration information you need and gives you space to conveniently record the information as you gather it. If your system currently runs Windows, you can obtain much of the needed information by using Windows utilities, as explained in the next section. To obtain the remaining information, you can consult your system documentation and the documentation for any devices installed by you. If your documentation is missing or incomplete, you may need to contact your hardware vendor or manufacturer. Alternatively, you may be able to find the needed information on the manufacturer's web site; use a search engine such as Google (which is powered by Linux) to discover the URL of the web site.

Sometimes, you'll need to examine your system's BIOS settings or open your system's case and examine the installed hardware; consult your system documentation to learn how to do so. Finally, if you're installing Linux in a large organization such as a business or a university, your system administrator is likely to be an excellent source of the information you need.

Table 2-1. Configuration information needed to install Linux

Device	Information needed	Your information
Hard drive(s)	The number, size, and model of each hard drive. Which hard drive is first, second, and so on. Which adapter type (IDE or SCSI) is used by each drive. For each IDE drive, whether or not the BIOS is set for LBA mode. The number and type of each existing partition and the amount of free disk space.	
CD-ROM drive(s)	Which adapter type (IDE, SCSI, or other) is used by each drive. For each drive using a non-IDE, non-SCSI adapter, the make and model of the drive.	

Table 2-1. Configuration information needed to install Linux

Device	Information needed	Your information
SCSI adapter (if any)	The make and model of the adapter.	
RAM memory	The amount of installed RAM.	
Video adapter	The make and model of the adapter and the amount of installed video RAM.	
Video monitor	The make and model of the video monitor and the manufacturer's specifications, if available, especially the horizontal and vertical sync (refresh) rates.	
Mouse	The type (serial, PS/2, or bus). The protocol (Microsoft, Logitech, MouseMan, etc.). The number of buttons. For a serial mouse, the serial port to which it's connected (COM1 or COM2).	
Sound adapter (if any)	The make, chipset, and model of the adapter.	
Network adapter (if any)	The make and model of the card.	
IP address	The dotted-quad number, such as 10.1.2.7, that identifies your system to other Internet hosts. Many Internet Service Providers (ISPs) assign IP addresses dynamically, by using a DHCP server; in that case, you don't need to know the IP address of your system.	
Netmask	A dotted-quad number that identifies the portion of your system's IP address that specifies its network address. The number typically contains one or more instances of the value 255. If your ISP is using DHCP, you don't need to know the netmask.	
Gateway IP address	The IP address of the host that routes traffic between your system and the Internet. If your ISP is using DHCP, you don't need to know the gateway IP address.	
DNS server	The IP address of one or more Internet hosts that provide domain name services (DNS) for your system. If your ISP is using DHCP, you don't need to know the DNS servers.	

Table 2-1. Configuration information needed to install Linux

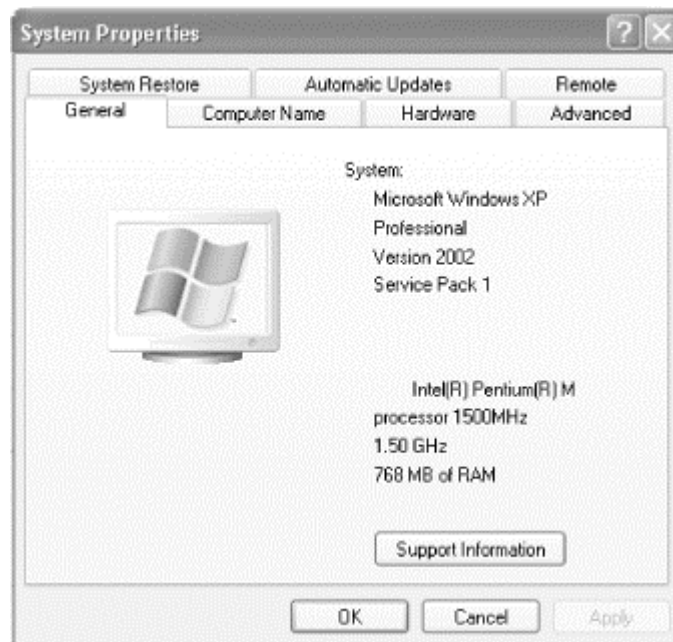
Device	Information needed	Your information
Domain name	The domain name of your system. This generally looks like xxx.net or xxx.com, with xxx identifying your Internet Service Provider. If your ISP is using DHCP, you don't need to know the domain name.	
Hostname	The hostname of your system. You can generally find the hostname prefixed to the domain name. If your ISP is using DHCP, you don't need to know the hostname.	

2.2.2 Collecting Configuration Information from Windows

If you run Windows 95/98, 2000, or XP, you can obtain much of the information needed to install Linux by using the Windows System Properties dialog box, which you can launch by using the Control Panel. Get the information as follows:

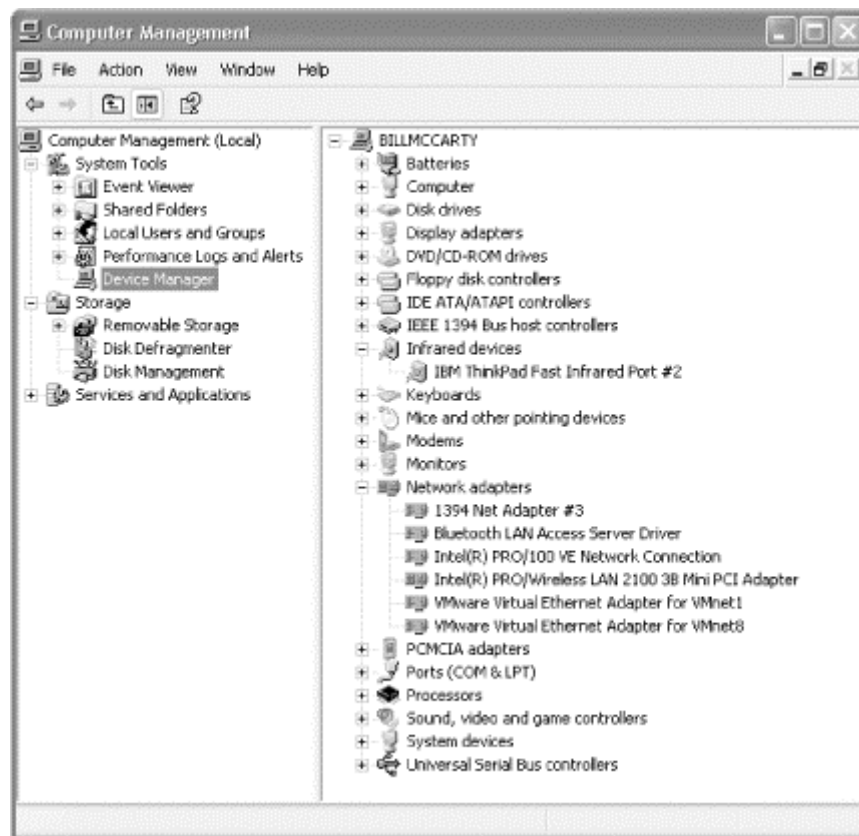
1. Click on the Start menu. A pop-up menu appears.
2. Click on Control Panel, if that menu item is available. Otherwise, select Settings and then click on Control Panel. The Control Panel appears.
3. Click (Windows XP) or double-click (Windows 95/98/2000) on System. The System Properties dialog box appears. If necessary, click on the General tab, so that the dialog box resembles the one shown in [Figure 2-1](#).

Figure 2-1. The General tab of the System Properties dialog box



4. The General tab of the System Properties dialog box shows the type of your system's processor and the amount of installed RAM.
5. Click on the Device Manager Tab (Windows 95/98) or Control Panel Administrative Tools Computer Management Device Manager (Windows 2000/XP) tab. The path you use to find the Device Manager may vary a bit, depending on your operating system and its configuration. The Device Manager appears, as shown in [Figure 2-2](#).

Figure 2-2. The Device Manager



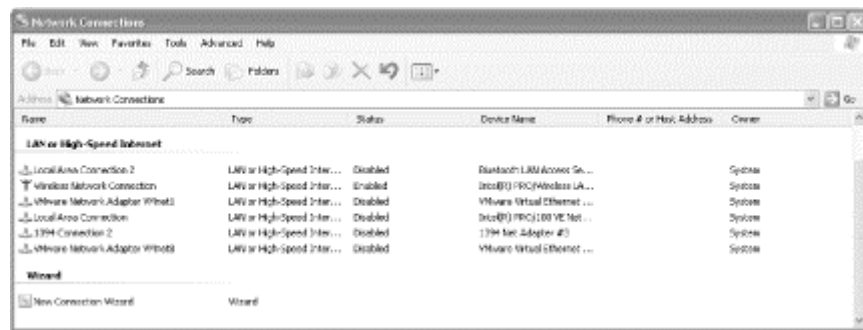
6. You can double-click on an icon (or single-click on the Plus key adjacent to an icon) to obtain additional information.
7. You can gather the following information from the Device Manager tab:
 - o The number and type (IDE or SCSI) of your system's hard drives
 - o The make and model of CD-ROM drives
 - o Under Windows 95/98, some installed CD-ROM drives do not appear in the Device Manager tab of the System Properties dialog box. Often the *C:\CONFIG.SYS* file will contain clues that help you learn more about such drives.
 - o The make and model of SCSI adapters, if any

- The make and model of the video adapter
 - The type of mouse installed
 - The make and model of multimedia adapters, such as sound cards, if any
 - The make and model of network adapters, if any
8. When you've recorded the information provided by the Device Manager tab, click Cancel to exit the System Properties dialog box.

Linux supports dial-up networking, as explained in [Chapter 10](#). But the installation program doesn't configure dial-up networking, so you don't need to collect information about dial-up networking prior to installation. However, if your computer is attached to a network, you should collect information describing your network adapter:

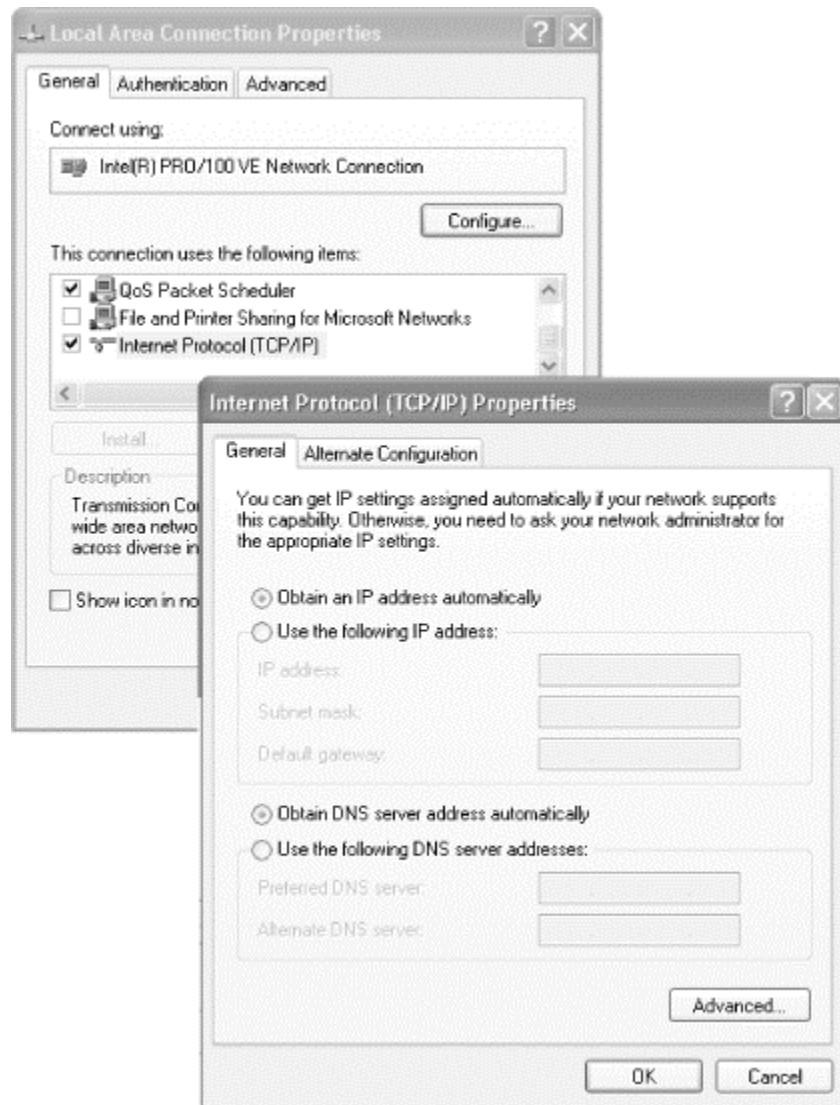
1. In the Control Panel, double-click Network Connections or Network and Dial-up Connections to launch the Network Connections dialog box, as shown in [Figure 2-3](#).

Figure 2-3. The Network and Dial-up Connections dialog box



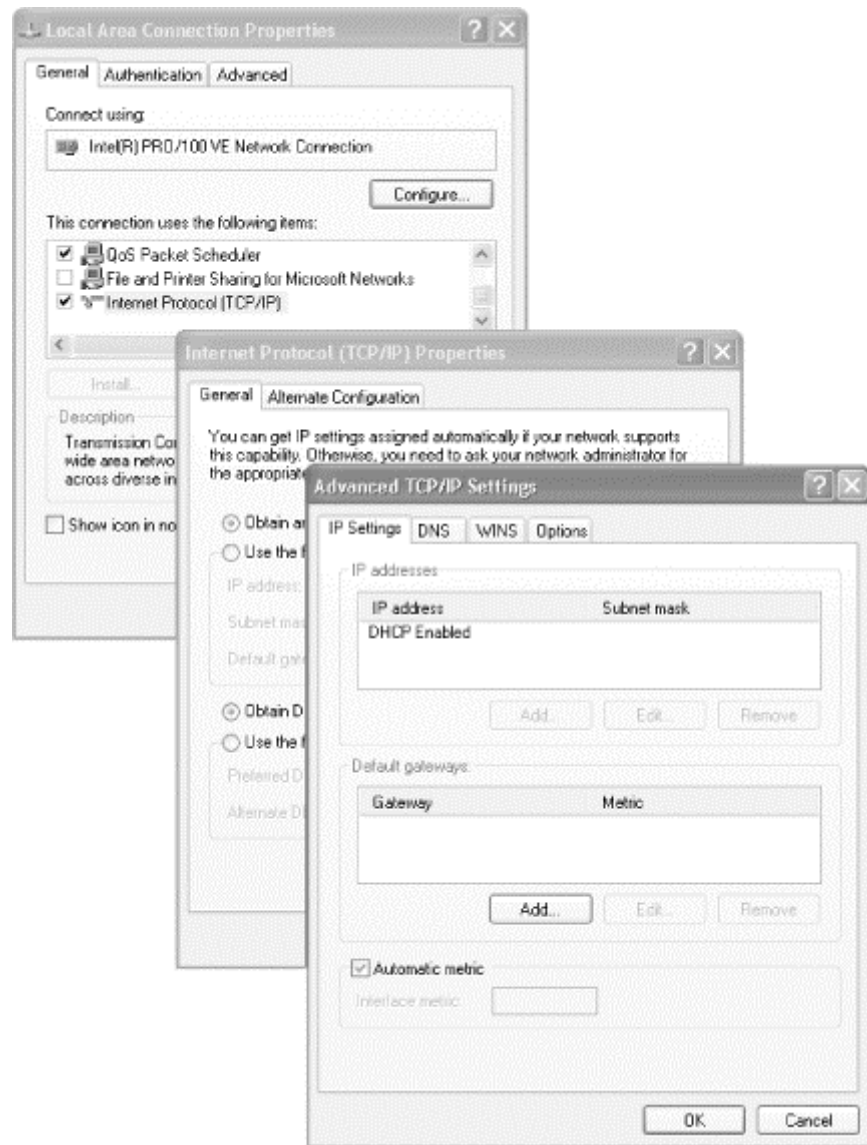
2. Double-click your Ethernet adapter and then the associated Internet Protocol (TCP/IP) entry, launching the Internet Protocol (TCP/IP) Properties dialog box, shown in [Figure 2-4](#). This dialog box tells you the IP address and subnet mask (netmask) of your system. If the "Obtain an IP address automatically" button is selected, the IP address and subnet mask will be blank. In that case, you don't need to be concerned about them because a DHCP server on your network supplies the network configuration automatically; Linux can obtain its network configuration from this same server.

Figure 2-4. The Internet Protocol (TCP/IP) Properties dialog box



3. Click the Advanced button and then the DNS tab of the Advanced TCP/IP Settings dialog box. This tab, shown as [Figure 2-5](#), provides the Host (hostname), Domain (domain name), and DNS Server information you'll need during installation.

Figure 2-5. The DNS configuration tab



4. Finally, click Cancel once to close the Advanced TCP/IP Settings dialog box and again to close the TCP/IP Properties dialog box. Clicking Cancel again closes the Network dialog box.

2.3 Preparing Your Hard Disk

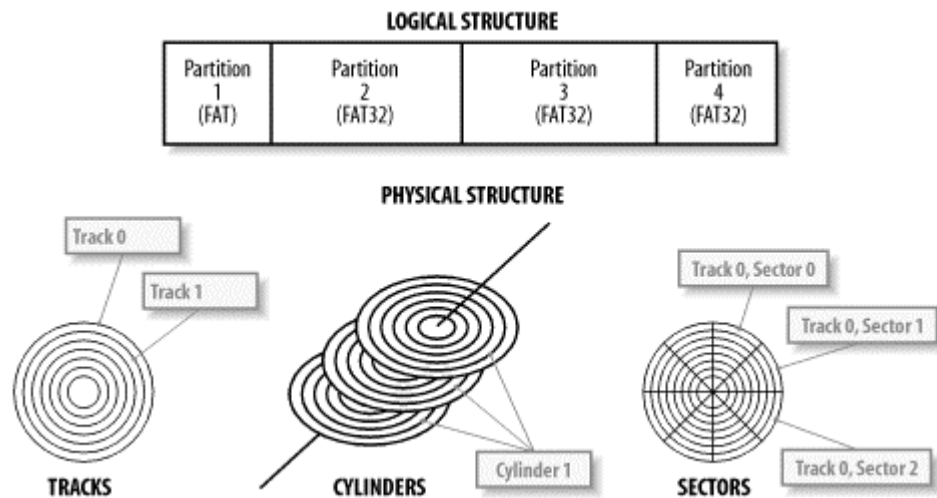
To prepare your hard disk for installing Linux, you must allocate the space in which Linux will reside. You'll learn how to do so in this section. First, I'll explain how hard disks are organized, followed by how to view the structure of a hard disk. Finally, I'll describe how to alter, or *partition*, the structure of your hard disk in preparation for installing Linux.

2.3.1 How Hard Disks Are Organized

Let's start by reviewing facts you've probably learned by working with Windows. Most operating systems, including Windows 95/98, 2000, and XP, manage hard drives by dividing their storage space into units known as *partitions*. So that you can access a partition, Windows associates a drive letter (such as C: or D:) with it. Before you can store data on a partition, you must *format* it. Formatting a partition organizes the associated space into what is called a *filesystem*, which provides space for storing the names and attributes of files as well as the data they contain. Windows supports several types of filesystems, such as FAT, FAT32, and NTFS.

Partitions comprise the *logical structure* of a disk drive, the way humans and most computer programs understand the structure. However, disk drives have an underlying *physical structure* that more closely resembles the actual structure of the hardware. [Figure 2-6](#) shows the logical and physical structure of a disk drive.

Figure 2-6. The structure of a hard disk



Mechanically, a hard disk is constructed of platters that resemble the phonograph records found in an old-fashioned jukebox. Each platter is associated with a read/write *head* that works much like the read/write head on a VCR, encoding data as a series of electromagnetic pulses. As the platter spins, the heads record data in concentric rings known as *tracks*, which are numbered beginning with zero. A hard disk may have hundreds or thousands of tracks.

All the tracks with the same radius are known as a *cylinder*. Like tracks, cylinders are numbered beginning with zero. The number of platters and cylinders of a drive determine the drive's *geometry*. Some PCs require you to specify the drive geometry in the BIOS setup. Most modern PCs autodetect the drive geometry but let you specify a custom value if you prefer.

Most operating systems prefer to read or write only part of a track, rather than an entire track. Consequently, tracks are divided into a series of *sectors*, each of which holds a fixed number of bytes, usually 512.

To correctly access a sector, a program needs to know the geometry of the drive. Because it's sometimes inconvenient to specify the geometry of a drive, some PC BIOS programs let you specify *logical byte addressing* (LBA). LBA sequentially numbers sectors, letting programs read or write a specified sector without the burden of specifying a cylinder or head number.

2.3.2 Viewing Disk Partitions

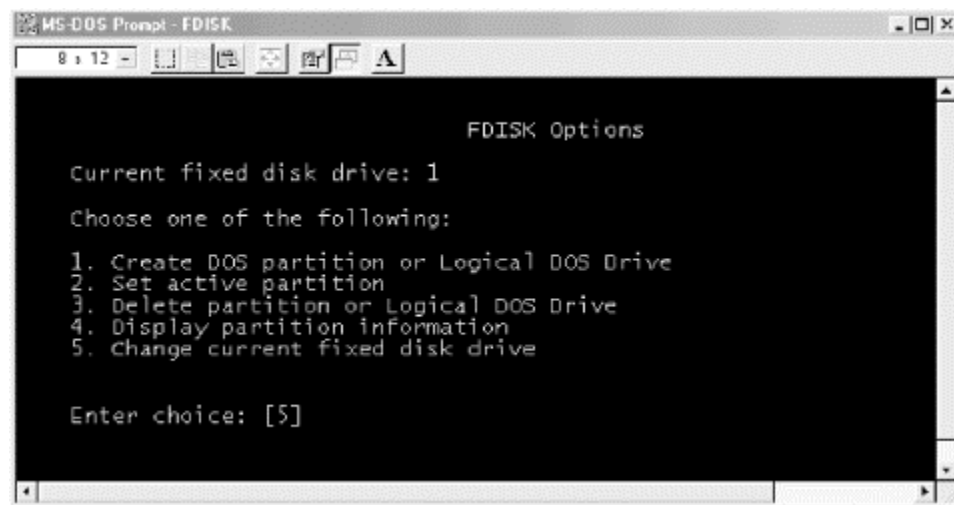
The first step in preparing your hard disk is viewing its partition information. Once you know how your hard disk is organized, you'll be able to determine how to reorganize it to accommodate Linux. To view the partitions that exist on your Windows 95/98 hard disk drives, you can use the **fdisk** utility. If your system runs Windows 2000 or XP, you must use the Disk Management tool, which resides within the Computer Management folder of the Administrative Tools control panel applet.

2.3.2.1 Using fdisk (Windows 95/98)

To use **fdisk**:

1. Click on the Windows Start menu. The Start pop-up menu appears.
2. Select Run. The Run dialog box appears.
3. Type "command" in the text box labeled Open. Then click OK or press Enter.
4. An MS-DOS Prompt window appears. Type **fdisk** and press **Enter**. The **fdisk** menu appears, as shown in [Figure 2-7](#).

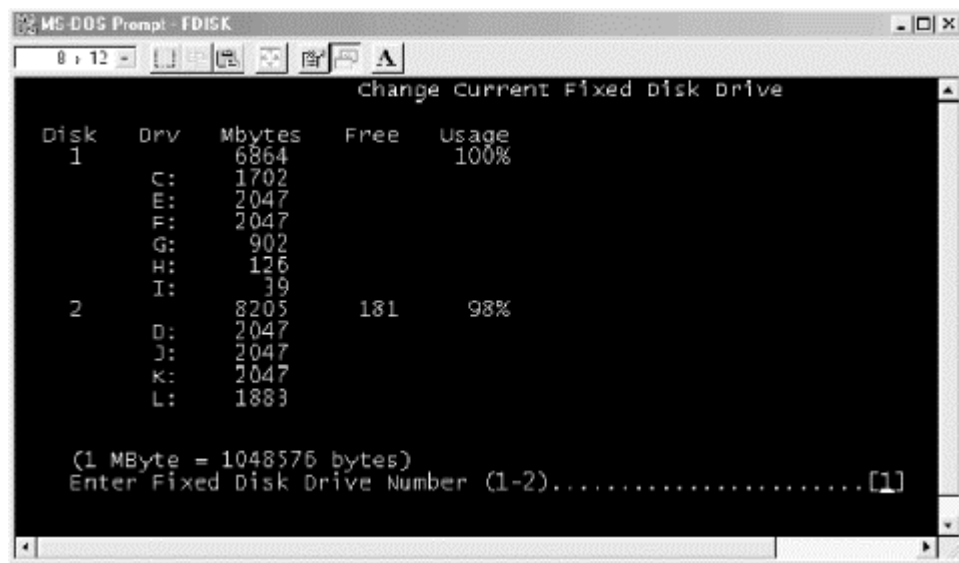
Figure 2-7. The fdisk Options screen



The **fdisk** menu may not appear immediately. Instead, Windows may ask if you want to enable large disk support; if this occurs, type **N** and press **Enter**. You don't need to enable large disk support to view partition information.

5. If your system has only one hard drive, you won't see option 5, titled "Change current fixed disk drive." If option 5 is available, type **5** and press **Enter**. This takes you to a screen resembling the one shown in [Figure 2-8](#) that lets you specify the current fixed disk drive.
6. If option 5 is not available, type the number associated with the "Display partition information" option and press **Enter**. The screen will resemble the one shown in [Figure 2-8](#), though its arrangement will be somewhat different.

Figure 2-8. The fdisk "Change current fixed disk drive" screen

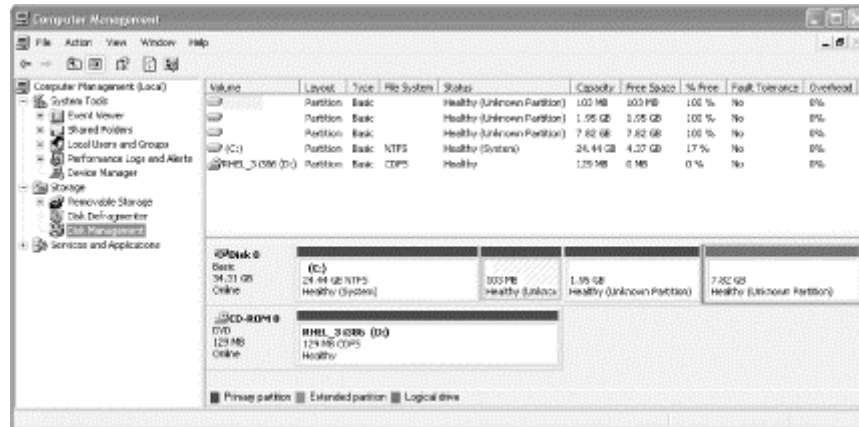


7. The screen shows each hard drive and its size, numbering the drives beginning with 1. If a drive contains free space not allocated to a partition, the screen shows the amount of space available. The screen also shows how much of the drive's space has been allocated to partitions, as a percentage of the total drive space.
8. Under the drive information section, the screen shows the size of each partition that resides on the drive. The screen also shows the associated drive letter, if any.
9. When you're done viewing partition information, press **Esc** once or twice to exit **fdisk** and return to the MS-DOS prompt. You can then close the MS-DOS Prompt window by clicking on the Close icon in the upper-right corner of the window or by typing **exit** and pressing **Enter**.

2.3.2.2 Using the Disk Management tool (Windows 2000/XP)

To use the Disk Management tool, click Start Control Panel Administrative Tools Disk Management. The tool graphically depicts your system's drives and the partitions and free space they contain, as shown in [Figure 2-9](#).

Figure 2-9. The Disk Management tool



2.3.3 Obtaining Sufficient Disk Space

Like Windows, Linux requires special partitions on which to store its filesystems and swap data. You must devote two—or preferably four—partitions to Linux. By viewing the partitions on your hard drive, you can determine which of the following two cases best describes your system:

- You have enough free (unpartitioned) disk space to accommodate Linux (4-10 GB, depending on the packages you want to install).
- In this case, make a note of the drive that holds the free disk space. You can then begin the installation process described in [Chapter 3](#). However, see the tip on PC BIOS limitations, later in this section.
- You lack enough free (unpartitioned) disk space to accommodate Linux.

If this is the case, you have several options:

- If your system has room for an additional disk drive, you can install a new drive and use it to hold Linux. The [Section 2.3.3.1](#) offers some considerations and tips on installing a new drive. This is generally the best option, because it sidesteps problems arising due to PC BIOS limitations.
- If you have one or more unused partitions, you can delete them and use the space you gain to hold Linux. The [Section 2.3.3.2](#) shows how to identify an unused partition.
- If you have one or more partitions that are larger than needed, you can shrink them and use the space you gain to hold Linux. The [Section 2.3.3.3](#) shows you how to determine whether a partition is larger than needed and how to free the excess space.

The BIOS of some PCs cannot access more than two hard drives and cannot access data beyond cylinder 1023 of a hard drive. In order to boot Linux, the installation program must create a 100-MB (or larger) boot partition (*/boot*) in an area accessible by the BIOS. If your

available free space does not satisfy these criteria, you must obtain additional free space as described in the following sections.

Linux supports LBA32, which can work around this problem. But some systems sold with advertised support for LBA32 do not actually support it. Moreover, enabling LBA32 support requires that you manually partition your system during installation. Therefore, I recommend that you partition your system as described in the following sections, to maximize the likelihood that it will work properly.

If you're unsure whether your free space satisfies these criteria, simply begin the installation; the installation program will notify you if it is unable to proceed. In that case, you can return to this chapter to learn how to gain or add additional disk space.

2.3.3.1 Installing a new disk drive

Often, the easiest way to install Linux is to install a new disk drive. If your system has only a single hard drive, you can probably install a second drive and place Linux on the new drive. Before obtaining a drive, you should make sure that the system provides room to mount the new drive and that you have the proper data and power cables. Be sure to install both disk drives on the primary disk controller so they can be booted; if you have an IDE CD-ROM drive, you should move it to the secondary controller.

If your system already has two disk drives, you probably can't simply add a third disk drive: the BIOS of most PCs lets you boot the system from only the first or second hard drive on the primary controller. In such a case, you can probably replace one of your existing drives with a larger drive adequate to support your existing needs and Linux.

2.3.3.2 Identifying an unused partition

You can use the drive letter information provided by **fdisk** to examine the contents of a partition in the Windows Explorer. If you can find a partition that holds no useful data but is large enough to accommodate the type of Linux installation you want, you can delete the partition and use the free space to hold Linux. Unless your system supports booting from outside the first 1023 cylinders of a hard drive, at least 100 MB of the unused partition should reside within the first 1023 cylinders of the drive; otherwise, you will have to use a boot floppy (see [Chapter 3](#)) to load Linux.

The easiest way to delete a partition is to start installing Red Hat from the installation media, and then request deletion of the partition when the installation program gives you the opportunity. Make note in [Table 2-1](#) of the partition you wish to delete and then begin the installation process described in [Chapter 3](#).

2.3.3.3 Shrinking a partition

Even if all of your partitions contain useful data, one or more partitions may be larger than required. In that case, you can reduce the size of each such partition and reorganize the drive to include contiguous unused space to hold Linux. Unless your system supports booting from outside the first 1023 cylinders of a hard drive, at least 100 MB of the unused space should reside within the first 1023 cylinders of the disk drive; otherwise, you'll have to use a boot floppy to load Linux.

You can use the Windows Explorer to determine the amount of free disk space in a partition. Simply right-click on the drive icon and click on Properties in the pop-up menu. The Properties dialog box shows the amount of used and free disk space associated with the drive.

If you are able to find one or more FAT or FAT32 partitions that have sufficient free space for a Linux installation, you can use a special utility to split the used and unused portions of a partition into separate partitions. Disc 1 of Linux includes the unsupported **fips** utility, which can split FAT and FAT32 partitions. For information on using **fips**, see the documentation in the *dosutils* directory of the CD-ROM.



If you make a mistake while attempting to shrink a partition or if the software malfunctions, you may lose all data in one or more partitions. You should not attempt to shrink a partition until you've completely backed up your system and made sure that your backup is usable.

Many Linux users find commercial partitioning tools—such as PowerQuest's *PartitionMagic* or VCOM's *Partition Commander*—helpful. Both tools are relatively inexpensive (approximately \$50-\$70) and support partition types and operations not supported by **fips**. For example, they can split NTFS and Linux *ext3* partitions. This is important, because you may not initially create Linux partitions of exactly the right size. Using **fips**, you'd be stuck; but, using PartitionMagic or Partition Commander, you can change your system's partition structure as many times as you like until you get it just right. For information on PartitionMagic, see the PowerQuest web site at <http://www.powerquest.com/partitionmagic>. For information on Partition Commander, see the VCOM web site at http://www.v-com.com/product/pc8_ind.html.



Use of PartitionMagic or Partition Commander is beyond the scope of this book. I mention them here because I believe that, while they aren't free, they are valuable, timesaving tools for partitioning your hard drive. Instructions on how to install and use either tool are included with the product documentation.

Chapter 3. Installing Linux

This chapter shows you how to install Red Hat Enterprise Linux or Fedora Core by following a simple, step-by-step procedure. During the installation, you'll need to refer to the information you collected in [Table 2-1](#) of [Chapter 2](#). Most users will be able to complete the installation procedure without difficulty; however, this chapter includes a section that describes how you can obtain help if you encounter installation problems. Once you successfully complete the installation procedure, you'll have your own working Linux system.

3.1 Installing the Operating System and Applications

To install Linux, attach any optional devices you plan to use, such as a PCMCIA network card or external floppy drive, to your PC. Then, follow this simple step-by-step procedure:

1. Start the installation.
2. Select installation options.
3. Create partitions.
4. Configure the boot loader.
5. Configure networking.
6. Configure language support.
7. Configure the system time.
8. Set the root password.
9. Select packages.
10. Install packages.
11. Configure X (Red Hat Enterprise Linux only).
12. Complete the video configuration (Red Hat Enterprise Linux only).
13. Create boot diskette (Fedora Core only)
14. Complete the installation.



Although the Linux installation procedure is generally trouble-free, errors or malfunctions that occur during the installation of an operating system can result in loss of data. You should not begin the installation procedure until you have backed up all data on your system and determined that your backup is error-free. You should also create boot floppies or other media that enable you to boot your system even if the boot information on the hard disk is damaged.

3.2 Start the Installation

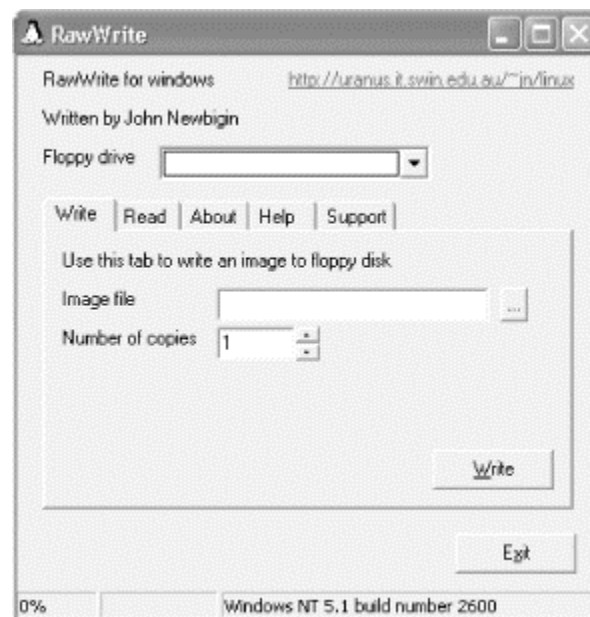
To begin installing Linux, you must boot your system from the installation media. Most recently manufactured PCs can boot the Installation CD 1 CD-ROM. However, unless you generally boot from a CD-ROM—which is quite unlikely—you'll need to reconfigure your PC's BIOS so your PC is able to boot from a CD-ROM. To do so, enter your PC's BIOS screen and look for a configuration item titled something like Boot Order or Boot Priority. Change the configuration so that the CD-ROM drive has the highest boot priority. Consult your PC's documentation for details on entering and using its BIOS configuration screens.

3.2.1 Creating a Boot Floppy

If your PC can't boot from a CD-ROM, you must create a boot floppy disk. Creating a boot floppy requires some special measures; you can't simply copy files onto a disk and then boot it. To create a Linux installation boot floppy by using a PC that runs Microsoft Windows, perform the following steps:

1. Format a floppy.
2. Insert Disc 1 of Linux into your system's CD-ROM drive.
3. Click **My Computer** and then your CD-ROM drive. Navigate to the `d:\dosutils\rawritewin` directory, where d is the drive letter associated with your CD-ROM drive. Double-click the program **rawritewin**. The RawWrite dialog box appears, as shown in [Figure 3-1](#). Specify the floppy drive and image file (images\bootdisk.img or other), and click **Write**. It takes perhaps a minute or so for the **rawrite** utility to create the floppy diskette.

Figure 3-1. Using rawrite to make a boot diskette



If your PC requires one or more PCMCIA or unusual SCSI devices during boot up, you must follow a somewhat more complicated procedure. See the *README* file on Installation CD-ROM 1 for details.

3.2.2 Boot the Installation Program

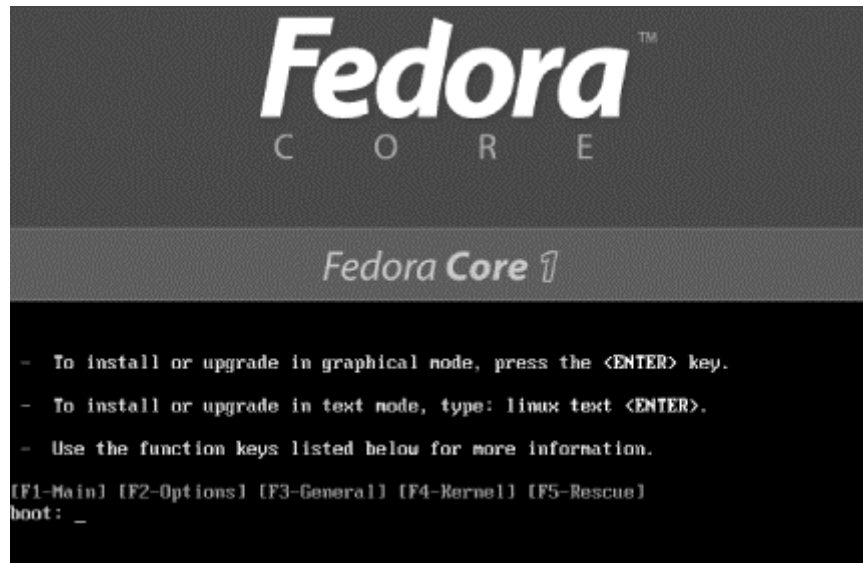
To start the installation process, insert Installation CD 1 of Linux into your system's CD-ROM drive. If your system cannot boot from a CD-ROM, insert the boot floppy you created and reboot your PC. When the system reboots, you should see a start-up screen featuring a **boot:** prompt and a series of messages explaining how to invoke the graphical

and text mode installation and upgrade facilities, as shown in [Figure 3-2](#). This prompt lets you enter special parameters to work around a variety of installation problems. Generally, it's not necessary to do so. Simply press **Enter** or wait about a minute and the installation program will start.



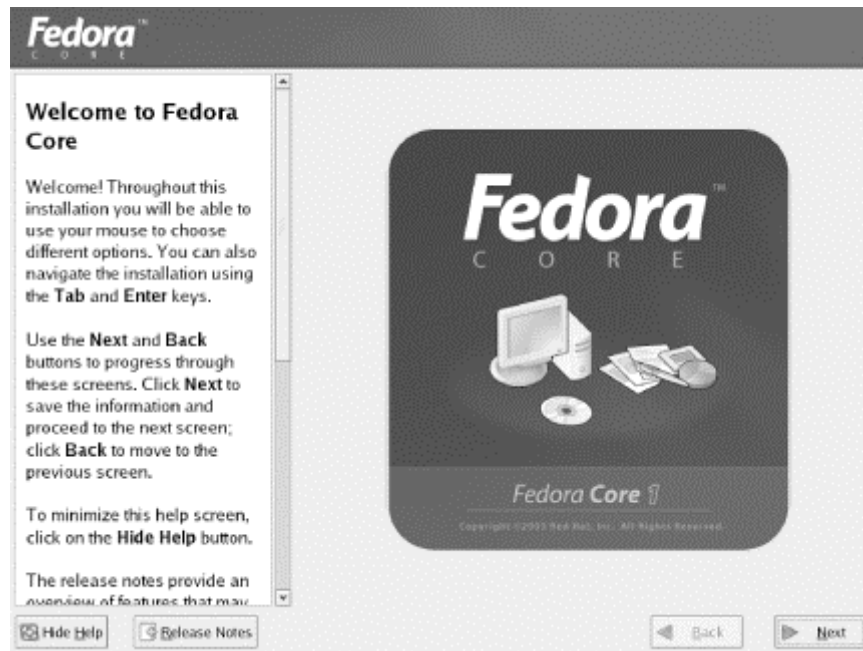
[Figure 3-2](#) shows the Fedora Core boot screen. If you're installing Red Hat Enterprise Linux WS 3, your screen looks a bit different, but works the same as the Fedora Core screen.

Figure 3-2. The Fedora Core boot screen



For perhaps a minute, you'll see text flashing by as the system boots. Then, you'll see the Linux Welcome screen, shown in [Figure 3-3](#). Click Next to proceed.

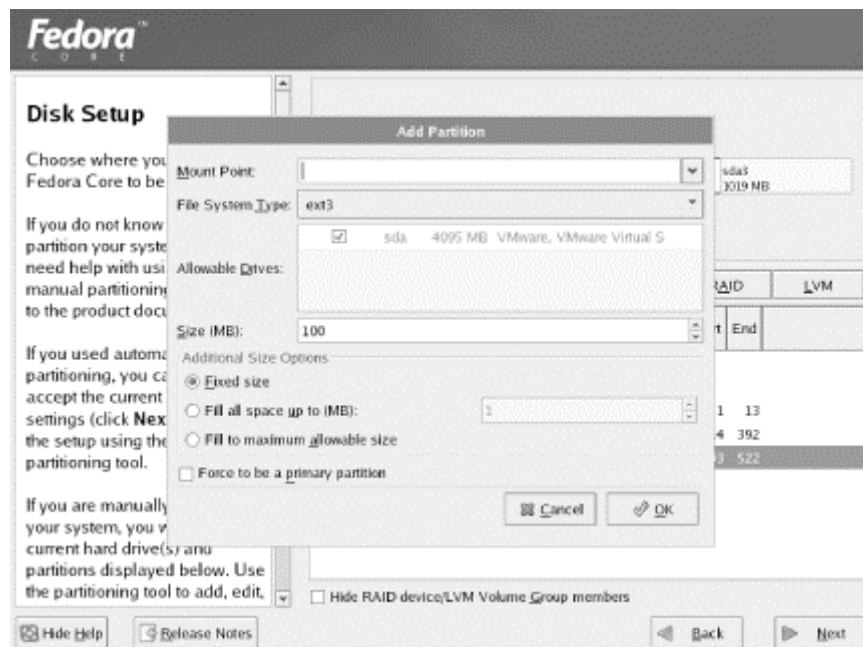
Figure 3-3. The Fedora Core Welcome screen



3.2.3 The Installation Program's User Interface

Like other modern Linux distributions, Linux includes a graphical installation program that simplifies the installation and initial configuration of Linux. [Figure 3-4](#) shows a typical screen displayed by the installation program. You won't see this particular screen until later in the installation process.

Figure 3-4. A typical installation screen



The installation screen includes the following elements:

A main window

The installation program runs in a full screen window that contains one or more child windows within it. The upper-left corner of each child window displays the name of the window. You cannot minimize or change the size of the installation program's main window.

The cursor

The installation program also has an on-screen cursor that resembles a small arrow pointing up and slightly to the left. The cursor is not visible in [Figure 3-4](#). When you position the cursor above a control and click the mouse, you set the input focus to the selected control. At any time, exactly one control has the input focus, which lets it respond to keyboard input. The control that has the input focus has a dotted rectangle outlining it or has a lighter color than otherwise similar controls.

Dialog boxes

The installation program uses dialog boxes to obtain user input. In [Figure 3-4](#), a dialog box titled Add Partition is visible. You can recognize it by the controls it contains, such as Mount Point and Size (MB). You dismiss a dialog box by clicking its OK or Cancel button. You cannot minimize an installation dialog box.

Buttons

When you click a button, the installation program performs a corresponding action. For example, clicking the OK button of a dialog box tells the installation program to accept the dialog box contents and proceed to the next step. Similarly, clicking the Cancel button of a dialog box tells the installation program to ignore the dialog box contents. Many installation screens include a helpful Back button that lets you return to the previous installation step. Most installation screens include a Next button that takes you to the next installation step.

Text boxes

Text boxes let you type text that is sent to the installation program when you dismiss the dialog box by using the OK button. In [Figure 3-4](#), the field labeled Size (MB) is a text box.

Checkboxes and radiobuttons

Checkboxes and radiobuttons let you specify that an option is enabled or disabled or select a specific option from a list. A dark area indicates an enabled option; a light area indicates a disabled option. You can click a checkbox or radiobutton to toggle the option between its enabled and disabled states. In [Figure 3-4](#), the control labeled "Force to be a primary partition" is a checkbox. The control labeled "Fixed size" is a radiobutton.

List boxes

List boxes let you choose an item from a predefined list. Two kinds of list boxes are used. One kind, called a drop-down list box, displays only a single item at a time. In [Figure 3-4](#), the control labeled Mount Point is a drop-down list box. If you click the checkmark that appears at the right of a drop-down list box, other items in the list become visible. When you select an item by clicking it, the list reverts to its original form, showing only the selected item.

The second kind of list shows multiple items simultaneously. If this kind of list box has more items than can be shown, it will have an associated scrollbar that lets you page through the list. The selected item, if any, is indicated by the item's dark background.

Online Help

The Online Help panel lets you view information that helps you understand what the current installation screen does and how to use it. If you don't understand the installation procedure or if you're curious to learn more, read the information in the Online Help panel.

Graphical Install . . . What Graphical Install?

If you don't see the Linux Welcome screen but instead see a screen with red and blue text areas over a black background, your system is not compatible with the Linux graphical install. Perhaps your system lacks sufficient RAM or has an unsupported video adapter.

In that case, you can use a text-based installation procedure. Because this special installation procedure is text-based, you won't be able to use a mouse. Instead, use **Tab** to move from field to field, use **Space** to select fields, and press **Enter** to click a selected button.

For additional help with text-based installation procedures, see Section 3.2 of *The Red Hat Enterprise Linux Installation Guide*, available at <http://www.redhat.com/docs>.

3.2.4 Use Virtual Consoles to Monitor the Installation

A console is a combination of a keyboard and a display device, such as a video monitor. A console provides a basic user interface adequate to communicate with a computer: you can type characters on the keyboard and view text on the display device.

Although a home computer system seldom has more than one console, Linux systems provide several virtual consoles. By pressing a special combination of keys, you can control which console your system's keyboard and monitor are connected to.

[Table 3-1](#) describes the virtual consoles used by the installation program. The main installation dialog appears in virtual console #7. If you like, you can use the indicated keystrokes to view a different virtual console.

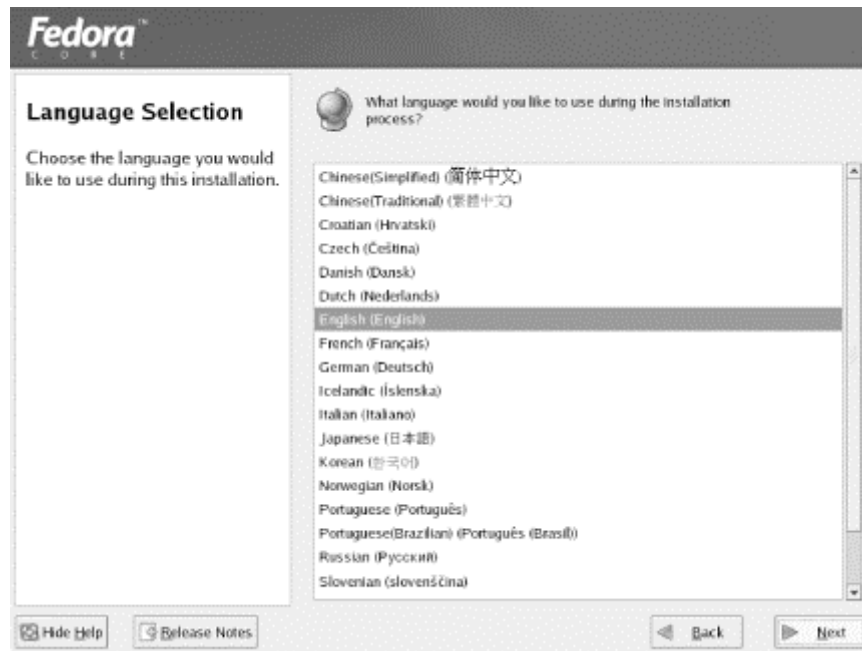
Table 3-1. Virtual consoles used by Red Hat's installation program		
Console	Keystroke	Contents
1	Ctrl-Alt-F1	Installation dialog
2	Ctrl-Alt-F2	A shell prompt that lets you enter commands to be processed by Linux
3	Ctrl-Alt-F3	The installation log, containing messages from the install program
4	Ctrl-Alt-F4	The system log, containing messages from the Linux kernel and other system programs
5	Ctrl-Alt-F5	May contain other messages, including those concerning the creation of filesystems
7	Ctrl-Alt-F7	The graphical window, which is the main window used by the installation program

You will not generally need to switch from one virtual console to another during installation. Nevertheless, you may find it interesting to view the contents of the virtual consoles. The contents of virtual consoles #1 through #5 can be useful in monitoring and troubleshooting.

3.2.5 Choose the Installation Language

Click Next to move from the Installation screen to the Language Selection screen. [Figure 3-5](#) shows that screen, which asks you to specify what language should be used during the installation process. Click the desired language and then click Next. The Keyboard Configuration screen appears.

Figure 3-5. The Language Selection screen



3.2.6 Select the Keyboard Type

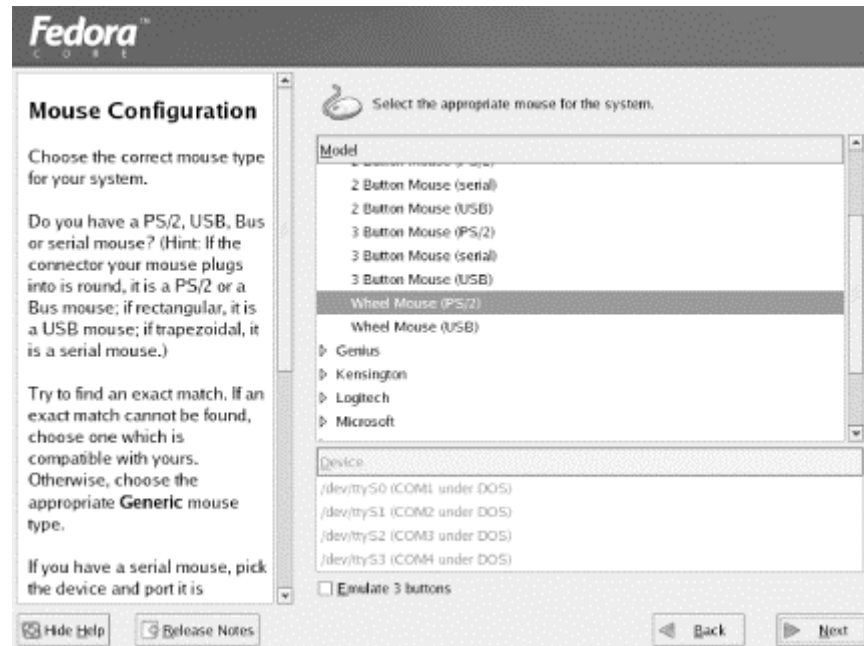
The Keyboard Configuration screen, shown in [Figure 3-6](#), lets you specify the type of keyboard attached to your system. The preselected choice is appropriate for most U.S. users. If you prefer another keyboard configuration, click the desired model or layout. Then, click Next to proceed. The Mouse Configuration screen appears.

Figure 3-6. .The Keyboard Configuration dialog box



The Mouse Configuration screen, shown in [Figure 3-7](#), lets you specify the type of mouse attached to your system. The installation program generally determines the type of mouse automatically. If you prefer a different mouse configuration, click the desired mouse type. Many graphical Linux programs are designed to use a three-button mouse. If your mouse has only two buttons, you should generally enable the Emulate 3 Buttons checkbox. Click Next to proceed to the Installation Type screen.

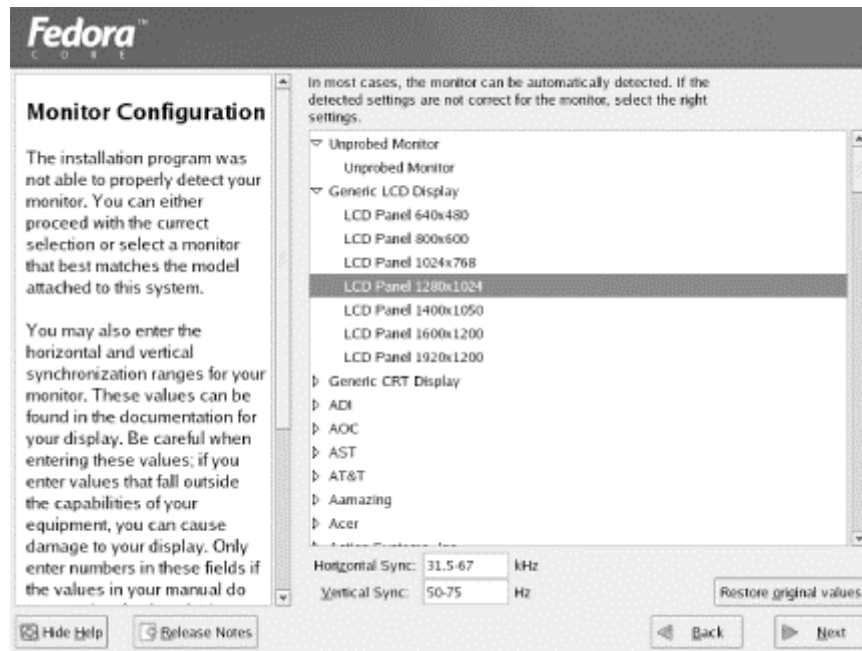
Figure 3-7. The Mouse Configuration screen



3.2.7 Select Monitor Configuration

If you're installing Fedora Core, the installation program prompts you to select the monitor type, as shown in [Figure 3-8](#). If you're installing Red Hat Enterprise Linux, you should skip ahead to [Section 3.2.9](#).

Figure 3-8. The Monitor Configuration screen



The Monitor Configuration screen includes a list of supported video monitors. If the installation program was able to determine the type of monitor associated with your PC, it will highlight the appropriate item in the Monitor Configuration list box. If you prefer to specify a different monitor, click the desired item.



Don't select a monitor that has an identifier merely similar to that of your monitor. Similarly identified models often have quite different characteristics.

Failing to select the appropriate monitor may result in permanent damage to your monitor, particularly if your monitor is an older, fixed-frequency model.

If your monitor displays a scrambled image, turn it off promptly and recheck your configuration.

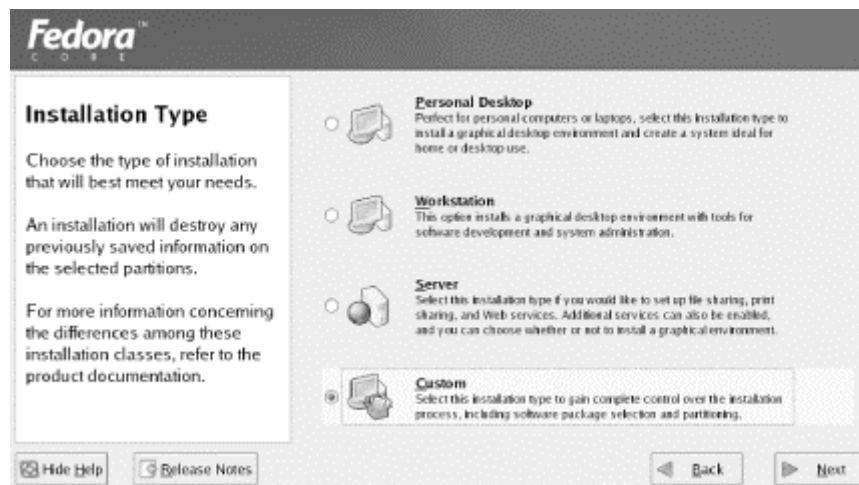
If you can't find your monitor listed, don't despair: you can select the Unprobed Monitor entry. If you do so, the installation program will suggest horizontal and vertical sync (also known as *vertical refresh*) rates or ranges. You should compare these with the characteristics of your monitor (which you can generally obtain from the owner's manual or from the manufacturer's web site) and adjust the rates or ranges if necessary. If you fail to find information describing your monitor, you can try some conservative values that are unlikely to damage all but the oldest of monitors. Low values are safer than high values. For example, try setting the horizontal sync range to 50-70 kHz and the vertical sync rate to 60 Hz.

Once you've selected your monitor or specified its sync rates, click Next to proceed. The Installation Type screen appears.

3.2.8 Select Installation Type

If you're installing Fedora Core, the installation program prompts you to select the installation type, as shown in [Figure 3-9](#). If you're installing Red Hat Enterprise Linux, you should skip ahead to [Section 3.2.9](#).

Figure 3-9. The Installation Type screen



The Installation Type screen lets you choose from four types of installation: *Personal Desktop*, *Workstation*, *Server*, and *Custom*.

3.2.8.1 Personal Desktop

If you're new to Linux, the Personal Desktop installation type is the easiest to perform, especially if you currently run Windows. In that case, the procedure will automatically configure your system to dual boot—in other words, whenever you start your system, a Linux utility known as GRUB will give you the choice of starting Windows or Linux. Both operating systems can reside on a single system as long as you have a large enough hard drive. A typical Personal Desktop installation requires about 2 GB of free disk space. However, 4 GB or more is a better working figure, as optional applications and extra packages can consume significant space beyond the minimum.



Even though the Personal Desktop installation type is generally the easiest, I recommend that you choose the Custom installation type, which is explained later. The Custom installation type is more flexible and therefore better able to help you cope with problems that may arise during installation.

3.2.8.2 Workstation

The Workstation installation type is based on the Personal Desktop installation type, but adds tools useful to software developers and system administrators. Like the Personal Desktop installation type, the Workstation Installation type requires 2-4 GB of free disk space.

3.2.8.3 Server

The Server installation type is appropriate for systems that will host a web server or other services. It does not include a GUI, so it's not suitable for desktop use. You shouldn't set up a system using the Server installation type until you've had significant experience with Red Hat Linux. A typical Server installation requires about 1 GB or more of free disk space.



The Server installation type *destroys all data* on your hard drive, including any existing Windows and non-Windows partitions. Do not perform a Server installation if you want to preserve *any* data on your system.

3.2.8.4 Custom

The Custom installation type gives you complete control over the installation process. You can specify whether to configure your system for dual booting, which software packages to install, and so on.

To perform a Custom installation, you should have from 520 MB to 5.3 GB of free disk space available. However, 520 MB is an absolute minimum, and 5.3 GB is needed only if you're planning to install everything (including the kitchen sink). More realistically, you should have at least 2 GB of free space available. If you have the expertise and patience, you can omit certain packages that would otherwise be installed during a Custom installation so that your Linux system occupies less disk space.

The appropriate choice for most users is Custom. Therefore, click the radiobutton next to Custom and click Next to proceed.

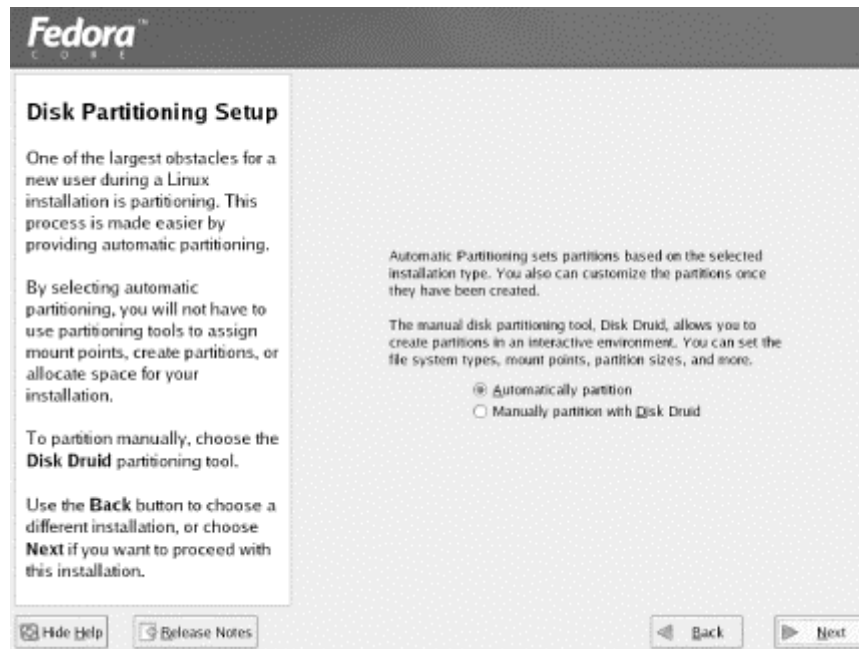


The step-by-step procedure given in this chapter describes only the Custom installation type. The procedures required for other installation types are similar and you can probably complete a non-Custom installation with the help of this chapter.

3.2.9 Create Partitions

Druids in Linux are analogous to wizards in Microsoft Windows. In the next phase of the installation procedure, you'll use Red Hat's Disk Druid to establish Linux partitions on your hard disk drive (see [Figure 3-10](#)). Disk Druid is generally capable of automatically creating the necessary partitions.

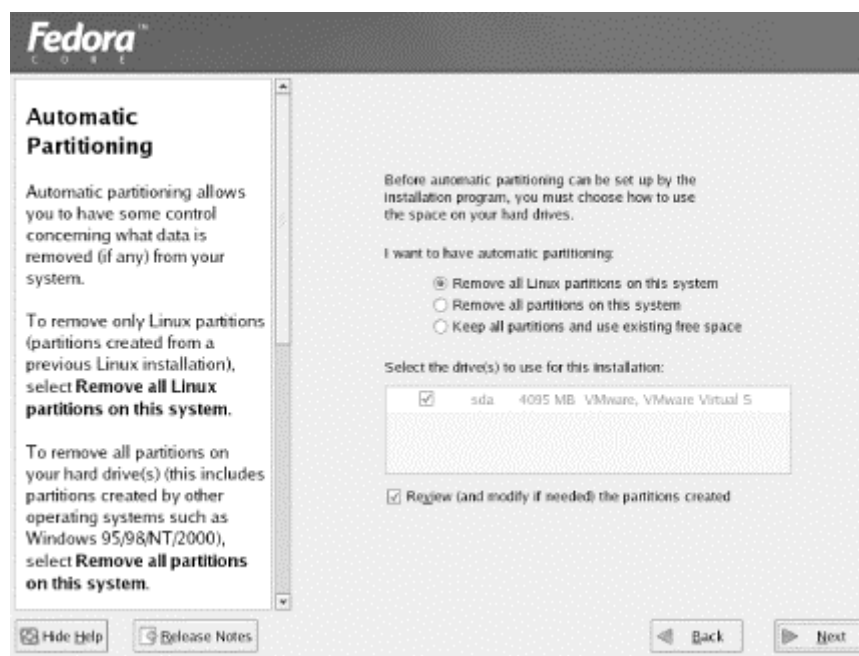
Figure 3-10. The Disk Partitioning Setup screen



If you prefer, the installation program lets you manually partition your hard disk using Disk Druid. However, it's generally easier to allow Disk Druid to automatically partition the hard disk and then review and edit the results. No changes are made to the partition table until you accept the final results.

You can generally accept the default choice, Automatically partition, and click Next to proceed. The Automatic Partitioning screen appears, as shown in [Figure 3-11](#).

Figure 3-11. The Automatic Partitioning screen





Before the Disk Partitioning Setup screen appears, a dialog box may appear, announcing that Disk Druid has found a problem with the partition table of one of your system's hard drives. The dialog box tells you how to resolve the problem. If the hard disk has never been used, its partition table won't be valid. In that case, you can continue the installation; Disk Druid will write an appropriate partition table to the disk.

Otherwise, you may need to restart the installation and specify the geometry of your hard drive in response to the **boot:** prompt. [Appendix C](#) describes the most common options. If you plan on erasing all the data on your hard drive, you can click Skip Drive and continue with the installation.

The Automatic Partitioning screen lets you choose how any existing partitions are treated. Three choices are available:

- Remove all Linux partitions on this system
- Remove all partitions on this system
- Keep all partitions and use existing free space

If you want to preserve an existing operating system, you should *not* choose Remove all partitions on this system. If you want to overwrite an existing Linux installation, choose Remove all Linux partitions on this system. Otherwise, choose Keep all partitions and use existing free space.

The Automatic Partitioning screen also lets you specify the hard disks on which the installation program will load Linux. You can prevent the installation program from using a hard disk by clearing the checkbox associated with the hard disk.

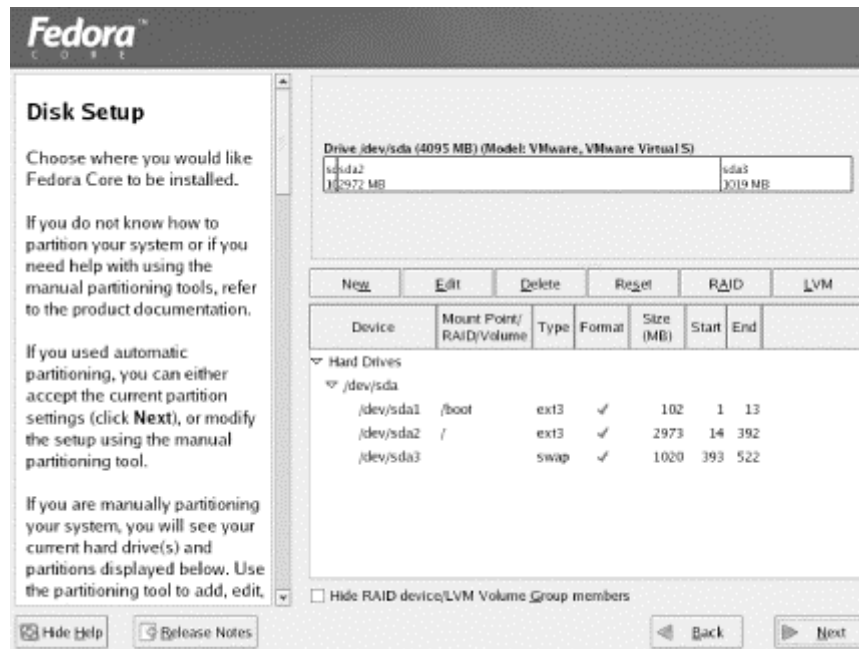
Finally, the Automatic Partitioning screen lets you specify whether to review the results of automatic partitioning. You should generally leave the associated checkbox set so that you can modify the results, if needed.

Click Next to perform automatic partitioning. If Disk Druid cannot find sufficient free space, you may see a dialog box announcing an error. In that case, you can use the Back button to return to the Disk Partitioning Setup screen and choose manual partitioning with Disk Druid. The following section explains how to review and modify the results of automatic partitioning, and also equips you to perform manual partitioning.

3.2.9.1 Manual disk partitioning

The Disk Setup screen, shown in [Figure 3-12](#), lets you add, edit, and delete Linux partitions. The bottom part of the screen contains a list box that describes each drive and partition. The middle part of the screen contains buttons that control the operation of Disk Druid, the tool that carries out partitioning. The top part of the screen graphically depicts the partition structure.

Figure 3-12. The Disk Druid screen



3.2.9.2 What to add

If you choose to perform automatic partitioning and are content with the results, you don't need to add or change any partitions. In that case, you can click Next to proceed. This section explains how to use manual partitioning if you need, or prefer, to do so.

Whereas Windows associates drive letters (such as **D:**) with partitions, Linux associates directories—known as mount points—with partitions. Two such directories, **/** (the root directory) and **/boot**, are essential. In addition, a third partition is necessary to manage your system's RAM. This partition has no associated mount point.

During this step of the installation process, you should establish three or four Linux partitions on your system's hard drive:

- A Linux swap partition to provide a work area used by Linux to efficiently manage your system's RAM memory. This partition should have a size two times the amount of your PC's installed RAM. However, the swap partition should not be smaller than 190 MB or larger than 2000 MB. This partition is not mounted by Linux and therefore has no associated mount point.
- A Linux native partition to hold the Linux kernel. This partition, which has the mount point **/boot**, should be at least 100 MB in size. However, there's no advantage to making it significantly larger than 100 MB.
- A Linux native partition to hold the Linux operating system. This partition, known as the root partition, has the mount point **/**. It should be as large as you can afford.
- Optionally, you can establish a **/var** partition. If you choose to do so, make the size of this partition 3 GB. The **/var** partition holds system files, such as logs, that grow in size during system operation. By establishing a **/var** partition, you protect your system against the results of an out-of-control process that uses up all space on the partition.

3.2.9.2.1 Create the swap partition

From the Disk Setup screen, click New to launch the Add Partition dialog box, shown in [Figure 3-13](#).

Figure 3-13. The Add Partition dialog box

The screenshot shows the 'Add Partition' dialog box with the following settings:

- Mount Point:** (Empty text field)
- File System Type:** ext3 (Dropdown menu)
- Allowable Drives:** sda 4095 MB VMware, VMware Virtual S (List box with one checked item)
- Size (MB):** 100 (Spin box)
- Additional Size Options:**
 - ☒ Fixed size
 - ☐ Fill all space up to (MB): 1 (Spin box)
 - ☐ Fill to maximum allowable size
- ☐ Force to be a primary partition
- Buttons:** Cancel, OK

Enter the following values in the Add Partition dialog box:

Mount Point

Leave this field blank.

Type

Select Swap.

Allowable Drives

Choose one or more hard disk drives on which to place the partition. If you select more than one hard disk drive, Disk Druid will choose a drive from among those you specify; Disk Druid will never create a partition that spans multiple disk drives.

Size

Specify twice the amount of RAM in your system. However, do not specify less than 190 MB or more than 2000 MB.

Additional Size Options

Specify Fixed size.

Force to be a primary partition

Set this checkbox if you want the swap partition to be a primary rather than logical partition. A primary partition is one that can be accessed by the BIOS and some old versions of operating systems. Because a hard disk can have only four primary partitions and the BIOS need not access the Linux swap partition, it's generally unwise to use a primary partition as a swap partition.

Click OK to accept the input values. The Add Partition dialog box disappears.

3.2.9.2.2 Create the /boot partition

From the Disk Setup screen, click New to launch the Add Partition dialog box. Enter the following values:

Mount Point

Select */boot* from the drop-down menu. The mount point specifies the directory name by which the partition will be known to Linux.

Partition type

Select ext3.

Size

Specify the size in megabytes of the */boot* partition, which should be 100 MB.

Allowable Drives

Choose one or more hard disk drives on which to place the partition. If you select more than one hard disk drive, Disk Druid will choose a drive from among those you specify; Disk Druid will never create a partition that spans multiple disk drives.

Additional Size Options

Specify Fixed size.

Force to be a primary partition

Set this checkbox if you want the */boot* partition to be a primary rather than logical partition. Because the */boot* partition must generally reside within the first 1024 cylinders of the hard disk and be accessible by the BIOS, it should generally be allocated as a primary partition.

Click OK to accept the input values; or, if you don't want to create the partition, click Cancel. The Add Partition Dialog box disappears.

If you enter an inappropriate value, Disk Druid may be unable to create the requested partition. In such a case, it displays a dialog box that explains the reason the partition could not be created. Study the dialog box to determine what you did wrong and try again.

3.2.9.2.3 Create the / (root) partition

From the Disk Setup screen, click New to launch the Add Partition dialog box. Enter the following values:

Mount Point

Select / from the drop-down menu.

Partition Type

Select ext3.

Allowable Drives

Choose one or more hard disk drives on which to place the partition.

Size

Specify the size in megabytes of the / partition, which should be at least 900 MB. More realistically, the size of the partition should be at least 3 GB (3000 MB).

Additional Size Options

Specify Fill to maximum allowable size, so the Linux native partition will be as large as possible. If you prefer to restrict the size of the partition, select Fill all space up to (MB) and specify the maximum desired size in the immediately following text box.

Force to be a primary partition

Set this checkbox if you want the / (root) partition to be a primary rather than logical partition.

Click OK to accept the input values. The Add Partition dialog box disappears.

3.2.9.2.4 Create the /var partition

If you choose to create a /var partition, click New to launch the Add Partition dialog box. Enter the following values:

Mount Point

Select /var from the drop-down menu.

Partition Type

Select ext3.

Allowable Drives

Choose one or more hard disk drives on which to place the partition.

Size

Specify the size in megabytes of the / partition, which should be 3 GB (3000 MB).

Additional Size Options

Specify Fixed size.

Force to be a primary partition

Set this checkbox if you want the /var partition to be a primary rather than logical partition.

Click OK to accept the input values. The Add Partition dialog box disappears.

3.2.9.3 Editing a partition

If you wish to change one or more values associated with a partition, highlight the partition you wish to change in the Disk Setup screen and click Edit. Disk Druid launches a dialog box that you can use to change the mount point of a previously existing partition or other options of a partition you've just created.

You cannot use the Add Partition dialog box to change the size, grow option, or type of a previously existing partition; instead, you must delete such a partition and re-create it.

3.2.9.4 Deleting a partition

If you wish to delete a partition, highlight it and click Delete. Disk Druid presents a dialog box that asks you to confirm the operation.



Deleting a partition destroys all the data it contains. Exercise great care to delete only unneeded partitions.

3.2.9.5 Starting over

If you determine that you've made mistakes and want to abandon the changes you've specified, simply click Reset. Disk Druid resets all partitions to their original state.

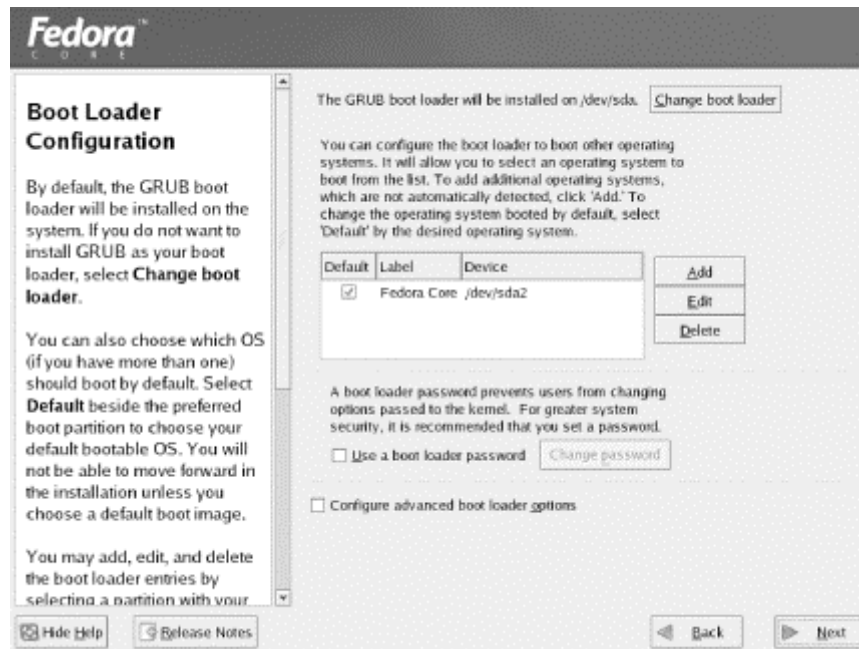
3.2.9.6 Saving your changes

When you're done, save your changes and proceed with the installation, by clicking Next.

3.2.10 Configure the Boot Loader

Next, the installation program presents the Boot Loader Configuration screen, shown in [Figure 3-14](#). GRUB, the Grand Unified Boot loader, is a special program used to start Linux—or another operating system—when you boot your system. This screen lets you choose an alternative boot loader (LILO) or omit installation of a boot loader altogether. You can also specify the location where GRUB will be installed.

Figure 3-14. The Boot Loader Configuration screen



Most Linux users install GRUB on their PC's primary hard drive. However, doing so doesn't always work out. For example, some antivirus applications detect GRUB's changes to the MBR and roll them back. The bottom line is that, for a few Linux users, GRUB can present some headaches. Moreover, LILO has some features useful to those responsible for servers, including better support for RAID and serial console devices. If you're among those bothered by GRUB, you can easily avoid it. To do so, you can use the Change Boot Loader Button to install the LILO boot manager. The installation program preselects Linux as the default operating system. To specify a different default operating system, highlight the corresponding partition and click Default.

If you like, you can change the label associated with an operating system by highlighting the corresponding partition and typing the desired label in the text box labeled Boot Label. When you've completely specified the desired boot loader configuration, click Next to proceed.

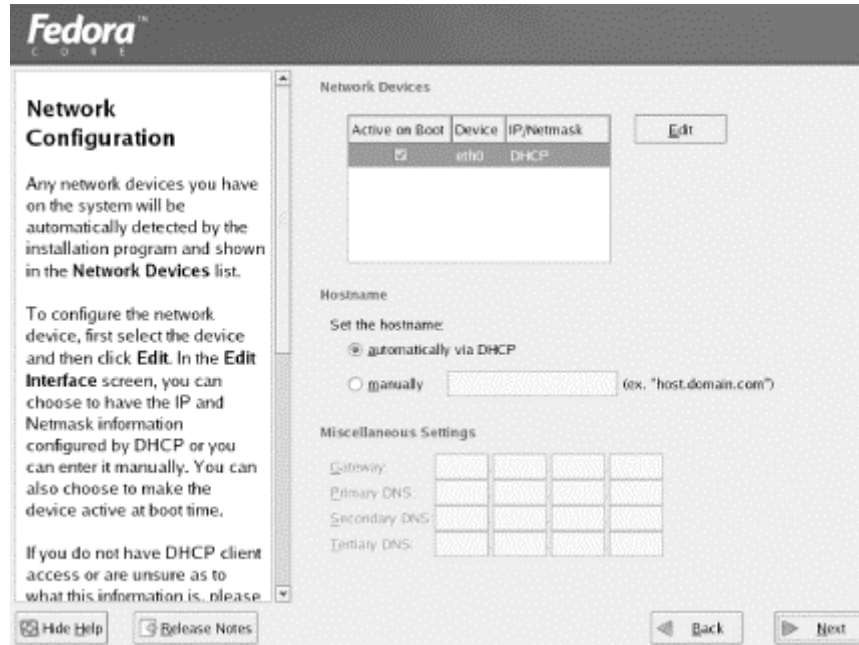


The Boot Loader Configuration screen also lets you specify a boot loader password. If you want to prevent unauthorized persons from using your PC, you can use a password to prevent someone from overriding security checks by passing special information to the Linux kernel. However, this level of protection is rarely necessary.

3.2.11 Configure Networking

After you've configured the boot loader, the installation program probes for a network card. If it finds one, the installation program presents the Network Configuration screen, shown in [Figure 3-15](#). If your computer is attached to a Local Area Network (LAN), you can use the Network Configuration screen to configure networking. If your computer is not attached to a LAN, click Next.

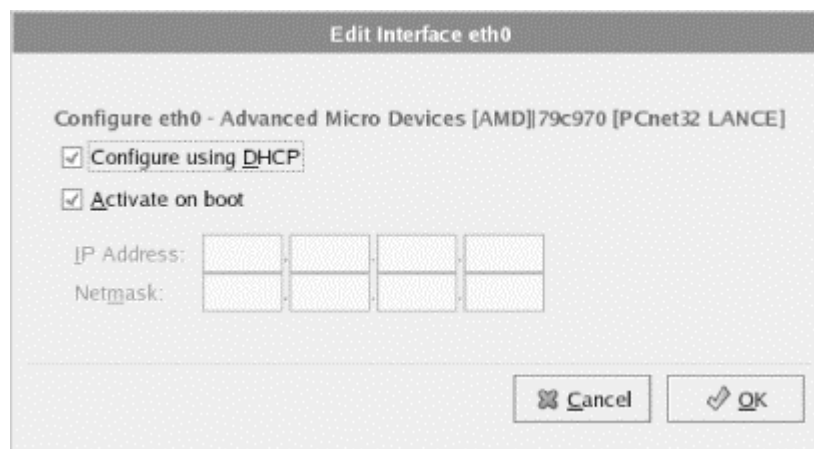
Figure 3-15. The Network Configuration screen



If your LAN provides a DHCP server, Linux can automatically determine your PC's network configuration when your PC boots. By default, the installer configures your PC to activate its network adapter and obtain its network configuration from a DHCP server whenever your PC is booted.

If you want to assign your PC a static IP address or if you don't want your network adapter to be activate when your system boots, click the Edit button. An Edit Interface dialog box, shown in [Figure 3-16](#), appears. To specify a static IP address, clear the checkbox labeled Configure using DHCP. Then, enter the IP address and netmask in the proper text fields, using the information you recorded in [Table 2-1](#). If you don't want your network adapter to be activated when your system boots, clear the checkbox labeled Active on boot. Click OK to save your changes.

Figure 3-16. The Edit Interface dialog box



If you want to assign your system a static IP address, you should select the radiobutton labeled "Set the hostname manually" in the Network Configuration Screen. You should also provide the following from [Table 2-1](#):

Hostname

The hostname of your system, including the domain name (for example, **newbie.redhat.com**)

Gateway

The host address of the router your system uses to send packets beyond its local network (for example, 192.168.1.1)

Primary DNS

The IP address of the system that provides hostname lookup services to your system (for example, 192.168.1.1)

Secondary DNS

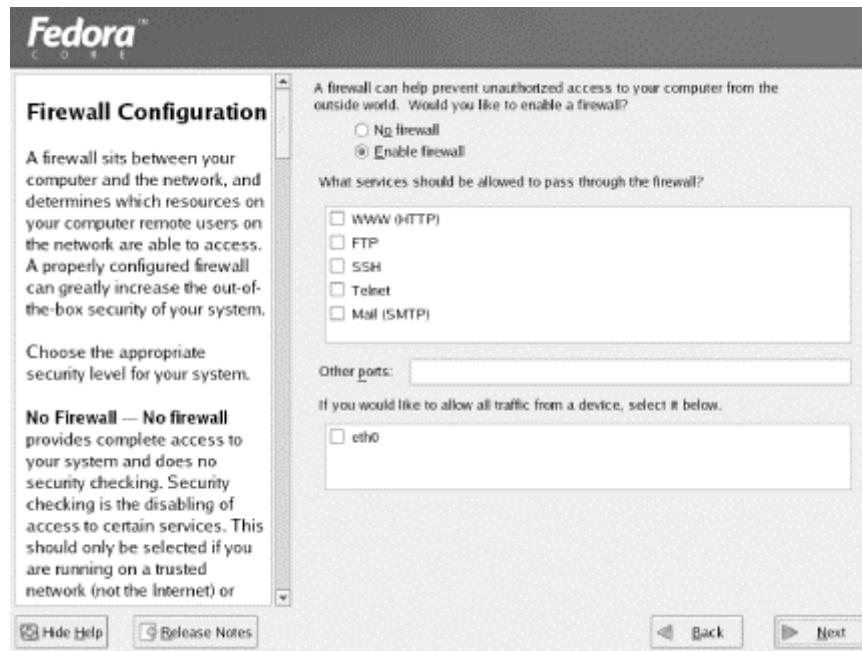
The IP address of the system used to look up hostnames if the primary name server is unavailable (optional)

Tertiary DNS

The host address of the system used to look up hostnames if the primary and secondary name servers are unavailable (optional)

When you've entered the desired network configuration, click Next to proceed. The Firewall Configuration screen, shown in [Figure 3-17](#), appears. This screen lets you specify protection against threats originating across the network.

Figure 3-17. The Firewall Configuration screen

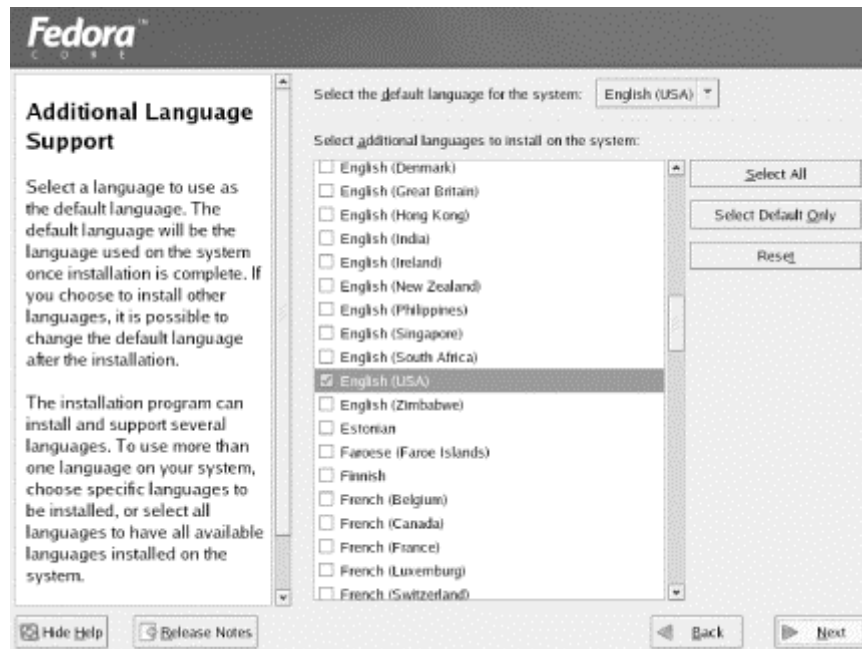


[Chapter 12](#) offers a detailed explanation of Linux's firewall capabilities. You should generally accept the default choice, Enable firewall, which prevents most network-based attacks on your system. If you plan to run a server, such as a web server, on your system, you'll need to allow the related traffic to pass through the firewall. However, you can easily open the proper ports at a later time. Click Next to proceed.

3.2.12 Configure the Language

Next, the installation program presents the Additional Language Support screen, shown in [Figure 3-18](#). You selected the language used earlier, during the installation procedure; this screen has a different function. It lets you specify the language or languages that you plan to use during system operation. It also lets you install support—including X fonts and spelling dictionaries—for these languages. As the screen explains, multiple languages consume significant disk space. So, select a single language if you're reasonably content doing so. After you've made your choice or choices, click Next to proceed.

Figure 3-18. The Additional Language Support screen



3.2.13 Configure the System Clock

After you bypass or complete the Network Configuration screen, the installation program presents the Time Zone Selection screen (Figure 3-19). Select a time zone by clicking on the map or by clicking an entry in the list box that appears below the map.

Figure 3-19. The Time Zone Selection screen

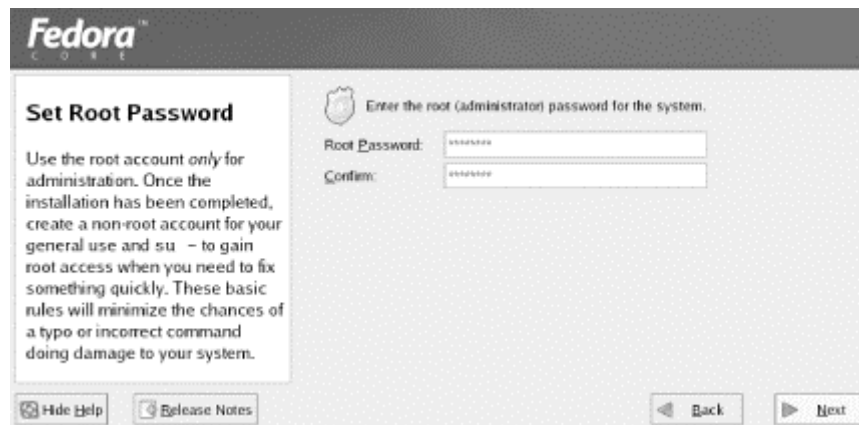


If you want to set your system's clock to UTC (Universal Coordinated Time), enable the System Clock Uses UTC checkbox. However, you should not enable this checkbox if your PC is set up to boot an operating system, such as Microsoft Windows 9x, that does not support setting the system clock to UTC. After making your selections, click Next to proceed.

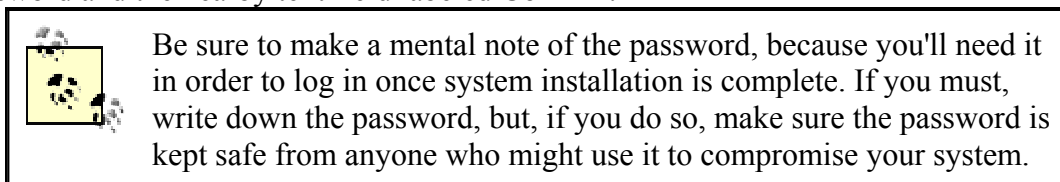
3.2.14 Set the root Password

The user who administers a Linux system is known as the root user, or simply *root*. To protect your system against mischief and misadventure, you must protect the root user's login with a password. To enable you to do so, the installation program presents the Account Configuration screen ([Figure 3-20](#)).

Figure 3-20. The Account Configuration screen



Simply choose a password for the root user and type it twice: in the text field labeled Root Password and the nearby text field labeled Confirm.



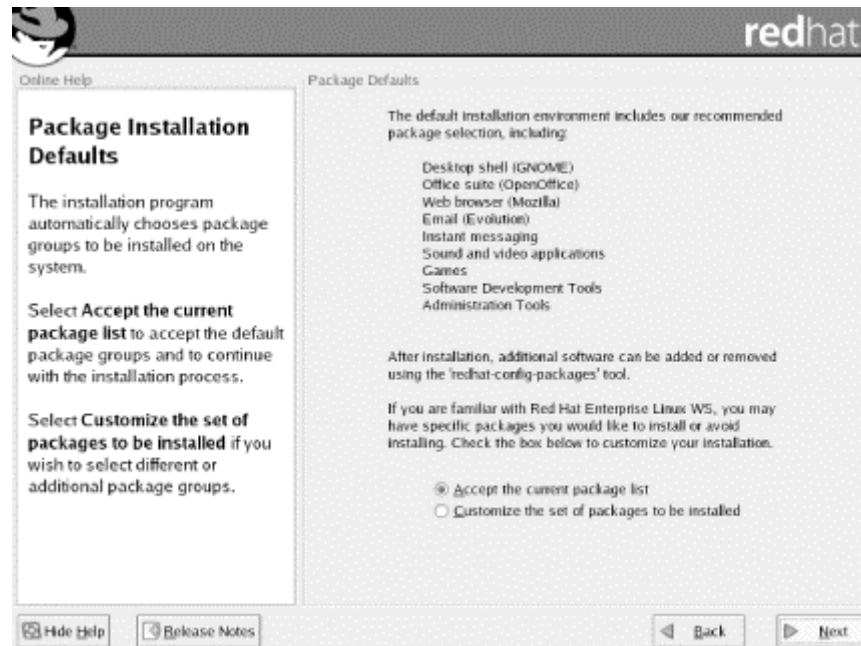
3.2.15 Select Packages

To install an application under Linux, you generally install a package that contains all the files needed by the application. If you like, you can specify the individual packages you want to install; however, the large number of available packages makes it tedious to specify them one at a time. Instead, the installation program lets you specify package groups you want to install. A package group is simply a group of related packages.

If you're installing Red Hat Enterprise Linux, the Package Installation Defaults screen, shown in [Figure 3-21](#), lets you accept a default set of packages or manually specify which packages should be installed. Generally, the default choices are appropriate. One common

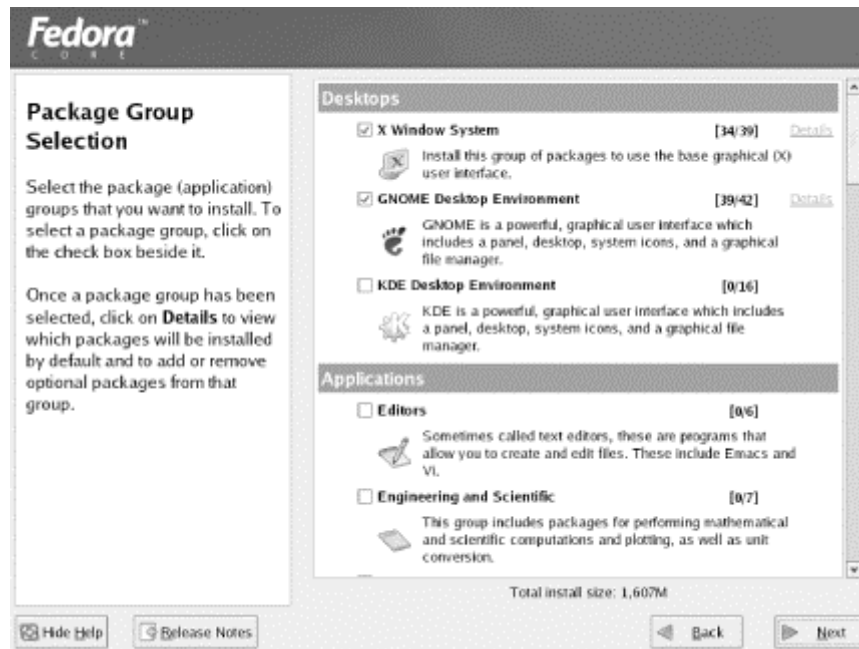
reason for manually specifying packages is if you prefer the KDE desktop shell. To see the full set of available packages, click the radiobutton labeled "Customize the set of packages to be installed" and then click Next.

Figure 3-21. The Package Installation Defaults screen



Whether you're installing Red Hat Enterprise Linux or Fedora Core, the Package Group Selection screen, shown in [Figure 3-22](#), appears. To install a package group, simply enable the associated checkbox. Some package groups let you customize the individual packages that will be installed. Such packages groups include a hyperlink labeled Details. Click the hyperlink to view a list of packages in the package group; enable or disable the checkbox associated with a package to specify whether the package should, or should not, be installed.

Figure 3-22. The Package Group Selection screen



The available package groups are:

X Window System

The graphical user interface provided by Linux. In addition to the X Window System, you should generally install either or both the GNOME Desktop or KDE Desktop.

GNOME Desktop

The default graphical desktop supported by Linux. You can use the alternative graphical desktop, the KDE Desktop, if you prefer. Both desktops can be installed on a system; an easy-to-use utility lets you switch between them.

KDE Desktop

The alternative graphical desktop supported by Linux.

Editors

Programs used to create and edit text files. [Chapter 7](#) explains how to use nano, a text-mode editor. The desktop also includes graphical text editors that work much like the Windows Notepad program.

Engineering and Scientific

Programs used to perform mathematical and scientific computations and to plot data.

Graphical Internet

Graphical programs for accessing the web, email, and other Internet services.

Text-Based Internet

Non-graphical programs for accessing the web, email, and other Internet services. Because these programs are non-graphical, they don't require the X Window System or a desktop.

Office/Productivity

Word processors and other programs useful for office work.

Sound and Video

Programs to hear audio, view video, and create audio/video files.

Authoring and Publishing

Programs for working with documents using the DocBook format.

Graphics

Programs for working with images.

Games and Entertainment

Programs that let you fill otherwise idle time.

Server Configuration Tools

Programs for graphically configuring servers.

Web Server

The Apache web server and related utilities.

Mail Server

Mail transfer agents and mail delivery agents, which are usually not needed by desktop computer users. Most users need only the mail user agents included in the Graphical Internet or Text-Based Internet Package Groups.

Windows File Server

Samba, a program that interoperates with Windows file and printer sharing, and related utilities.

DNS Server (Fedora Core 1 only)

BIND, a program that resolves host names to IP addresses and IP addresses to hostnames.

FTP Server (Fedora Core 1 only)

VSFTPD, a program that enables transfer of files between systems.

SQL Database Server and MySQL Database (Red Hat Enterprise Linux only)

The PostgreSQL and MySQL database management systems.

Legacy Network Server (Red Hat Enterprise Linux) or Network Servers (Fedora Core)

Miscellaneous network servers for obsolete and seldom-used protocols.

Development Tools

Programs useful to programmers and others working with programs.

Kernel Development

Files and programs needed to compile the Linux kernel.

X Software Development

Files and programs needed to compile programs that run under the X Window System.

GNOME Software Development

Files and programs needed to compile programs that run under the GNOME desktop.

KDE Software Development

Files and programs needed to compile programs that run under the KDE desktop.

Legacy Software Development (Red Hat Enterprise Linux only)

Packages that provide compatibility with previous releases of Linux.

Administration Tools

Programs to help you manage your system.

System Tools

Packages to help monitor and administer a system.

Printing Support

Programs to administer and use printers.

Everything

Selecting this package group causes installation of all packages. About 5 GB of disk space are needed in the / partition.

Minimal

Selecting this package group causes installation of a small set of packages that support basic operation of the system. You shouldn't generally select this package group.

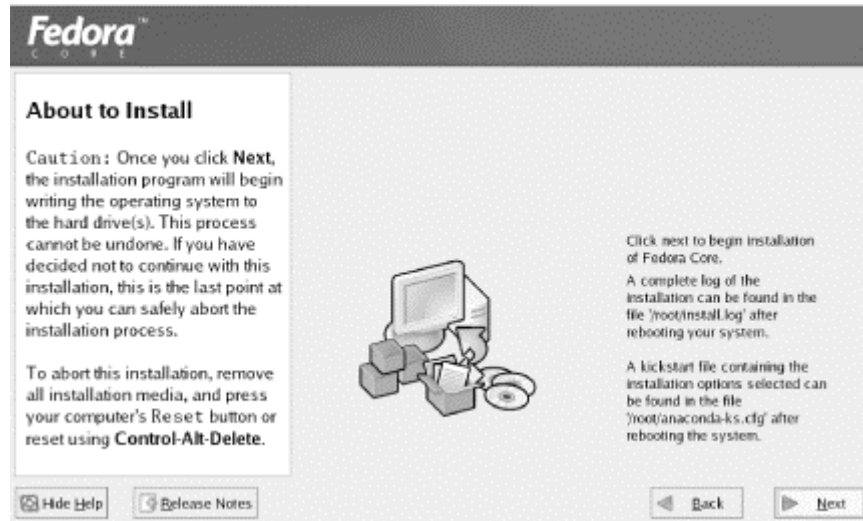
If you don't know what package groups to select, don't worry; you can install additional package groups after setting up your Linux system. When you're satisfied with your choices, click Next to proceed.

3.2.16 Install Packages

When the installation program is ready to begin installing packages, it presents the About to Install screen, shown in [Figure 3-23](#). Up to this point, the installation program has made no changes to your system's hard drive. This is your last chance to terminate the installation

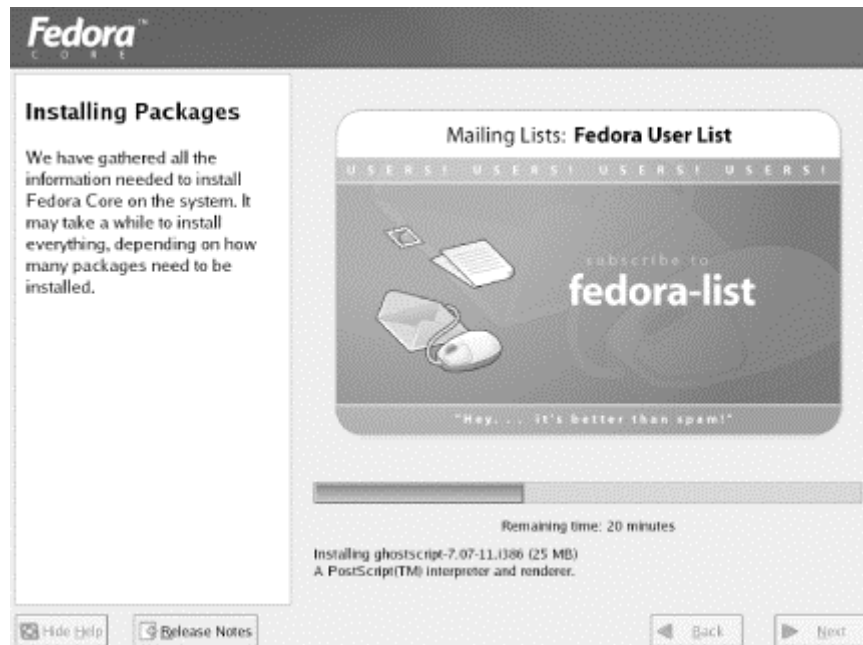
procedure before any data is written. To abort the installation procedure, press **Ctrl-Alt-Delete** or press your system's hardware reset button.

Figure 3-23. The About to Install screen



The installation program now formats any partitions you earlier specified for formatting. Depending on the size of your system's hard drive, it may require several minutes to complete this step. When formatting is done, the Installing Packages screen ([Figure 3-24](#)) appears and the installation program begins installing packages.

Figure 3-24. The Installing Packages screen

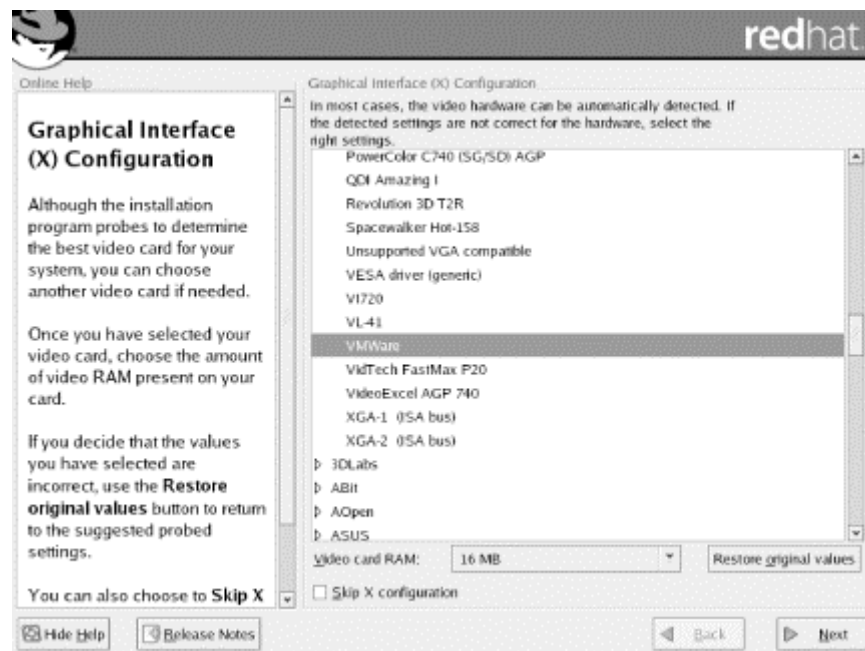


The Installing Packages screen displays the name of each package as it is installed and presents a progress bar that shows the relative progress of the installation process. You'll likely be asked to insert other installation CDs during the installation process. In particular, the installation program generally asks for CD 2 shortly after it begins installing packages. It asks you to re-insert CD 1 after installing the contents of the other CDs. When all the packages have been installed, you're ready to create a boot diskette. Click Next to proceed.

3.2.17 Configure Video

If you're installing Fedora Core, you've already selected your video monitor. You should skip ahead to [Section 3.2.18](#). If you're installing Red Hat Enterprise Linux, the installation program displays the Graphical Interface (X) Configuration screen ([Figure 3-25](#)), which helps you configure X, the Linux graphical user interface. If the installation program was able to determine the type of video card associated with your PC, it will highlight the appropriate item in the X Configuration list box. If you prefer to specify a different video card, click the desired item.

Figure 3-25. The Graphical Interface (X) Configuration screen



If your video card is not listed, you may be able to use the Generic SVGA Compatible entry. This will yield a basic, working X configuration. Later, you can follow the instructions given in [Appendix D](#) to achieve a better configuration. You should specify the amount of video memory installed on your video card. Specifying a value that is too large will probably prevent X from starting. If you followed the procedure given in [Chapter 2](#), you should have learned the amount of video memory installed on your card; otherwise, consult your video card's documentation to determine the proper value. If you can't locate the information, select a conservatively low value; choosing a value that's

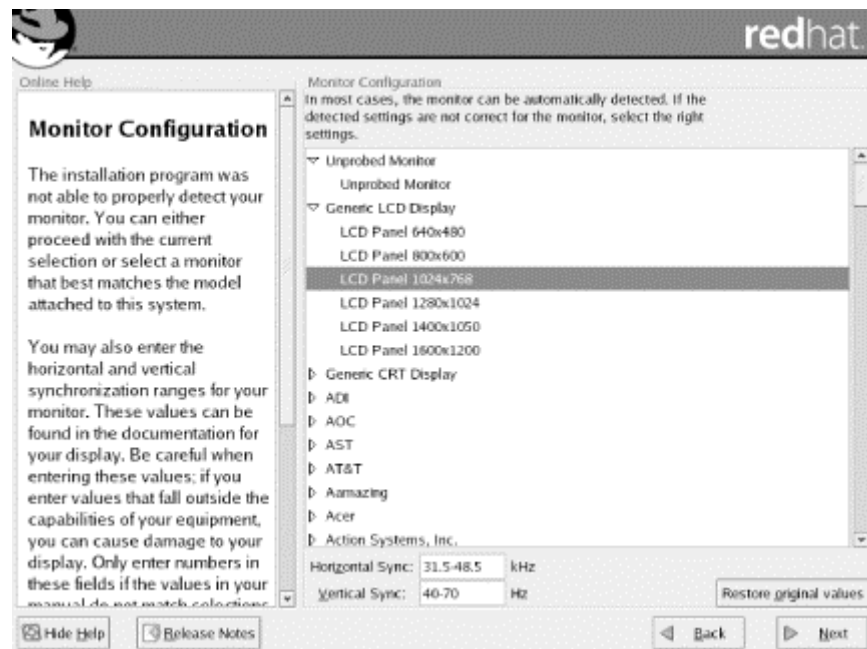
too low will prevent you from using high-resolution video modes but won't damage your monitor.

Earlier in the installation, you specified the make and model of your system's video card. The next two screens guide you in completing the video configuration.

3.2.17.1 Identifying your system's monitor

The installation program next presents the Monitor Configuration screen, shown in [Figure 3-26](#). The Monitor Configuration screen includes a list of supported video monitors. If the installation program was able to determine the type of monitor associated with your PC, it will highlight the appropriate item in the Monitor Configuration list box. If you prefer to specify a different monitor, click the desired item.

Figure 3-26. The Monitor Configuration screen



Don't select a monitor that has an identifier merely similar to that of your monitor. Similarly identified models often have quite different characteristics.

Failing to select the appropriate monitor may result in permanent damage to your monitor, particularly if your monitor is an older, fixed-frequency model.

If your monitor displays a scrambled image, turn it off promptly and recheck your configuration.

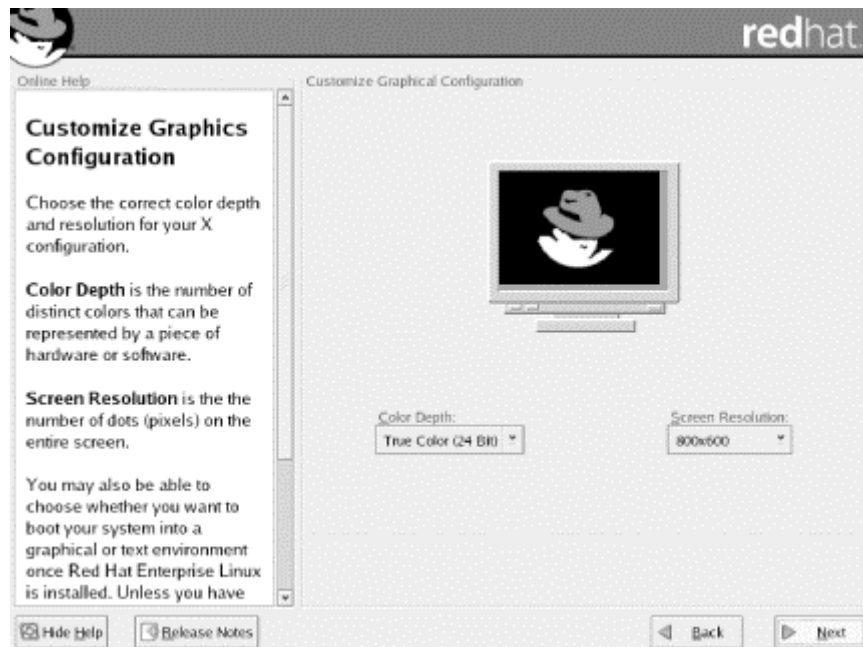
If you can't find your monitor listed, don't despair: you can select the Unprobed Monitor entry. If you do so, the installation program will suggest horizontal and vertical sync (also known as *vertical refresh*) rates or ranges. You should compare these with the characteristics of your monitor, which you can generally obtain from the owner's manual or from the manufacturer's web site; adjust the rates or ranges if necessary. If you fail to find information describing your monitor, you can try some conservative values that are unlikely to damage all but the oldest of monitors. Low values are safer than high values. For example, try setting the horizontal sync range to 50-70 kHz and the vertical sync rate to 60 Hz.

Once you've selected your monitor or specified its sync rates, click Next to proceed. The Custom Graphics Configuration screen appears.

3.2.17.2 Selecting custom graphics options

The Custom Graphics Configuration screen ([Figure 3-27](#)), lets you specify several X-related options. You can choose the color depth, which determines the number of colors your system will display. You can also choose the screen resolution, which determines the number of pixels your system will display. Larger resolutions result in greater detail; however, your system's video adapter and monitor may not operate with all possible settings. Moreover, specifying a high resolution will consume additional system resources, such as RAM. When you've completed the X configuration, click Next to proceed.

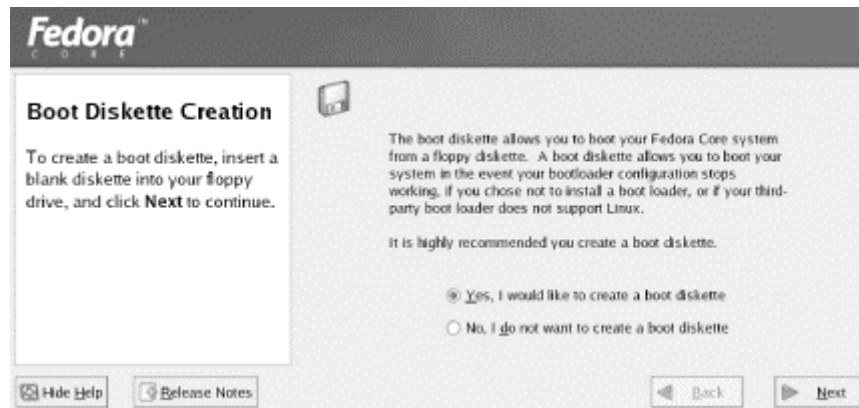
Figure 3-27. The Custom Graphics Configuration screen



3.2.18 Boot Disk Creation

If you're installing Fedora Core, the Boot Disk Creation screen ([Figure 3-28](#)) appears. If you're installing Red Hat Enterprise Linux, the installation program doesn't prompt you to create a boot disk; you should skip ahead to [Section 3.2.19](#).

Figure 3-28. Boot Disk Creation screen



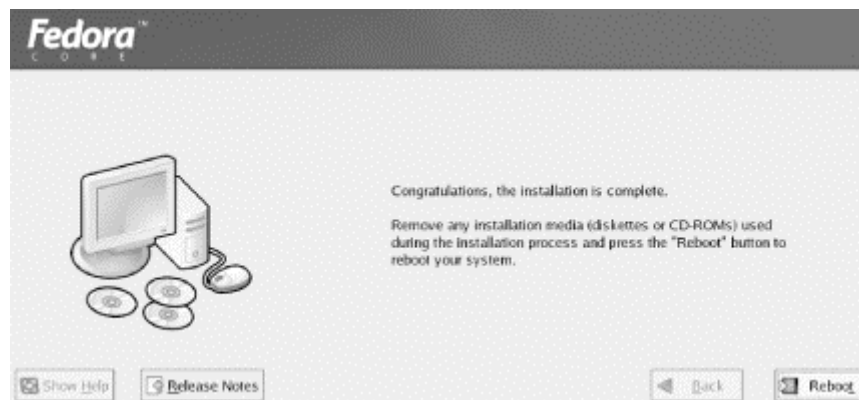
The floppy disk you create using this screen is not the same as the one you may have created and used to start the installation procedure. You should take the opportunity to create the boot disk, because it may enable you to boot your Linux system even if the boot loader fails to install properly or the system boot information is damaged.

By default, the option to create a boot disk is enabled. To create a boot disk, click Next. The installation program will prompt you to insert a blank floppy in your system's floppy drive. Insert the diskette and click OK. The installation program may take several minutes to create the boot floppy.

3.2.19 Complete the Installation

The installation program then presents its final screen ([Figure 3-29](#)), which explains that Linux has been successfully installed.

Figure 3-29. The Congratulations screen



Try booting your system from its hard drive or a Linux boot floppy, as appropriate. If your system successfully boots Linux, you're ready to log in to your Linux system.

3.3 The Firstboot Service

When you boot your system for the first time, the Firstboot service is launched to help you configure your system. [Figure 3-30](#) shows the Welcome screen the service displays. Click Next to continue.

Figure 3-30. The Firstboot service Welcome screen



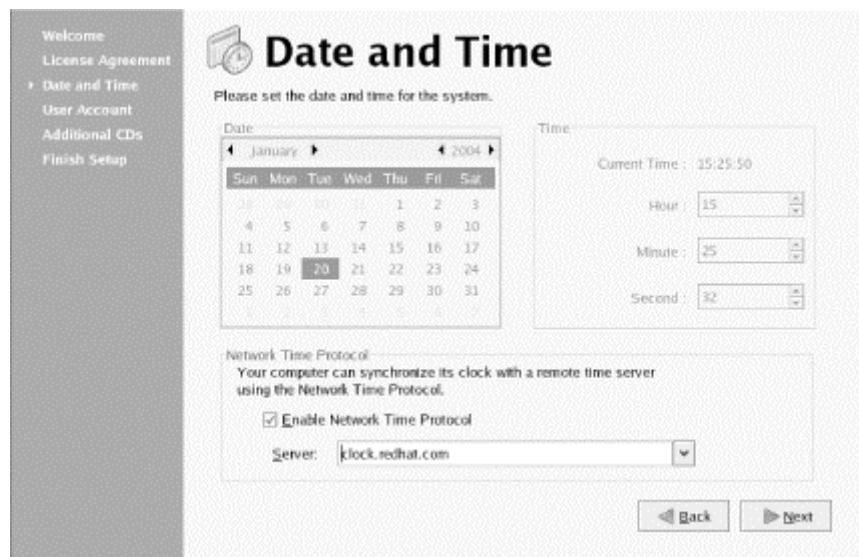
The License Agreement screen ([Figure 3-31](#)) requires you to consent to Red Hat's or the Fedora team's license agreement, according to the operating system you chose to install; otherwise you'll be unable to complete the configuration of your system. To consent to the agreement, click the radiobutton labeled "Yes, I agree to the License Agreement" and then click Next.

Figure 3-31. The License Agreement screen



The Date and Time screen ([Figure 3-32](#)) lets you specify the current date and time. You can also specify a time server to which your system can synchronize, so that your system's time is highly accurate. If necessary, revise the date and time. If you want to synchronize your system's time, enable the checkbox labeled Enable Network Time Protocol and select a time server from the list labeled Server. Click Next to continue.

Figure 3-32. The Date and Time screen



The User Account screen ([Figure 3-33](#)) lets you create a personal user account. You should generally create a personal user account and use it to log into your system, reserving use of the root account for configuring and administering the system. Doing so will help keep your system secure. To create a user account, specify a username, the full name of the user,

and a password that will be used to authenticate the user. The screen requires you to enter the password twice, to help avoid mistyping the password. Click Next to proceed.

Figure 3-33. The User Account screen



The screenshot shows the 'User Account' window. On the left is a sidebar with a list of setup steps: 'Welcome', 'License Agreement', 'Date and Time', 'User Account' (which is highlighted with a right-pointing arrow), 'Additional CDs', and 'Finish Setup'. The main area has a title 'User Account' with a small icon. Below the title is a message: 'It is recommended that you create a personal user account for normal (non-administrative) use. To create a personal account, provide the requested information.' There are four input fields: 'Username:' with 'bmccarty', 'Full Name:' with 'Bill McCarty', 'Password:' with 'password', and 'Confirm Password:' with 'password'. Below these fields is a note: 'If you need to use network authentication such as Kerberos or NFS, please click the Use Network Login button.' There is a button labeled 'Use Network Login...'. At the bottom right are two buttons: 'Back' and 'Next'.



The User Account screen also provides a button labeled Use Network Login, which provides access to a dialog box that lets you configure login and authentication options. If you plan to log into a local network that authenticates its users, you can use this dialog box to specify the authentication method and options. Consult the administrator of your local network to learn about the network's authentication requirements.

3.3.1 Red Hat Network

If you're installing Red Hat Enterprise Linux, the Red Hat Network screen ([Figure 3-34](#)) appears. Otherwise, you should skip ahead to the [Section 3.3.2](#).

Figure 3-34. The Red Hat Network screen



The Red Hat Network screen lets you register your system to receive updates via Red Hat Network. [Chapter 8](#) explains Red Hat Network and how to use it. The default choice is "Yes, I want to register my system with Red Hat Network." However, it's likely that your system is not yet ready for this procedure. Unless your network administrator has directed otherwise, override the default choice by specifying "No, I do not want to register my system." Then, click Next to continue.

3.3.2 Additional CDs

The Additional CDs screen ([Figure 3-35](#)) lets you install Linux documentation from a documentation CD, install package groups you omitted during system installation, or install applications from specially prepared CDs. However, you can easily install documentation, package groups, or applications later. So, merely click Next to continue.

Figure 3-35. The Install Additional CDs screen



Finally, the Firstboot service displays the Finish Setup screen ([Figure 3-36](#)). Click Next to prepare to log in to your system for the first time.

Figure 3-36. The Setup Agent Finished Setup Screen



3.4 Logging into the Desktop

The login screen should resemble the one shown in [Figure 3-37](#). To log in, type **root**, or the name of another user account you created, in the text box labeled Login and press **Enter**. A second login screen appears, requesting your password. Type the password you earlier assigned to the user and press **Enter**.



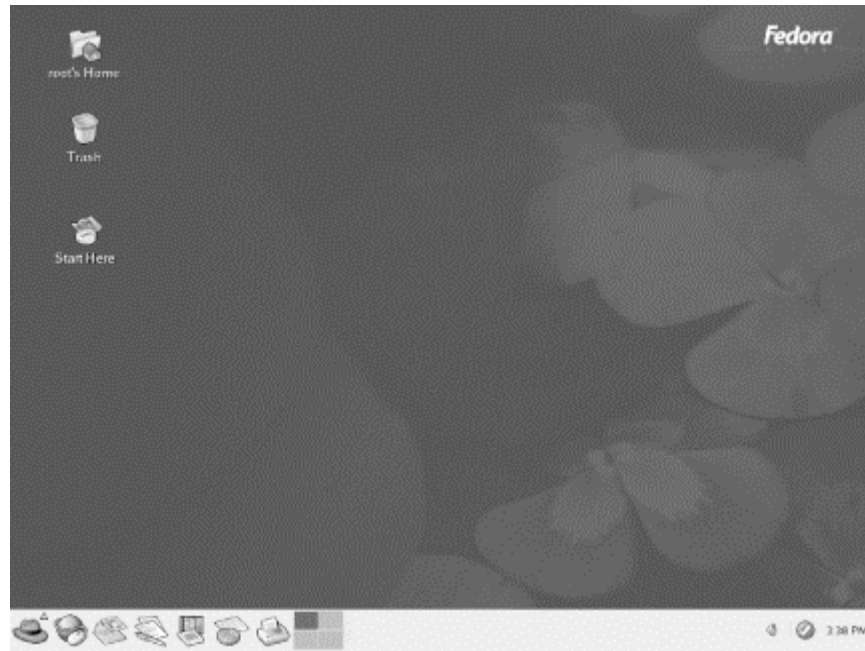
If you chose to install the KDE desktop rather than the GNOME desktop, the appearance of your login screen will differ somewhat from that shown in [Figure 3-37](#). However, the GNOME and KDE login screens are functionally similar.

Figure 3-37. The GNOME login screen



The GNOME desktop ([Figure 3-38](#)) appears. If you like, click around the desktop and see what you can discover. [Chapter 5](#) explains how to use the GNOME desktop. However, you should read [Chapter 4](#) before reading that chapter.

Figure 3-38. The GNOME desktop



To shut down your system, click the red hat at the lower left of the desktop. Then click Shutdown and click OK. Wait for your system to power down; now you're ready to move on to [Chapter 4](#) to begin learning how Linux works.



You may see a flashing red ball at the lower right of the GNOME desktop. The ball is associated with the Red Hat Network Alert Notification facility, which you'll learn about in [Chapter 8](#).

3.5 Getting Help

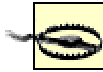
If your system fails to boot or if you're unable to complete the Linux installation process, don't despair: this section will help you troubleshoot your installation.

3.5.1 Failed Graphical Startup or Login

Sometimes, the graphical startup or login fails. This is more likely if your system wasn't manufactured recently or if it's a laptop rather than a desktop.

If you don't see either display manager's screen, it's likely that your X configuration isn't appropriate for your system's video hardware. It could be that your X configuration requires

revision. In the worst case, your system's video hardware may not be compatible with X; in that event, you'll nevertheless be able to run nongraphical Linux applications.



If you see a scrambled image rather than text or images on your monitor, immediately switch off the monitor. If your monitor is an older model, it can be damaged by the incorrect configuration. Using the resources described in the following section, seek help in configuring X to work with your system's video adapter and monitor.

3.5.2 Additional Resources

As much as I'd like to help you solve your problems in installing Linux, I get too much email to be able to respond personally. But don't fret: the help you need is probably close by, in one of these sources (roughly ordered by importance and value):

The Red Hat Enterprise Linux Installation Guide

<http://www.redhat.com/docs>

This guide is distributed with boxed copies of Red Hat Enterprise Linux. It's also available online. It provides a step-by-step guide to installing Red Hat Enterprise Linux that includes a few details not given in this chapter. Unfortunately, no similar guide has yet been developed for Fedora Core 1.

RedHat Enterprise Linux General Advisories

<http://www.redhat.com/apps/support/errata>

The Red Hat Enterprise Linux General Advisories web page describes problems with, and enhancements to, Red Hat Enterprise Linux and provides links to updated packages. Sometimes, the installation media themselves are found to contain bugs or security problems. In that case, you may be able to find and download fixes from this web page. [Chapter 8](#) explains Red Hat Network, which enables you to configure your system to automatically notify you of available software updates.

Fedora Updates

<http://fedora.redhat.com/download/updates.html>

The Fedora project has not yet established a web page for publishing advisories. However, their updates web page will likely link to such information in the future.

Red Hat Global Support Services

<https://www.redhat.com/apps/support>

This web page provides hyperlinks to a variety of information sources and resources for Red Hat Enterprise Linux users. Because Red Hat does not support Fedora Core, this resource applies only to Red Hat Enterprise Linux.

Red Hat Linux Product Activation Page

<http://www.redhat.com/now>

Red Hat Enterprise Linux users can access this page to register their product. Users who purchased Red Hat Linux on-line do not need to separately register their product, because it was automatically registered during purchase.

The Linux on Laptops Page

<http://www.cs.utexas.edu/users/kharker/linux-laptop>

Installing Linux on a laptop presents special difficulties, because laptops regularly contain peculiar hardware. Worse, two laptops that have identical model numbers may contain different hardware. The Linux on Laptops web page is an essential resource for those who want to install Linux on a laptop.

Bugzilla

<http://bugzilla.redhat.com/bugzilla>

This searchable database covers problems reported by users of Linux. You can use Bugzilla to determine if someone else has had the same problem you're experiencing. Often, the Bugzilla record will include a workaround or fix for your problem. The same bugzilla web site is used for Red Hat Enterprise Linux and Fedora Core.

Mailing Lists

<https://listman.redhat.com/mailman/listinfo>

<http://fedora.redhat.com/participate/communicate>

Red Hat hosts several mailing lists to which you can post questions and expect to receive replies from other list members. In addition, you can view archives of past postings. The Redhat-install-list is dedicated to issues related to the installation of Linux. The Taroon-list is dedicated to Red Hat Enterprise Linux, Version 3.

Running Linux

The book *Running Linux* (O'Reilly), by Matt Walsh, Matthias Kalle Dalheimer, Terry Dawson, and Lar Kaufman, takes a more advanced look at Linux generally, rather than only at the Red Hat Enterprise Linux or Fedora Core distributions. It's full of insights and techniques that will help you master Linux. You'll find it especially useful for continuing study after completing this book. You may also find it useful in troubleshooting and resolving installation and configuration problems.

Chapter 4. How Linux Works

Before you can effectively use a desktop environment, you need to know some Linux fundamentals. This chapter explains basic Linux concepts that underlie graphical and nongraphical system use. It describes Linux user accounts and how Linux organizes data as filesystems, directories, and files. This chapter also explains how to use the X Window System (often known simply as X). Because both GNOME and KDE are built on top of X, an understanding of X is central to using either desktop environment. Even though you're probably eager to get working with your new system, I suggest you at least skim this

chapter. I also predict that you'll come back to it when you have some more experience and run into something confusing.

4.1 User Accounts

Like other multiuser operating systems, such as Windows NT/2000/XP, Linux uses user accounts to identify users and allocate permissions. Every Linux system has a special user known as the *root user*. The root user is analogous to the Windows user known as Administrator. The root user can perform privileged operations that are forbidden to other users. For instance, only the root user can perform most system administration operations. By default, the username associated with the root user is **root**.

You should be judicious in your use of the root account. For instance, you should safeguard the associated password so that no one uses it to compromise your system. Also, you should log in as the root user only when performing privileged operations; that is, special administrative operations that can be performed only by the root user. Following this advice will help you avoid disasters such as accidentally deleting important files that are protected against access by ordinary, non-root users.

4.2 How Linux Organizes Data

In order to make the most effective use of your Linux system, you must understand how Linux organizes data. If you're familiar with Windows or another operating system, you'll find it easy to learn how Linux organizes data, because most operating systems organize their data in similar ways. This section explains how Linux organizes data and introduces you to several important Linux commands that work with directories and files.

4.2.1 Devices

Linux receives data from, sends data to, and stores data on *devices*. A device generally corresponds to a hardware unit, such as a keyboard or serial port. However, a device may have no hardware counterpart: the kernel creates several *pseudodevices* that you can access as devices but that have no physical existence. Moreover, a single hardware unit may correspond to several devices. For example, Linux defines each partition of a disk drive as a distinct device. [Table 4-1](#) describes some typical Linux devices; not every system provides all these devices and some systems provide devices not shown in the table. The device name often appears in messages and filenames on the system.

Table 4-1. Typical Linux devices

Device	Description
audio	Sound card

Table 4-1. Typical Linux devices

Device	Description
<code>cdrom</code>	CD-ROM drive
<code>console</code>	Current virtual console
<code>fdn</code>	Floppy drive (<i>n</i> designates the drive; for example, fd0 is the first floppy drive)
<code>ftape</code>	Streaming tape drive, not supporting rewind
<code>hdxn</code>	Non-SCSI hard drive (<i>x</i> designates the drive and <i>n</i> designates the partition; for example, hda1 is the first partition of the first non-SCSI hard drive)
<code>lpn</code>	Parallel port (<i>n</i> designates the device number; for example, lp0 is the first parallel port)
<code>modem</code>	Modem
<code>mouse</code>	Mouse
<code>nrftn</code>	Streaming tape drive, supporting rewind (<i>n</i> designates the device number; for example, nrft0 is the first streaming tape drive)
<code>nstn</code>	Streaming SCSI tape drive, not supporting rewind (<i>n</i> designates the device number; for example, nst0 is the first streaming SCSI tape drive)
<code>null</code>	Pseudodevice that accepts unlimited output and throws it away
<code>printer</code>	Printer
<code>psaux</code>	PS/2 mouse
<code>rftn</code>	Streaming tape drive, not supporting rewind (<i>n</i> designates the device number; for example, rft0 is the first streaming tape drive)
<code>scdn</code>	SCSI CD-ROM (<i>n</i> designates the device number; for example, scd0 is the first SCSI CD-ROM)
<code>sdxn</code>	SCSI hard drive (<i>x</i> designates the drive and <i>n</i> designates the partition; for example, sda1 is the first partition of the first SCSI hard drive)
<code>stn</code>	Streaming SCSI tape drive, supporting rewind (<i>n</i> designates the device number; for example, st0 is the first streaming SCSI tape drive)
<code>ttyn</code>	Virtual console (<i>n</i> designates the particular virtual console; for example, ttv0 is

Table 4-1. Typical Linux devices

Device	Description
	the first virtual console)
<code>ttySn</code>	Modem (<i>n</i> designates the port; for example, <code>ttyS0</code> is an incoming modem connection on the first serial port), serial device (such as Palm Pilot), or some PCMCIA devices
<code>zero</code>	Pseudodevice that supplies an inexhaustible stream of zero-bytes

4.2.2 Filesystems

Whether you're using Windows or Linux, you must format a partition before you can store data on it. The installation procedure automatically formats the partitions you create during system installation. When the installation procedure formats a partition, it also writes special data, called a *filesystem*, on the partition. The filesystem organizes the available space and provides a directory that lets you assign a name to each *file*, which is a set of stored data. A filesystem also enables you to group files into directories, which function much like the folders you create using the Windows Explorer: directories store information about the files they contain.

Just as every partition must have a filesystem, every CD-ROM and floppy diskette must have a filesystem. The filesystem of a CD-ROM is written when the disk is created; the filesystem of a floppy diskette is rewritten each time you format it.

Windows 98 lets you choose to format a partition as FAT or FAT32. Windows NT, 2000, and XP also support the NTFS filesystem type. Linux supports a wider variety of filesystem types; [Table 4-2](#) summarizes the most common ones. The most important filesystem types are *ext3*, which is used for Linux native partitions; *msdos*, which is used for FAT partitions (and floppy diskettes) of the sort created by MS-DOS and Microsoft Windows; and *iso9660*, which is used for CD-ROMs. Linux also provides the *vfat* filesystem, which is used for FAT32 partitions of the sort created by Windows 9x. Linux can also read Windows NT/2000/XP NTFS filesystems; however, the support for doing so is not enabled in the standard Red Hat Linux kernel.

Table 4-2. Common filesystem types

Filesystem	Description
<i>ext2</i>	The predecessor of the <i>ext3</i> filesystem; supported for compatibility.
<i>ext3</i>	The standard Linux filesystem.
<i>iso9660</i>	The standard filesystem used on CD-ROMs.
<i>msdos</i>	A filesystem compatible with Microsoft's FAT filesystem, used by MS-DOS and Windows.
<i>nfs</i>	A filesystem compatible with Sun's Network File System. An <i>nfs</i> filesystem

Table 4-2. Common filesystem types	
Filesystem	Description
	does not reside on a physical partition; it is accessed via the network.
<i>reiserfs</i>	A Linux filesystem designed for high-reliability, large-capacity storage systems.
<i>vfat</i>	A filesystem compatible with Microsoft's FAT32 filesystem, used by Windows 9x.

4.2.3 Directories and Paths

If you've used MS-DOS, you're familiar with the concepts of file and directory and with various MS-DOS commands that work with files and directories. Under Linux, files and directories work much as they do under MS-DOS.

4.2.3.1 Home and working directories

When you log in to Linux, you're placed in a special directory known as your *home directory*. Generally, each user has a distinct home directory, where the user creates personal files. This makes it simple for the user to find files previously created, because they're kept separate from the files of other users and system files.

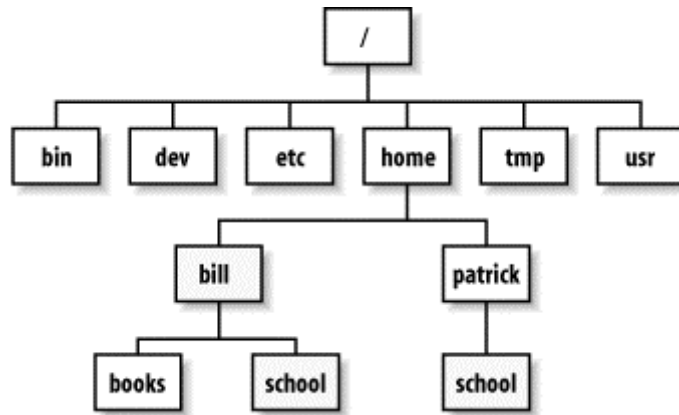
The *current directory*—or *current working directory*, as it's sometimes called—is the directory you're currently working in. When you log in to Linux, you're automatically placed in your home directory.

4.2.3.2 The directory tree

The directories of a Linux system are organized as a hierarchy. Unlike MS-DOS, which provides a separate hierarchy for each partition, Linux provides a single hierarchy that includes every partition. The topmost directory of the directory tree is the *root directory*, which is written using a forward slash (/), not the backward slash (\) used by MS-DOS to designate a root directory.

[Figure 4-1](#) shows a hypothetical Linux directory tree; a real Linux system contains many more directories. The root directory contains six subdirectories: */bin*, */dev*, */etc*, */home*, */tmp*, and */usr*. The */home* directory has two subdirectories; each is the home directory of a user and has the same name as the user who owns it. The user named *bill* has created two subdirectories in his home directory: *books* and *school*. The user named *patrick* has created the single *school* subdirectory in his home directory.

Figure 4-1. A hypothetical Linux directory tree



Each directory (other than the root directory) is contained in a directory known as its parent directory. For example, the parent directory of the *bill* directory is *home*.



The *root* user has a special home directory, */root*. This directory is commonly called "slash root" to distinguish it from the root directory, */*.

4.2.3.3 Absolute and relative pathnames

If you look closely at [Figure 4-1](#), you'll see that two directories named *school* exist; one is a subdirectory of *bill* and the other is a subdirectory of *patrick*. To avoid confusion that could result when several directories have the same name, directories are specified using *pathnames*.

There are two kinds of pathnames: *absolute* and *relative*. The absolute pathname of a directory traces the location of the directory beginning at the root directory; you form the pathname as a list of directories, separated by forward slashes (/). For example, the absolute pathname of the unique directory named *bill* is */home/bill*. The absolute pathname of the *school* subdirectory of the *bill* directory is */home/bill/school*. The absolute pathname of the identically named *school* subdirectory of the *patrick* directory is */home/patrick/school*.

When a subdirectory is many levels below the root directory, its absolute pathname may be long and cumbersome. In such a case, it's often more convenient to use a relative pathname, which gives the current directory, rather than the root directory, as its starting point. For example, suppose that the *bill* directory is the current directory; you can refer to its *books* subdirectory by the relative pathname *books*. A relative pathname must never begin with a forward slash, whereas an absolute pathname must begin with a forward slash. As a second example, suppose that the */home* directory is the current directory. The relative pathname of the *school* subdirectory of the *bill* directory would be *bill/school*; the relative pathname of the identically named subdirectory of the *patrick* directory would be *patrick/school*.

Linux provides two special directory names. Using a single dot (.) as a directory name is equivalent to specifying the current directory. Using two dots (..) within a pathname refers to the directory up one level in the current path; that is, to the parent directory. For example, if the current directory is */home/bill*, then *..* refers to the */home* directory.

Similarly, if the current directory is `/home/bill` and the directory tree is that shown in [Figure 4-1](#), the path `../patrick/school` refers to the `/home/patrick/school` directory .

4.2.4 File Permissions

Unlike Windows 9x, but like other varieties of Unix and Windows NT/2000, Linux is a multiuser operating system. Therefore, it includes mechanisms to protect data from unauthorized access. The primary protection mechanism restricts access to directories and files based on the identity of the user who requests access and on access modes assigned to each directory and file.

Each directory and file has an associated user, called the *owner*. The user who initially creates a file is the owner of the file. Each user belongs to one or more sets of users known as *groups*. Each directory and file has an associated group, which is assigned when the directory or file is created. The user and the group can be changed later.

Access permissions, also known as *modes*, determine what operations a user can perform on a directory or file. [Table 4-3](#) lists the most common permissions and explains the meaning of each. Notice that permissions work differently for directories than for files. For example, permission `r` denotes the ability to list the contents of a directory or *read* the contents of a file. A directory or file can have multiple permissions. Only the listed permissions are granted; any other operations are prohibited. For example, a user who had file permission `rw` could *read* or *write* the file but could not *execute* it, as indicated by the absence of the execute permission, `x`.

Table 4-3. Common access permissions		
Permission	Operations allowed on a directory	Operations allowed on a file
<code>r</code>	List the directory	Read contents
<code>w</code>	Create or remove files	Write contents
<code>x</code>	Access files and subdirectories	Execute

The access modes of a directory or file consist of three sets of permissions:

User/Owner

Applies to the owner of the file

Group

Applies to users who are members of the group assigned to the file

Other

Applies to other users, sometimes referred to as *world*

The `ls` command, which you'll meet in [Chapter 7](#), lists the file access modes in the second column of its long output format, as shown in [Figure 4-2](#). The GNOME and KDE file managers use this same format. The column contains nine characters: the first three specify

the access allowed the owner of the directory or file, the second three specify the access allowed users in the same group as the directory or file, and the final three specify the access allowed to other users (see [Figure 4-3](#)).

Figure 4-2. Access modes as shown by the ls command

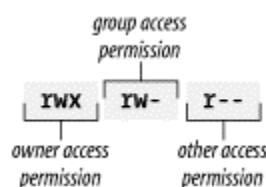
```

drwxr-xr-x  5 root root 1024 Dec 23 13:48 GNUstep
-rw-r--r--  1 root root  331 Feb 11 10:19 Xrootenv.0

```

access modes
owner
group

Figure 4-3. Access modes specify three permissions



4.2.5 Mounting and Unmounting Filesystems

You cannot access a hard drive partition, CD-ROM, or floppy disk until the related device or partition is *mounted*. Mounting a device checks the status of the device and readies it for access when it boots or when you launch a desktop environment. By default, the GNOME and KDE desktop environments automatically mount devices that use removable media when you insert new media.

Before you can remove media from a device, you must unmount it. You can unmount a device by using a desktop environment or issuing a command. For your convenience, the system automatically unmounts devices when it shuts down. A device can be unmounted only if it's not in use. For example, if a user's current directory is a directory of the device, the device cannot be unmounted.



A common error is attempting to mount a CD-ROM or floppy without first inserting the media. If you're unable to mount a device that uses removable media, check that the media is in place. Also, be sure to unmount media before ejecting it, especially when using USB or Firewire drives. If you fail to do so, data on the media may be corrupted.

4.3 Using X

X is the standard graphical user interface (GUI) for Linux. Like other GUIs, such as Windows and Mac OS, X lets you interact with programs by using a mouse (or other pointing device) to point and click, providing a simple means of communicating with your computer.

Despite its age, X is a remarkable and very modern software system offering a cross-platform, network-oriented GUI. It runs on a wide variety of platforms including essentially every flavor of Unix, such as Solaris, Linux, and the BSDs (FreeBSD, NetBSD, and OpenBSD). X clients are available for use, for example, under Windows 3.x, 9x, NT, 2000, and XP. The sophisticated networking capabilities of X let you run a program on one computer while viewing the graphical output on another computer via a network connection. X was designed to provide room for experimentation with new interfaces, so a wide variety of X-based window managers and desktops is available for every taste. On the other hand, this variety can provoke some minor confusion, as interfaces and behaviors vary slightly from one system to another.

Most Linux users run XFree86, a freely available software system compatible with X, which is distributed with Red Hat Linux. XFree86 was developed by the XFree86 software team, which began work in 1992. In 1994, the XFree86 Project, Inc.

(<http://www.xfree86.org>) assumed responsibility for ongoing research and development of XFree86.

4.3.1 Window Managers

Using X means interacting with Linux on several different levels. X itself merely provides the graphics facility for displaying components of a GUI: X draws the screen, draws objects on the screen, and tracks user input actions such as keyboard input and mouse operations. To organize the desktop into familiar objects like windows, menus, and scrollbars, X relies on a separate program called a window manager. But even more functionality is required. A window manager alone doesn't provide tight integration between applications of the sort required by drag-and-drop operations; that higher degree of integration comes from what's called a desktop environment. While X itself is a single program, X under Linux supports several popular window managers and two popular desktop environments, GNOME and KDE.

Window managers create the borders, icons, and menus that provide a simple-to-use interface. Window managers also control the look and feel of X, letting you configure X to operate almost any way you desire. At one time, it was common for Linux users to separately choose a window manager and desktop environment. However, today most users retain the window manager with which their preferred desktop environment is initially configured. Under Red Hat Linux Enterprise 3, GNOME uses the Metacity window manager and KDE uses Kwin, formerly known as the K Window Manager, or simply K. Because of the variety of window managers, scrollbars and other widgets may behave differently from one system to another. But, the differences are minor and determined

clicking generally discovers the proper method of interacting with a widget. See [Section 4.3.5.2](#) later in this chapter.

4.3.2 Desktop Environments

A *desktop environment* is a set of desktop tools and applications. The Windows desktop includes applications such as the Windows Explorer, accessories such as Notepad, games such as FreeCell and Minesweeper, and utilities such as the Control Panel and its applets. Although you can run X without a desktop, having a desktop helps you work more efficiently. Both GNOME and KDE are free software and are developed by teams in an open, collaborative manner.

The default Red Hat Linux desktop environment is GNOME. However, you can easily reconfigure KDE as the desktop, if you prefer. The choice between GNOME and KDE is now not so important as in the past. Red Hat has reworked GNOME and KDE to give them a consistent look and feel. Moreover, almost every GNOME application can now be run under KDE and almost every KDE application can now be run under GNOME. So your Linux experience will be similar whether you're using GNOME or KDE.

4.3.2.1 GNOME

GNOME stands for the GNU Network Object Model Environment (pronounced as *guh-nome* or *gee-nome*). One of GNOME's most interesting features is session awareness. When you re-enter GNOME after logging out, it reconfigures your desktop to match the state at the time you exited by launching each application that was open when you exited. GNOME even restores each application to its former state by, for example, moving to the page that was open when you exited.

Both GNOME and KDE support a myriad of standard and optional desktop tools and applications, such as:

- Games and amusements such as freecell, gnibbles, gnobots, gnomine, mahjongg, and sol
- The GNU Image Manipulation Program (GIMP)
- Ghostview, which lets you view PostScript files and print them on non-PostScript printers
- Internet applications such as Evolution, Mozilla, gFTP, X-Chat and gaim, an instant messenger compatible with AOL's AIM
- Multimedia applications such as Audio Player, Sound Mixer, and CD Player
- General applications such as gEdit, a text editor; Mozilla, the popular open source web browser; and OpenOffice.org, a desktop suite featuring word processing, spreadsheets, presentations, and other facilities
- Utilities for configuring and using your Red Hat Linux system

4.3.2.2 KDE

KDE (the K Desktop Environment) includes Kwin, the K Window Manager, as an integral component. KDE provides a file manager, a help system, a configuration utility, and a variety of accessories and applications, such as:

- Games such as Kasteriods, K Mines, and K Poker
- Graphical applications such as KuickShow, an image browser/viewer
- Multimedia applications such as KMid, a karaoke MIDI file player; and KsCD, a CD player
- Network applications such as Kmail, a mail client; and KNewsTicker, an applet that displays news from web sites of your choice

As with GNOME, new KDE accessories and applications are available almost weekly.



At one time, KDE was distributed under a license that suggested that some users owed a fee to developers of an important library used to develop KDE. This inhibited acceptance of KDE within the free software community. As a result, several releases of Red Hat Linux featured only GNOME, despite the popularity of KDE. Currently, KDE is open source and may be freely distributed.

4.3.3 Keyboard Operations

Using the keyboard with X closely resembles using the keyboard with Windows. X sends your keyboard input to the active window, which is said to have the *input focus*. The active window is usually the window in which you most recently clicked the mouse.



This chapter refers to your pointing device as a *mouse*. However, like Windows, X supports a variety of pointing devices.

While Windows lets you choose to perform most operations by using the keyboard or mouse, X was designed for use with a mouse. If your mouse isn't functioning, you'll find it quite challenging or even impossible to use most X programs. X allows you to perform only a few important functions via the keyboard:

- Switching video modes
- Using virtual consoles
- Abruptly terminating X

4.3.3.1 Switching video modes

When you configured X, you specified the video modes in which X can operate. Recall that the current video mode determines the resolution and color depth of the image displayed by your monitor—for example, 16 bits per pixel color depth and 1024x768 pixels for screen resolution.

By pressing **Ctrl-Alt-+** (using the Plus key on the numeric keypad), you command X to switch to the next video mode in sequence. X treats the video modes as a cycle: if X is operating in the last video mode, this key sequence causes X to return to the first video mode.

The similar key sequence **Ctrl-Alt-** (using the minus key on the numeric keypad) causes X to switch to the previous video model. If you shift to a video mode that your monitor doesn't support—as demonstrated by an unsteady or garbled image—you can use this key sequence to return to a supported video mode, avoiding the inconvenience of terminating X and reconfiguring your system.

4.3.3.2 Using virtual consoles

Even while X is running, you can access the Linux virtual consoles. For instance, you may find it useful to do so in order to recover from a X-related problem, so long as the problem hasn't frozen the keyboard. To switch from graphical mode to a virtual console running in text mode, type **Ctrl-Alt-F_n**, where **F** is a function key and *n* is the number of the desired virtual console. X uses virtual console #7, so only virtual consoles #1-6 are accessible while running X.

To switch from a virtual console back to X, type **Ctrl-Alt-F7**. Nothing is lost when you switch from X to a virtual console or back, so you can move freely between the graphical and text operating modes.

4.3.3.3 Terminating X

You can terminate X abruptly by typing **Ctrl-Alt-Backspace**. However, this method of terminating X is appropriate only when X is malfunctioning. Terminating X abruptly closes down running applications, which may result in loss of data. You should generally exit X gracefully, by clicking the Red Hat and choosing Logout from the main menu.

4.3.4 Terminal Windows and Pop-up Menus

In Windows, you don't need to restart in DOS mode simply to have access to the DOS command line. Similarly, in X you don't need to switch to a virtual console simply to have access to the command line. X enables you to open a terminal window. A terminal window resembles the MS-DOS Prompt window or command-line interface window; like a Linux virtual console, it lets you type commands and view command output. Various window managers support different methods of accessing a terminal window, as described in [Chapter 5](#).

The terminal window is just one example of a frequently used program under X that you'll want to access. Most window managers install with a default set of common programs that can be accessed by right-clicking with the mouse on the desktop. For example, most window managers let you right-click on the desktop and select a terminal window program from the pop-up menu that appears. However, the pop-up menu displayed by a window manager may display program names rather than program functions. In this case, you may have some difficulty determining which entry on the pop-up menu corresponds to a terminal program. Many programs that provide terminal windows have names that include

the sequences *xt* or *xterm*. Selecting such an entry will launch a terminal window. You'll learn more about window managers later in this chapter.

4.3.5 Mouse Operations

Mouse operations under X are similar to mouse operations under Windows, although you perform them differently. The most common mouse operations that behave differently are:

- Copying and pasting text
- Using scrollbars

4.3.5.1 Copying and pasting text

To copy and paste text, you must first mark the text by moving the mouse to the beginning of the text; then click the left mouse button and drag the mouse across the text to be copied. X automatically copies the marked text into a buffer; you don't need to press **Ctrl-C** or perform any other operation. However, for compatibility with Windows, some window managers let you copy text by pressing **Ctrl-C**, or by right clicking and choosing from a pop-up menu. If you find that you need to change the size of the marked text section, you can click the right mouse button and move the mouse to adjust the marked text.



Some window managers display a pop-up menu when you click the right button, even when the mouse cursor is above text. When using such a window manager, you cannot use the right mouse button to adjust the size of the marked text section.

To paste the text, properly position the insertion point and click the middle mouse button. If your mouse has only two buttons, simultaneously click the left and right buttons to simulate clicking the middle mouse button. You may find that this operation requires a little practice before you get it right, but once you've mastered it, you'll find it works almost as well as having a three-button mouse. Some window managers will let you paste text by pressing **Shift-Ctrl-V**, or by right-clicking and choosing from a pop-up menu.

4.3.5.2 Using scrollbars

Many X programs provide scrollbars that resemble those provided by Windows. However, the operation of scrollbars under X originally differed from that under Windows. Most X programs have been revised to display scrollbars that work like Windows scrollbars, although a few have not.

If you're having trouble using a scrollbar, it may have been programmed to use the original X method of interaction. This problem is more likely to affect old X programs than recent ones. To page forward using the original X method, click the left mouse button on the scrollbar. Clicking near the top of the scrollbar scrolls forward a short distance, as little as a single line. Clicking near the bottom of the scrollbar scrolls the window by a page. To page backward, click the right mouse button on the scrollbar. Again, clicking near the top of the

scrollbar scrolls a short distance, as little as a single line. Clicking near the bottom of the scrollbar scrolls the window by a page.

4.3.6 Virtual Desktop

Under X, your desktop can be scrollable; that is, larger than the size of your monitor. For example, even if your monitor has a maximum resolution of 800 600, you might have a desktop of 1600 1200 or even 3200 2400. Such a desktop is known as a *virtual desktop*.



Don't confuse the term *virtual desktop* with the term *virtual console*. A virtual console is used to log in and enter commands in text mode; a virtual desktop is used to obtain an oversized desktop in graphics mode.

Most desktop environments provide a tool called a *pager*, which lets you move around the virtual desktop. The pager provides a thumbnail view of your virtual desktop; by clicking within the thumbnail, you center your actual desktop on the clicked location. You'll learn more about pagers in the next two chapters.

Chapter 5. Using the GNOME and KDE Desktops

Red Hat Enterprise Linux and Fedora Core support two desktops, GNOME and KDE. This choice is consistent with the Linux philosophy of having it your own way. But the reasons behind having multiple desktops have more to do with history and law than technology. At one time, parts of KDE were distributed under a license that some believed required commercial users to pay a license fee. Because Red Hat wanted Red Hat Linux to be freely redistributable and usable, Red Hat included only GNOME in the Red Hat Linux distribution. Red Hat also assisted in the development of the GNOME desktop. However, the KDE license was eventually clarified. At that point, Red Hat warmed toward KDE and included it in the Red Hat Linux distribution. Many users prefer KDE to GNOME, finding it in many ways more mature than GNOME. But GNOME retains a somewhat favored status in Red Hat's eyes, as indicated by the installation program's default choice of GNOME as the desktop. If you want to install KDE, you must manually select the KDE package group.

To minimize the confusion that might otherwise result from dueling desktops, Red Hat has worked to give GNOME and KDE a more consistent look and feel. This has upset many KDE fans, who prefer KDE's native look and feel to that imposed by Red Hat. An advantage of Red Hat's decision is that most applications work properly under both GNOME and KDE. However, a disadvantage of Red Hat's decision is that KDE now includes applications that lack the distinctive look and feel that unites the KDE desktop.

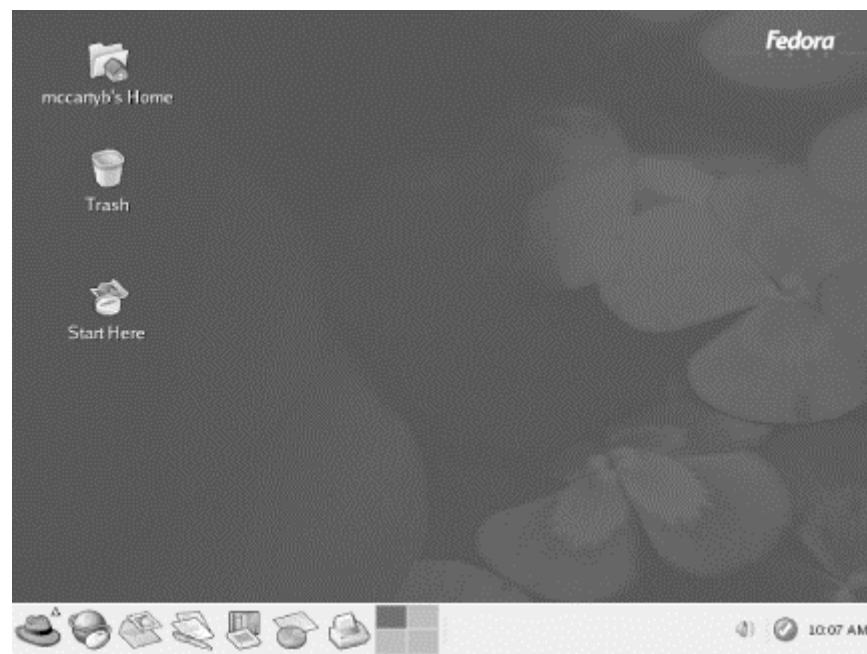
I'm not much interested in justifying or attacking Red Hat's decision. The decision has been made, and the goal of this book is to describe Red Hat Enterprise Linux as it is, rather than as it might be. Therefore, this chapter describes both desktops, devoting roughly equal space to each. My personal recommendation is that you try each desktop for a while and use the one you prefer.

Some readers of earlier editions of this book complain that I favor one desktop over the other. Yes, I do have a personal favorite. However, almost all such readers have incorrectly identified my preference. And, my preference has recently changed. So, I believe that my presentation of the desktops is reasonably fair and unbiased.

5.1 Using the GNOME Desktop

When you first log in to your Red Hat Enterprise Linux or Fedora Core system, you will see the GNOME desktop ([Figure 5-1](#)). The contents of your desktop may be slightly different, of course.

Figure 5-1. The GNOME desktop



If you want to launch a GNOME session, but KDE is configured as the default desktop environment, select Gnome from the Session menu of the system login screen. Of course, GNOME must be installed in order for this to work.

To log out of GNOME, left-click on the main menu, which resembles a red hat. From the pop-up menu that appears, select the Log Out menu item ([Figure 5-2](#)). A Log Out dialog

box (Figure 5-3) appears and asks you to confirm your decision to log out. Clicking OK terminates your GNOME session. If you enable the checkbox titled Save Current Setup, the GNOME session manager will save the state of your desktop and restore it when you log in again to GNOME.

There are two other options in the Log Out dialog box. Select the Shut Down button to shut down your system, or the Restart the computer button to restart it.

Figure 5-2. Logging out of GNOME

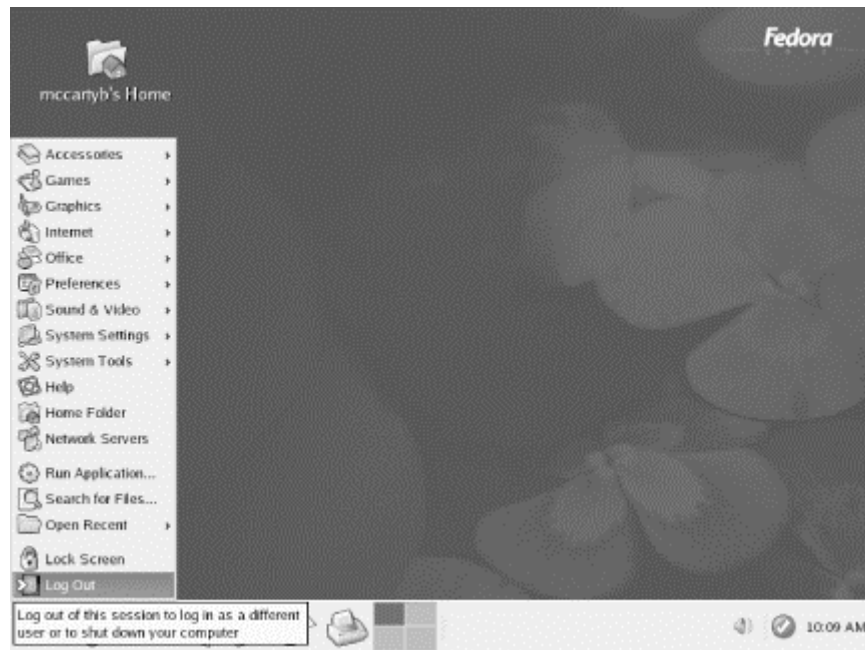
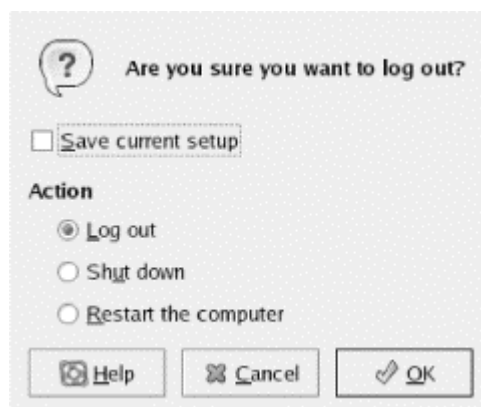


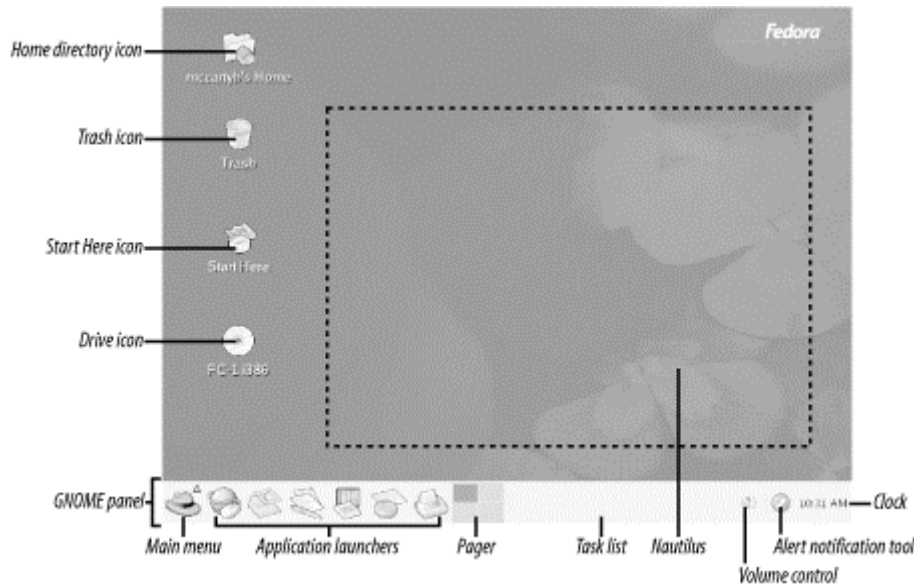
Figure 5-3. The Log Out dialog box



5.1.1 The GNOME Desktop

The term *desktop* can be used in either of two senses. It can refer either to the entire GNOME display, or to the empty area of the display where no windows or icons appear, as indicated in [Figure 5-4](#). To keep straight these meanings, *GNOME desktop* will be used when referring to the entire display and *desktop* will be used when referring to the empty area of the display.

Figure 5-4. Parts of the GNOME desktop



Right-clicking the desktop causes a pop-up menu to appear; this menu lets you perform a variety of operations, such as:

- Creating a new folder
- Creating a new launcher
- Opening a terminal window

[Figure 5-4](#) shows the elements of the GNOME desktop, which are described in the following sections.

5.1.1.1 Pager

There's one particular GNOME and KDE feature that I miss when working with Windows: the pager. The pager is the rectangle at the bottom of the screen, just left of center. You can recognize it by the four smaller rectangles nested within it.

The pager provides what's called a *virtual desktop*; that is, a desktop that's larger than the size of your monitor screen. Of course, you can't see the entire virtual desktop at once, since it's larger than your monitor screen. So, you use the pager to select which quadrant of the virtual desktop you're viewing.

To see how the pager works, click on a pager quadrant other than the default, which is the upper-left quadrant. Any open windows vanish. Click on the upper-left pager quadrant to restore them. As a further experiment, click on a quadrant other than the default. Then,

launch a program from the menu. When the program's window appears, restore the default desktop view by clicking the upper-left pager quadrant. The new window disappears. Click again on the pager quadrant that was active when you launched the program. The new window reappears.

If you look carefully at the quadrants, you'll see that they provide a bird's eye view of the associated desktop configurations. You may not be able to identify a window from its pager image. But, you can easily distinguish an empty virtual desktop quadrant from one containing one or more windows.

5.1.1.2 Home Directory icon

The Home Directory icon, which resembles a house superimposed on a file folder, is normally located in the upper-left corner of the display. The icon provides a convenient way to access the file manager: double-clicking the icon with the left mouse button launches Nautilus, GNOME's browser and file manager, which displays the contents of the user's home directory.

5.1.1.3 Drive icons

If you have permission to mount a CD-ROM or floppy drive, and media is present, your desktop includes an icon representing the drive. Depending on the type of drive, the icon may resemble a CD-ROM or a floppy. If you right-click on the icon, a pop-up menu appears. The menu lets you unmount and eject the media, or open Nautilus to view the files residing on the device. You can also simply double-click to launch Nautilus and view the contents of the media.

5.1.1.4 Start Here icon

By double-clicking the Start Here icon, you can launch Nautilus to view a folder that contains several useful icons. Double-clicking any icon in the folder launches a window containing icons that provide access to GNOME facilities. You can access the same facilities by using the GNOME menu. The icons and a summary of the operators each offers appear in the following list:

Applications

The Applications icon lets you launch various applications.

Preferences

The Preferences icon provides access to a folder containing icons that enable you to view and modify a variety of preferences, including those for the desktop, document handlers, user interface look and feel, multimedia, and peripherals.

System Settings

The System Settings icon provides access to tools for viewing and modifying the system configuration.

5.1.1.5 Trash icon

The Trash icon lets you view files that have been deleted by using Nautilus. Files deleted by using the shell's **rm** command are not stored in the trash; they are immediately deleted. Simply double-click the icon, and GNOME launches Nautilus to view the folder where deleted files are stored. To restore a deleted file, you can drag it to a new location. You can permanently delete files stored in the Trash by right-clicking the Trash icon and selecting Empty Trash Bin from the pop-up menu.

5.1.2 The GNOME Panel

By default, the GNOME panel appears along the bottom edge of the display. However, if you prefer a different location, you can move the panel; to do so, click and drag the panel to the desired location. The panel functionally resembles the Windows taskbar; you can use it to launch programs, switch from one program to another, and perform other tasks.



Moving the mouse cursor over an icon in the panel reveals a message informing you of the function of the icon.

The panel can also contain *applets*, programs represented as panel icons. Applets are typically small programs that display information or take action when clicked. For example, a *launcher applet* launches an application when its button in the panel is clicked. If you right-click the panel and then choose Help from the pop-up menu, GNOME launches Nautilus to view the GNOME Panel Manual. The GNOME Panel Manual explains the function and operation of the GNOME panel in detail.

By default, the GNOME panel contains the following items:

Main menu

The main menu icon resembles a red hat. Left-clicking the main menu icon presents a menu from which you can choose a variety of programs. Several of the menu entries are submenus; selecting such an entry pops up a new menu to the side of the original entry.

Web browser

Launches the Mozilla web browser.

Email

Launches the Evolution email client, described in [Chapter 6](#).

OpenOffice Writer

Launches the OpenOffice word processor, described in [Chapter 6](#).

OpenOffice Impress

Launches the OpenOffice presentation creator, described in [Chapter 6](#).

OpenOffice Calc

Launches the OpenOffice spreadsheet, described in [Chapter 6](#).

Print Manager

Manages printers and documents queued for printing, described in [Chapter 9](#).

Pager

Described earlier in [Section 5.1.1.1](#).

Task list

The task list contains a button for each active task. Clicking a task's button raises the task's window to the front of the screen, so that you can view it.

Volume Control

Lets you adjust the level of sound; appears only if your system has a sound adapter.

Alert Notification Tool

Alerts you when errata or updates are available.

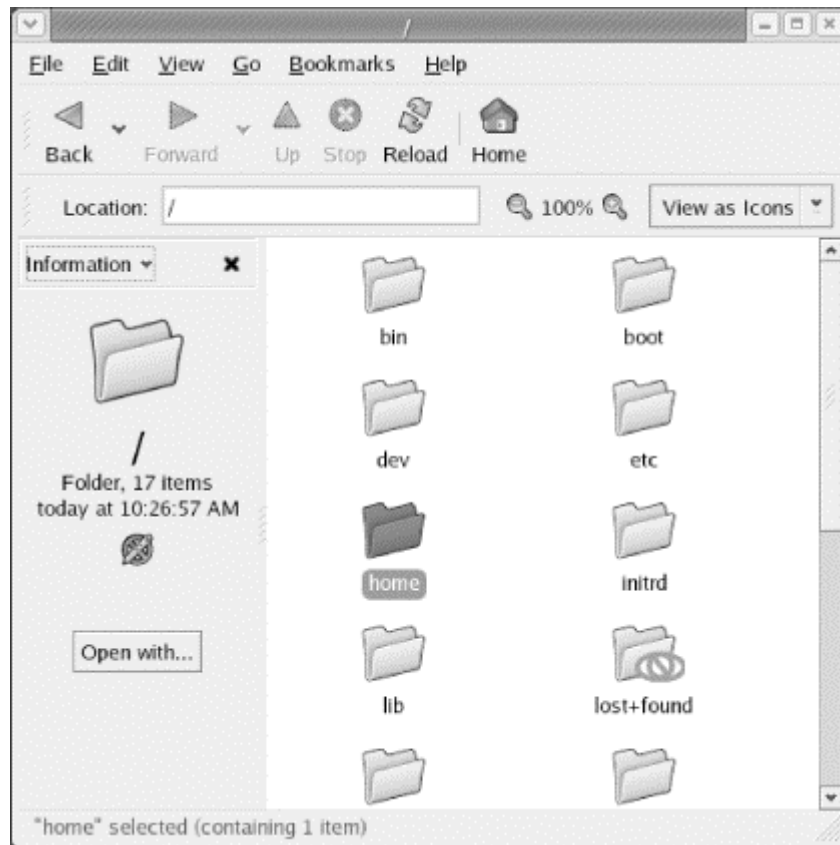
Clock

The clock displays your system's current time.

5.1.3 Using Nautilus

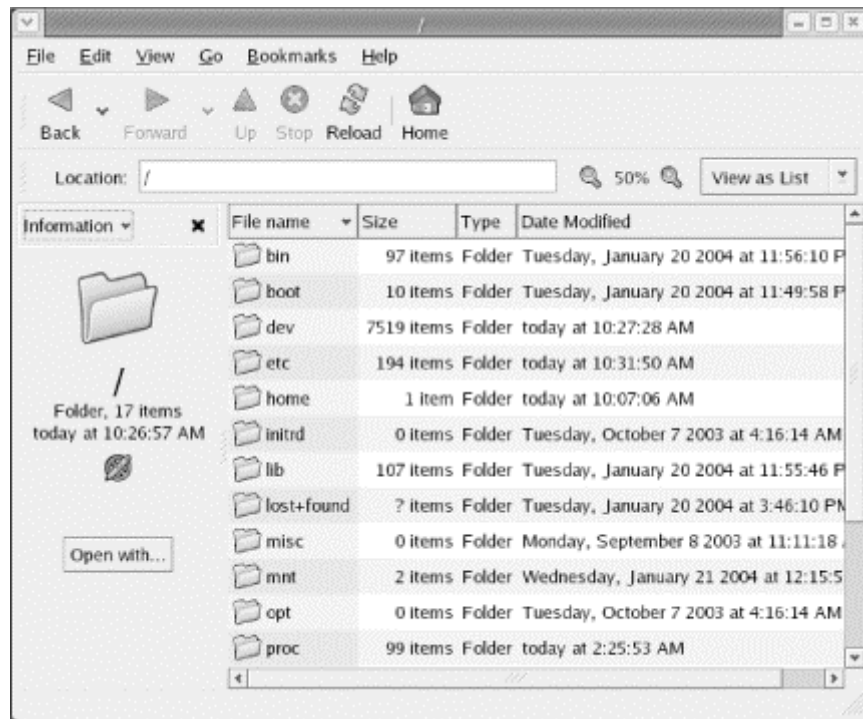
To launch the GNOME file manager, double-click the Start Here icon, or a folder or drive icon if one is visible. You can use the GNOME file manager, named Nautilus, to browse any drive, folder, or administrative interface. Nautilus has two main panes, as shown in [Figure 5-5](#). If the left pane is not visible, click View → Side Pane. The left pane may display information about the current folder or a hierarchical folder tree. To switch between these views, click the "View as Icons" drop-down list and select "View as List". To select a folder in the left pane, simply left-click it. The right pane then shows the contents of the selected folder.

Figure 5-5. Nautilus in icon mode



The right pane can show an icon for each file or detailed information about each file, as shown in [Figure 5-6](#). To switch from icon to detailed mode, left-click the control labeled "View as Icons" or "View as List". You can also select custom mode, which lets you tailor the display appearance according to your own taste.

Figure 5-6. Nautilus in list mode



To view the contents of a folder shown in the right pane, simply double-click the folder's icon. To view the contents of the folder that contains the current folder, click the Up toolbar icon.

Nautilus can be used as a file manager to copy, move, rename, and delete files. To perform one of these operations, you must first select the file (or files) by left-clicking in the right pane. To select additional files, hold down the **Ctrl** key as you select individual files. The Edit menu includes a menu item that lets you click on all files that appear in the right pane. Nautilus provides many ways to perform common file manager operations. Here are a few of the most important ones:

- To move a file, simply drag it to its new location.
- To copy a file, hold down **Ctrl** while dragging it. Alternatively, you can right-click on a file and use the pop-up menu to specify the action you want to perform. Nautilus then displays a dialog box that lets you specify additional options.
- To rename a file, right-click on the file's icon and select Rename from the pop-up menu. Simply type the new name and press Enter.
- To delete a file, right-click on the file and select Move to Trash from the pop-up menu.

If you keep Nautilus's left pane open, operations such as the preceding can be performed conveniently. For instance, if you want to move a file from one folder to another, list folders in the left pane and navigate to the folder containing the file in the right pane. Nautilus lets you double-click on a file to launch the application associated with the file. Alternatively, you can right-click on the file and select Open With from the pop-up menu. Nautilus launches a dialog box that lets you specify the application that should be launched.

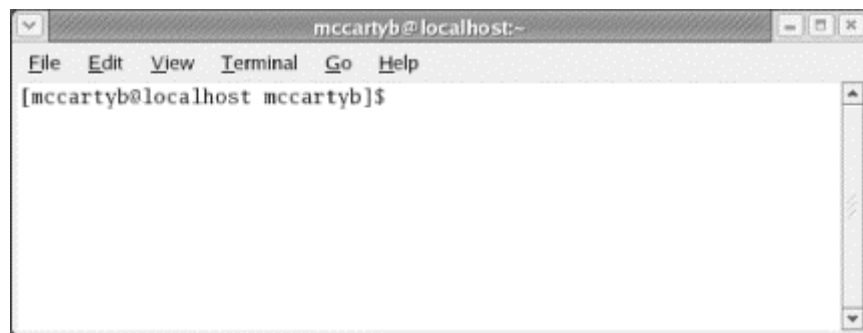
Many applications are GNOME-compliant, supporting drag-and-drop operations like those supported by Windows. For example, you can open two file manager windows and drag-and-drop files or folders between them.

The Nautilus menus provide additional functions, including the ability to configure the operation of Nautilus. If you're familiar with the Windows Explorer, you'll find most of these functions and capabilities familiar. To learn more about Nautilus, use the Nautilus Help menu to view the Nautilus Quick Reference.

5.1.4 Using GNOME Terminal

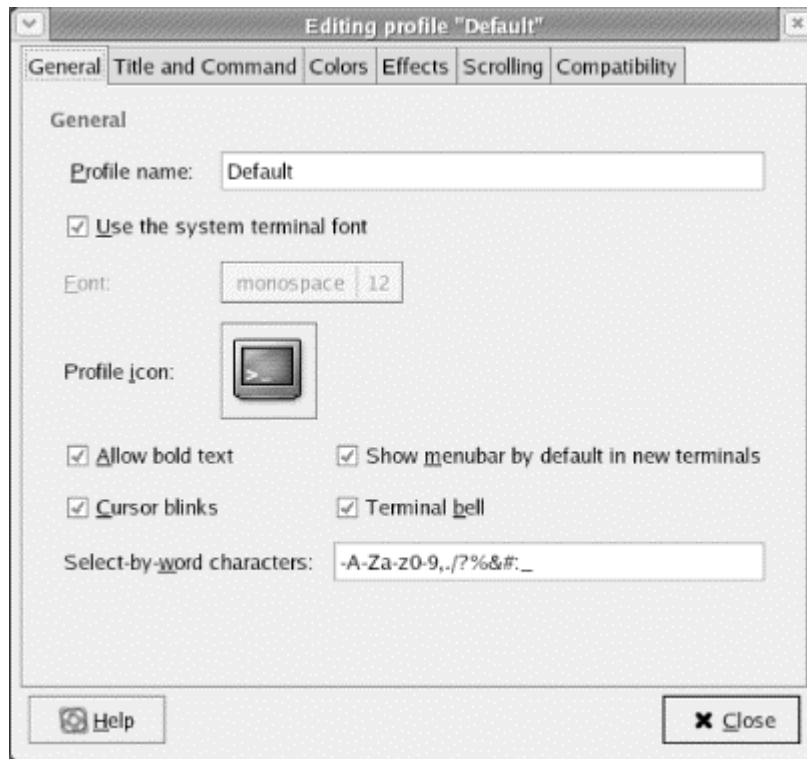
Similar to the MS-DOS Prompt window, the GNOME terminal ([Figure 5-7](#)), provides a window in which you can type shell commands and view their output. To launch GNOME terminal, right-click the desktop and select New Terminal from the pop-up menu. You can open multiple GNOME terminal windows if you like.

Figure 5-7. The GNOME terminal application



The Edit → Current Profile menu item lets you configure the operation of GNOME terminal ([Figure 5-8](#)). For example, you may find that the default font is too large or too small for your liking. From the Editing profile dialog box, select the General tab and disable the checkbox labelled Use same font as other applications. Click the name of the current font and a second dialog box pops up, from which you can select the font, font style, and font size you prefer.

Figure 5-8. Editing terminal settings



To exit GNOME terminal, simply type **exit** on the command line and press **Enter**. Alternatively, select Close Window from the File menu or simply type **Ctrl-D**.

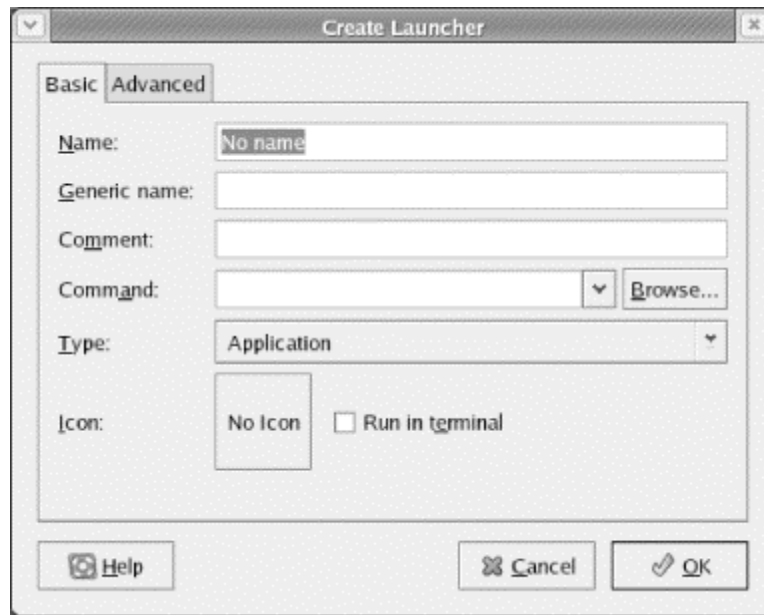
5.1.5 Configuring GNOME

Like most GNOME applications, GNOME itself is highly configurable. You can configure GNOME's panel, its main menu, and its overall appearance and function. The following sections briefly show you how.

5.1.5.1 The GNOME panel

Clicking on a launcher launches a predetermined application. You can easily add a launcher to the GNOME panel. To do so, right-click on the panel and select Add to Panel → Launcher from the pop-up menu. The Create Launcher dialog box ([Figure 5-9](#)) appears.

Figure 5-9. Creating a launcher



You can specify a name for the launcher, a comment, and the command that GNOME executes to launch the application. GNOME automatically provides a default icon. If the program to be run is a text-based program, check the Run in Terminal checkbox. If an application is already on the main menu, you can quickly create a launcher for it. Simply right-click on the application's menu item and select "Add this launcher to panel" or "Add this applet as a launcher to panel" from the pop-up menu. If your panel contains many launchers, it may become crowded and confusing. To remedy this, you can create one or more drawers, like that shown in [Figure 5-10](#). Drawers act like menus; you click on a drawer to open it and view the launchers it contains. Clicking an open drawer closes it and hides its contents.

Figure 5-10. A drawer



To add a drawer, right-click on the panel and select Add to Panel → Drawer from the pop-up menu. To move a launcher into the drawer, click on the drawer to open it, right-click on the launcher, and select Move from the pop-up menu. Next, move the cursor over the open drawer and click the left mouse button to drop the launcher on the rectangular box above the drawer icon, not the drawer icon itself. If you add a launcher or drawer and later decide you don't want it, you can remove it from the panel. Simply right-click on the unwanted launcher and select Remove From Panel from the pop-up menu, as shown in [Figure 5-11](#).

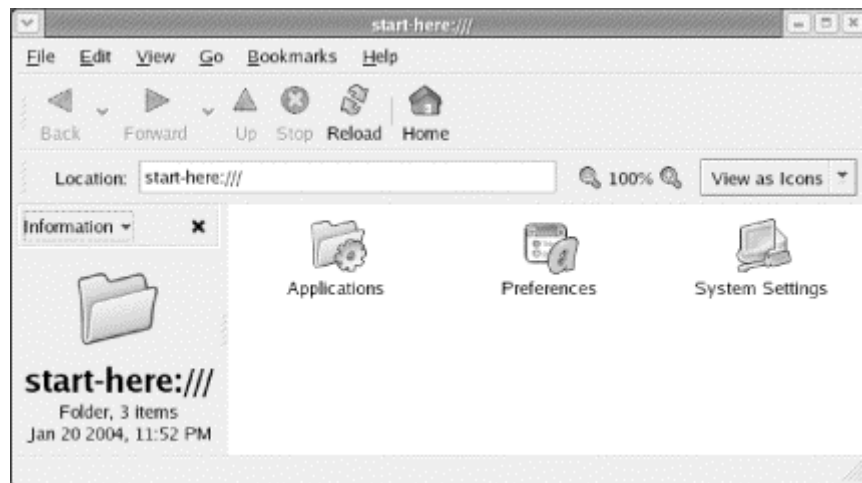
Figure 5-11. Removing a panel item



5.1.5.2 The GNOME Start Here facility

You can configure the appearance and operation of GNOME and GNOME-compliant applications by using the GNOME Start Here facility, shown in [Figure 5-12](#). The Start Here facility resembles the Windows Control Panel, although it looks and works somewhat differently. To launch the Start Here facility, click the Start Here icon on the desktop or panel.

Figure 5-12. The GNOME Start Here facility



You can use the Start Here facility to perform tasks such as:

- Selecting background properties
- Configuring a screensaver
- Selecting a desktop theme
- Configuring the default web browser, text editor, and terminal application
- Specifying MIME types that control the handling of multimedia files
- Configuring the keyboard bell and sounds
- Configuring keyboard and mouse properties
- Specifying applications that GNOME automatically launches when it starts
- Specifying a variety of options governing the appearance of GNOME-compliant applications

Simply select the configuration category by double-clicking the appropriate folder in the Start Here window. Like the Windows Control Panel, the Start Here facility uses small

programs called *applets* to perform its functions. A Start Here window generally contains a mixture of applets and folders.

When you double-click on an applet, a configuration dialog box appears. You can then revise the configuration parameters by specifying the desired values.

5.1.6 GNOME Resources

Several books have been written on GNOME. This chapter has described only a small fraction of what GNOME can do. The following additional resources are available:

GNOME User's Guide for GNOME 2.2

<http://www.gnome.org/learn>

The official guide to using GNOME, available via the GNOME help system.

The GNOME FAQ

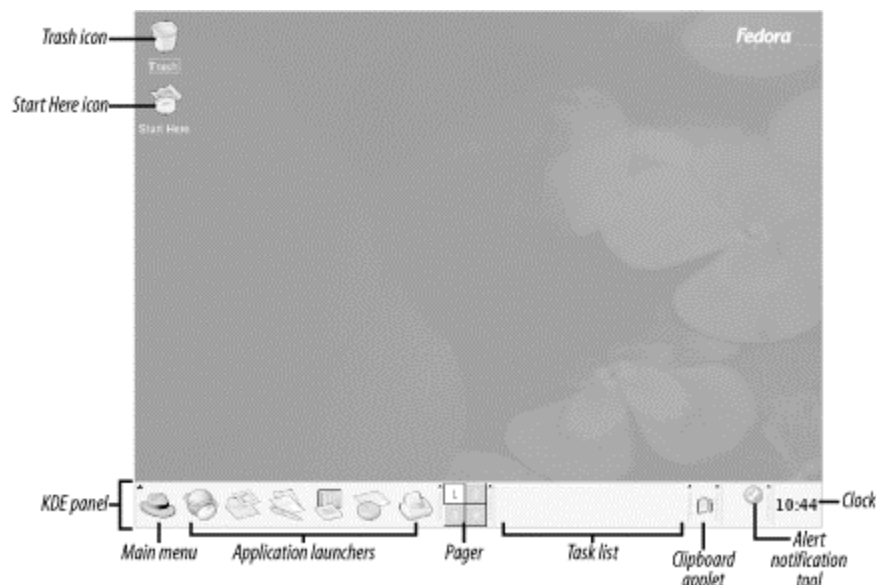
<http://www.gnome.org/faqs/users-faq>

Written by Telsa Gwynne, the GNOME FAQ provides answers to some of the most commonly asked questions concerning GNOME. However, it's somewhat out of date.

5.2 Using the KDE Desktop

As explained at the beginning of this chapter, Red Hat Linux initially included only GNOME; however, Red Hat Enterprise Linux and Fedora Core support both GNOME and KDE. [Figure 5-13](#) shows KDE's desktop. If your system is configured to use GNOME and you want to launch a KDE session, select KDE from the Session menu of the system login screen. Of course, KDE must be installed in order for this to work.

Figure 5-13. The KDE desktop



5.2.1 The KDE Desktop

KDE has a main menu icon in its panel, at the lower left of the screen. The icon is identical to that associated with GNOME's main menu, a red hat. Clicking the icon reveals a menu that includes a Logout menu item. You can use the Logout menu item to terminate KDE. Right-clicking the KDE desktop causes a pop-up menu to appear. From this menu, you can create desktop shortcuts and perform a variety of other functions. The desktop includes a variety of icons and folders. The specific icons and folders that appear may vary depending on the software installed on your system and your KDE configuration. The most common icons are described in the following subsections.

5.2.1.1 Start Here icon

By double-clicking the Start Here icon, you can launch Konqueror, KDE's file manager, to view a folder that contains several useful icons. Double-clicking any icon in the folder launches a window containing icons that provide access to KDE facilities. You can access the same facilities by using the KDE menu. The icons within the Start Here folder include:

Applications

The Applications icon lets you launch various applications.

Preferences

The Preferences icon provides access to a folder containing icons that enable you to view and modify a variety of preferences, including those for the desktop, document handlers, user interface look and feel, multimedia, and peripherals.

System Settings

The System Settings icon provides access to tools for viewing and modifying the system configuration.

5.2.1.2 Home Directory icon

The Home Directory icon enables you to view your home directory by using KDE's file manager, Konqueror. This icon may not appear on your desktop.

5.2.1.3 Drive icons

If you have permission to mount a CD-ROM or floppy drive, your desktop includes an icon representing the drive. If you click the icon, a pop-up menu appears. If your system is configured to do so, it will automatically mount media. However, you can use the menu to manually mount or unmount media. Right-clicking the drive icon lets you eject or unmount the media. These icons may not appear on your desktop.

5.2.1.4 Trash icon

Clicking the Trash icon lets you view the contents of the folder in which KDE stores files moved to the Trash bin by Konqueror. Files deleted by the **rm** command are not stored in the trash; they are immediately deleted.

5.2.1.5 The KDE Panel

KDE's panel normally appears along the bottom edge of the display. However, you can relocate it by dragging it to a side or the top edge of the display. The panel normally contains the main menu icon, launchers, the pager, the task list, the clipboard tool, the alert notification tool, and the clock.



Moving your mouse over an icon in the Panel displays a message informing you of the icon's function.

Main menu icon

As mentioned, the main menu icon features a red hat. Left-clicking the main menu presents a menu from which you can choose a variety of programs. Several of the menu items are submenus; selecting such a menu item pops up a new menu to the side of the original menu item.

Web browser

Launches the Mozilla web browser.

Email

Launches the Evolution email client, described in [Chapter 6](#).

OpenOffice Writer

Launches the OpenOffice word processor, described in [Chapter 6](#).

OpenOffice Impress

Launches the OpenOffice presentation creator, described in [Chapter 6](#).

OpenOffice Calc

Launches the OpenOffice spreadsheet, described in [Chapter 6](#).

Print Manager

Manages printers and documents queued for printing, described in [Chapter 9](#).

Pager

Like GNOME, KDE features a virtual desktop that's larger than your system's monitor. The pager lets you navigate the virtual desktop. By default, one of four virtual desktop pages is visible. The four pager buttons let you select a different desktop page. The button that shows window contents rather than a numeral indicates the page you're currently viewing as your desktop. To view a different page, simply left-click the button that represents the desktop page you want to view.

Task list

The task list contains a button for each active task. Clicking a task's button raises the task's window to the front of the screen so you can view it.

Clipboard Tool

The clipboard lets you view and manipulate the contents of KDE's clipboard, which holds text during copy-and-paste or cut-and-paste operations.

Clock

The KDE clock gives the current time.

5.2.2 Using Konqueror

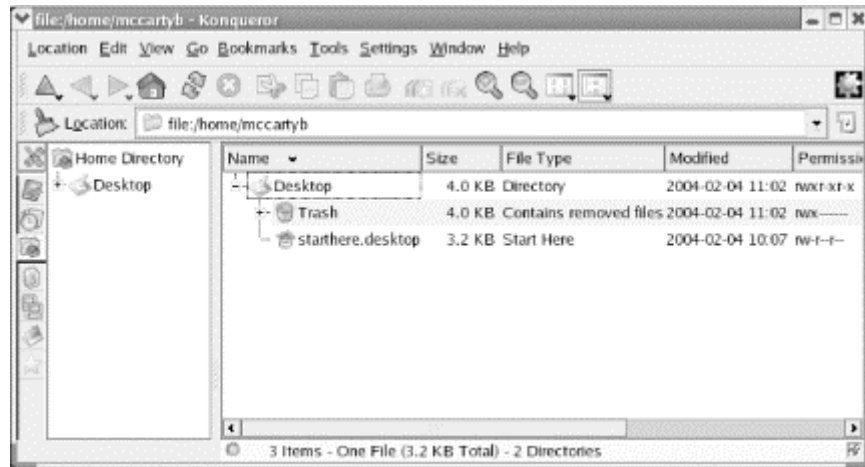
Konqueror is KDE's file manager and web browser. When you click the home directory icon, which resembles a small house superimposed on a larger file folder, Konqueror displays the contents of your `/home` folder ([Figure 5-14](#)). If the home directory icon does not appear, you can launch Konqueror by clicking the Start (red hat) icon and then clicking Home.

Figure 5-14. Konqueror displaying the contents of a folder



By clicking the Tree View icon, which is the rightmost icon on Konqueror's toolbar, you can cause Konqueror to display information in a format that resembles the familiar two-pane layout used by the Microsoft Windows Explorer and GNOME's Nautilus. [Figure 5-15](#) shows Konqueror in Tree View.

Figure 5-15. Konqueror's detailed mode in Tree View



Konqueror can be used to move, copy, rename, and delete files and folders. You can perform these and other file operations in a variety of ways. To rename a file, right-click on the file's icon and select Rename from the pop-up menu. Simply type the new name and press **Enter**. To delete a file, right-click on the file and select Delete from the pop-up menu. A dialog box asks you to confirm your decision.

You can move, copy, or delete multiple files in a single operation. Select the files by holding down the **Ctrl** key as you select them one at a time. Alternatively, you can click and drag the cursor around a group of files. To move or copy the selected files, simply drag them to the new location. When you release the mouse, a pop-up menu lets you specify whether you want to move or copy the files.

Rather than move or copy a file, you can use the pop-up menu to create a link. Konqueror lets you click on a link to launch an application on the file associated with the link. Alternatively, you can right-click on the file or link and select Open With from the pop-up menu. KDE launches a dialog box that lets you specify the application that should be launched.

5.2.3 Using KDE Terminal

Similar to the MS-DOS Prompt window, the KDE terminal, also known as Konsole, provides a window in which you can type shell commands and view their output. To launch KDE terminal, right-click on the desktop and select Open Terminal from the pop-up menu. You can open multiple KDE terminal windows if you like.

The Settings menu lets you configure the operation of KDE terminal. For example, you may find that the default font is too large or too small for your liking. If so, select Settings → Font from the KDE terminal window. Then simply select the font size you prefer.

To exit KDE Terminal, simply type **exit** on the command line and press **Enter**.

Alternatively, select Quit from the File menu or type **Ctrl-D**.

5.2.4 Configuring KDE

KDE is highly configurable. This section explains how to use the KDE Panel, the KDE Control Panel, the KDE Control Center, and the KDE menu editor.

5.2.4.1 The KDE Panel

It's simple to add a launcher icon to the KDE Panel. Right-click on the Panel, select Add from the pop-up menu, and choose a program from the menu that appears. To remove a launcher from the Panel, right-click the launcher and select Remove from the pop-up menu. If your panel contains many launchers, it may become crowded and confusing. To remedy this, you can create a child panel, like that shown at the left in [Figure 5-16](#). Right-click the Panel and select Add → Extension → Child Panel. You can move the child panel to a different edge of the screen by dragging its hide panel—the arrowhead that appears at one end of the child panel—to the desired location. Launchers can be added to a child panel, just as you add them to the Panel. To remove a child panel and its contents, right-click the hide button at the end of the child panel and select Remove from the pop-up menu.



The child panel shown in [Figure 5-16](#) includes the Red Hat Network Alert icon, which does not appear by default on the KDE panel of Red Hat Enterprise Linux. The icon is important, because it notifies you of updates, as explained in [Chapter 8](#). Apparently, the absence of the icon from the Red Hat Enterprise Linux KDE panel is an oversight, and may be corrected in an update. However, you can manually add the icon to a panel, as was done before preparing [Figure 5-16](#). To do so, right-click on the desired panel and select Add Application Button → System Tools → Red Hat Network Alert Icon.

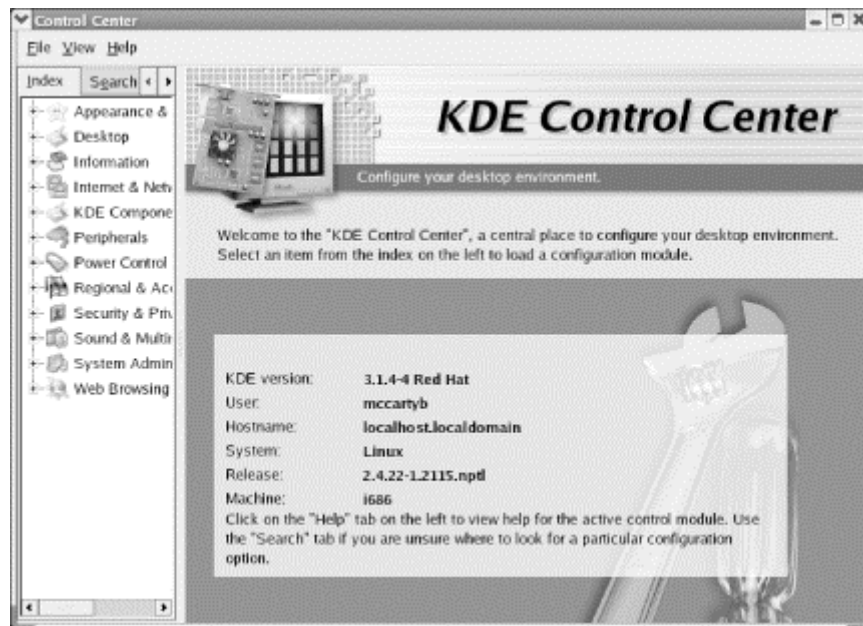
Figure 5-16. A child panel



5.2.4.2 The KDE Control Center

You can launch the KDE Control Center by choosing Control Center from the KDE main menu. [Figure 5-17](#) shows the KDE Control Center.

Figure 5-17. The KDE Control Center



The Control Center user interface features two panes. The left pane presents a hierarchically structured set of configuration categories, and the right pane displays information pertaining to the current choice. Control Center categories include:

Appearance & Themes

Lets you specify colors, fonts, icons, and other desktop attributes

Desktop

Lets you specify behaviors and visual characteristics of the desktop

Information

Lets you browse hardware devices and running services

Internet & Network

Lets you configure your email identity, KDE LAN browser options, network options, and SOCKS proxy client options

KDE Components

Lets you configure KDE components such as the address book, file associations, Konqueror options, and the spelling checker

Peripherals

Lets you configure a digital camera, keyboard, and mouse

Power Control

Lets you configure power management options

Regional & Accessibility

Lets you configure accessibility options for disabled persons, and internationalization options such as language and keyboard layout

Security & Privacy

Lets you configure identity certificates and passwords, and other cryptographic options

Sound & Multimedia

Lets you configure audio sources, levels, and other options

System Administration

Lets you configure KDE's alarm, login, and printing features

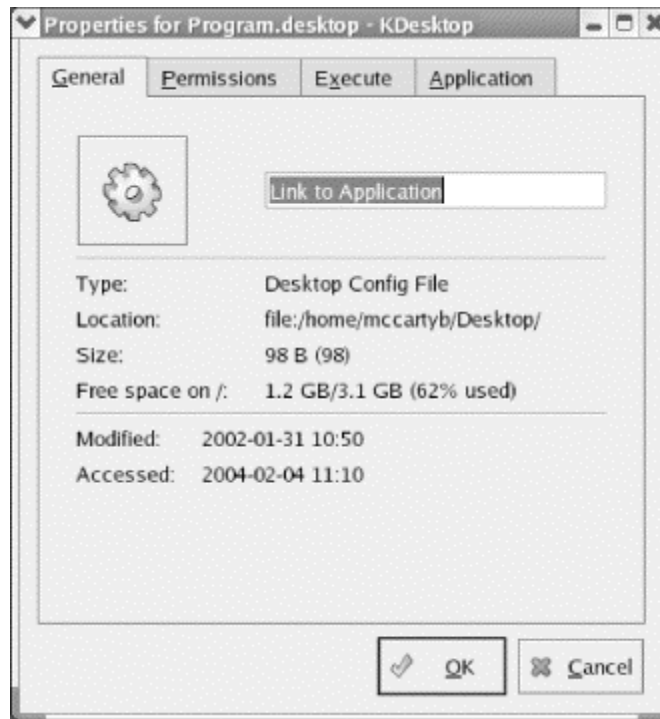
Web Browsing

Lets you configure Konqueror and Mozilla web browser options
Simply select the configuration category by clicking in the left pane. You can then revise the configuration parameters by specifying the desired values in the right pane. The contents of the right pane vary depending on the current selection in the left pane.

5.2.4.3 Adding an application link

You can easily add to your desktop an icon called an *application link* that lets you launch an application with a double-click. To do so, right-click the desktop and select Create New → Link to Application from the pop-up menu. The "Properties for Program.desktop" dialog box ([Figure 5-18](#)) appears.

Figure 5-18. The Properties for Program dialog box



In the General tab, type a name for your link, replacing the text "Link to Application." Then, click the Execute tab and the Browse button. An Open dialog box appears. Use it to navigate to the program file you want to launch, click to select the file's icon, and click OK. The "Properties for Program.desktop" dialog box reappears. If the program you chose isn't an X program, enable the checkbox titled "Run in terminal." Finally, click OK to close the "Properties for Program.desktop" dialog box.

Move the application link icon to a suitable location. Now, you can launch the configured program simply by double-clicking the icon. Who ever said that Linux is hard to use?

5.2.5 KDE Resources

Entire books have been written on using KDE, so this chapter has provided a mere overview of KDE's many features and facilities. The following additional resources are available via KDE's help information function:

The Konqueror Handbook

<http://docs.kde.org/en/3.1/kdebase/konqueror>

An overview of Konqueror.

KDE Quickstart Guide: An Introduction to KDE

<http://www.kde.org/documentation/quickstart/index.html>

A fast-paced introduction to KDE's most important features.

KDE User's Guide

<http://www.kde.org/documentation/userguide/index.html>

The most complete KDE reference: "a detailed overview of the basic workings of KDE."

The KDE FAQ

<http://www.kde.org/documentation/faq/index.html>

Answers to some of the most commonly asked questions concerning KDE can be found online.

Use these resources to learn more about KDE. Also, visit the KDE web site, <http://www.kde.org>. There you'll find more information—and more current information—about KDE and the KDE project.

Chapter 6. Using Linux Applications

Red Hat Enterprise Linux and Fedora Core include a plethora of applications. This chapter introduces you to several of the most popular and useful applications. These include: OpenOffice.org, a desktop suite; Evolution, an email client and personal scheduler; and Kpilot, an application for syncing a personal digital assistant (PDA) with your system. This chapter also explains how to use the Nautilus file manager to burn CDs.

6.1 OpenOffice.org

OpenOffice.org is a desktop suite that functionally resembles Microsoft Office. That is, OpenOffice.org can perform many of the functions performed by Microsoft Office and includes many of the familiar features of Microsoft Office, along with a few features not found in Microsoft Office. The distinctive advantage of a desktop suite is that its component applications are designed to work together. The applications of a desktop suite have a similar look and feel, which makes them easy to learn and use.

Linux users have long had access to applications that help them prepare documents. However, development of Linux desktop suites has lagged behind that of Microsoft Office. The applications and suites have tended to be somewhat clumsy to use, unreliable, and poor in features. OpenOffice.org sets a new standard for Linux desktop suites, providing features and capabilities that are adequate to satisfy most computer users, not merely Linux fans. OpenOffice.org began as a commercial desktop suite known as StarOffice, created by StarDivision. When Sun Microsystems acquired StarDivision in 1999, Sun soon thereafter released a freely available version of StarOffice. More recently, Sun has made certain StarOffice technologies available to the open source community, which created the freely redistributable OpenOffice.org desktop suite. Sun plans to continue development of StarOffice, which is a component of Sun's Java Desktop System, as a commercial product. At the same time, the open source community plans to continue development of OpenOffice.org.

OpenOffice.org is a multi-platform product, and is currently available for Linux, PPC Linux, Solaris, Windows, and Mac OS X (under Apple's X11). Work is underway to support other platforms, including FreeBSD, OpenVMS, and IRIX. OpenOffice.org is also a global product, currently supporting 32 languages. Support for new languages is added regularly.

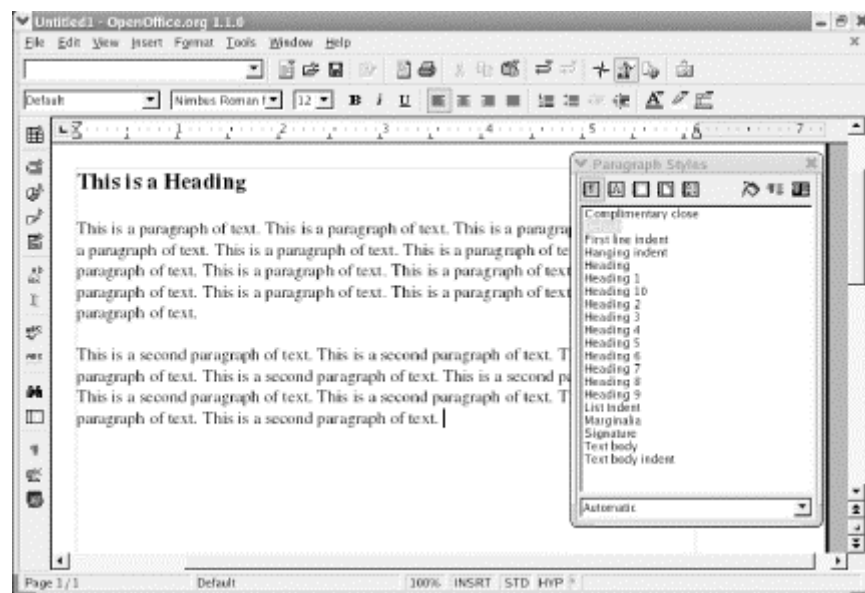
OpenOffice.org includes translation filters that let you share documents with users of Microsoft Office and other popular applications. It also includes convenient features such as Print to PDF (Adobe Portable Document Format); AutoPilot, which assists you in creating complex documents; and Stylist, which helps you take control of the look of your document.

OpenOffice.org includes word processor, spreadsheet, graphics, presentation manager, and drawing applications. The next several sections describe these applications. You can learn more about OpenOffice.org at <http://www.openoffice.org>.

6.1.1 Writer: The OpenOffice.org Word Processor

Writer is available via the Office menu item of the GNOME and KDE menus and, still more conveniently, has its own panel icon, which resembles a pen superimposed on two sheets of paper. When you launch Writer, you'll see a window resembling that shown in [Figure 6-1](#).

Figure 6-1. Writer: the OpenOffice.org word processor



If you're launching Writer for the first time, a dialog box invites you to register as an OpenOffice.org user. Your registration is invited—and appreciated by the developers of OpenOffice.org—but not required.



Red Hat Enterprise Linux ships with Version 1.0.2 of OpenOffice.org, while Fedora Core ships with the newer Version 1.1.0. This chapter shows and describes OpenOffice.org Version

1.1.0. So, if you're using Red Hat Enterprise Linux, you may note minor differences between screen shots or descriptions in this chapter and the appearance or behavior of your system.

If—as is likely the case—you've used Microsoft Word or another word processor, you'll find Writer's user interface intuitive and easy to use. You can type text in the middle area of the window. To style text, highlight the text and double-click on a style in the Stylist. The stylist is the window titled Paragraph Styles that appears in [Figure 6-1](#). If the Stylist is not visible, you can summon it by pressing F11 or choosing Format → Stylist.

As mentioned, Writer includes many impressive features, some of which are available to users of Microsoft Office only at significant cost. For instance, Writer makes it simple to create PDF documents. To create a PDF document, prepare and save your document as usual. Then choose File → Printer Settings. The Printer Setup dialog box appears, as shown in [Figure 6-2](#).

Figure 6-2. The Printer Setup dialog box



Choose PDF converter from the Name list box and click OK. Then, choose File → Print. When the Print dialog box appears, review the Options and Properties as necessary and then click OK. The PDF document is created and stored as a file in your home directory. Launch Nautilus or another file browser and click on the file to view the PDF document. Here are some other Writer features you may enjoy exploring and using:

AutoPilot

Choose File → AutoPilot. From the submenu, choose the type of document you want to compose, such as a fax, agenda, web page, or form. AutoPilot will assist you in composing the document.

Address Book

Choose File → Template → Address Book Source. Writer can access your Netscape or Mozilla address book, an LDAP or SQL database, or other data sources and obtain names, addresses, and other information.

Navigator

Choose Edit → Navigator. The Navigator helps you find bookmarks, sections, hyperlinks, references, indexes, and notes. It's especially helpful when working on a large document.

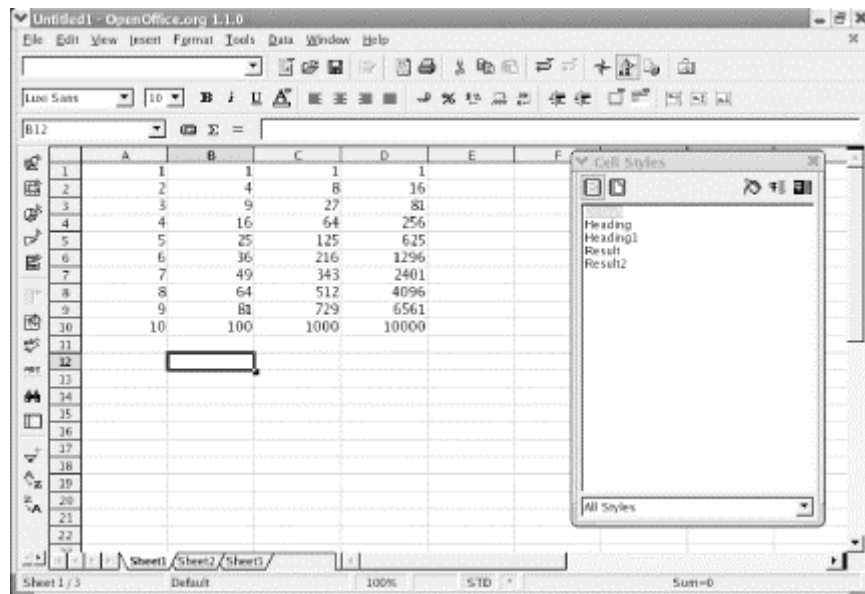
To sample Writer's ability to work with documents prepared using Microsoft Word, I opened a draft version of chapter 4—which was converted from DocBook to Microsoft Word—in Writer. In editions of this book published prior to the advent of OpenOffice.org, Linux word processors were not fully up to this challenge. They generally mangled the manuscript and sometimes crashed.

Writer, however, did not balk. The only problem I found was that several unnumbered lists were erroneously converted to numbered lists. This problem took only a few seconds to fix.

6.1.2 Calc: The OpenOffice.org Spreadsheet

Like Writer, Calc is available via the Office menu item of the GNOME and KDE menus. It too has a convenient panel icon, which resembles a pie chart superimposed on a tabular spreadsheet. Calc's main window appears in [Figure 6-3](#).

Figure 6-3. Calc's main window



Just as Writer resembles Microsoft Word, Calc resembles Microsoft Excel. In particular, the language used in writing cell formulas is similar to that used by Excel. Moreover, the resemblance between the products is not merely syntactic. If your fingers are so accustomed to Excel that they race to perform operations in the familiar way, disaster will

not likely ensue: many of Calc's keyboard shortcuts are identical to those used by Excel. You may need to look up from the keyboard to realize you're using Calc rather than Excel. Like Excel, Calc can produce tables, graphs, and charts. Like other OpenOffice.org applications, Calc can send your work product to a printer or write it as a PDF document. Again, like Writer, Calc can obtain data from databases and other external sources. Choose View → Data Sources to view available data sources. To configure a new data source, right-click in the left pane of the window that appears. To return to the standard view, choose View Data Sources a second time. Here are some interesting Calc features for you to explore:

Detective

Select a cell. Choose Tools → Detective. Then, choose from the submenu. Calc will show you cells that determine the value of the selected cell (Precedents), cells with values determined by the selected cell (Dependents), or other relationships.

Goal Seek

Select a cell. Choose Tools → Goal Seek. Specify a target value. Specify a precedent cell as the Variable cell. You can do so by typing the cell's name or by clicking the cell. Click OK. Calc determines the value of the variable cell that causes the selected cell to have the specified target value and offers to enter the value in the variable cell.

Subtotals

Select a block of cells in which the left column designates an entity (such as a department) and the right column or columns designate characteristics of the entity (such as sales, sales returns, and net sales). Ideally, the top row of the block should contain a name for each column. Choose Data → Subtotals. On the 1st Group tab, enable the checkboxes associated with the right column or columns. Click the left column name and choose the function count. Click each right column name and choose the function Sum. Click OK. The block now includes subtotals and a grand total.

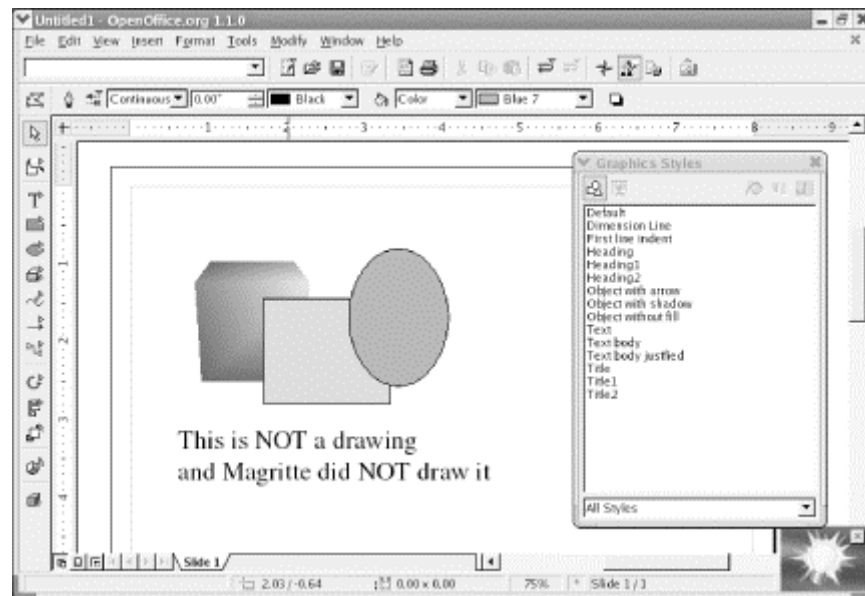
Input Validation

Select a cell. Choose Data → Validity. On the Criteria tab, specify rules that govern valid cell values. On the Error Alert tab, enable Show error message when invalid values are entered. Optionally, specify an action, title, and error message. Click OK. Enter an invalid value for the cell and press Enter. Calc presents a dialog box warning you that the value is invalid.

6.1.3 Draw: The OpenOffice.org Drawing Program

Draw is the OpenOffice.org drawing program, available via the Office submenu of the GNOME and KDE menus. Unlike its sister programs Writer and Calc, Draw has no panel icon. Launch Draw by choosing Office → Draw from the main menu. [Figure 6-4](#) shows Draw's main window.

Figure 6-4. Draw's main window



Draw enables you to draw two- and three-dimensional objects, and specify their color and other characteristics. Using Draw, you can move, align, and manipulate your objects to arrive at a complete composition. You can use Draw to create graphics that you incorporate into documents prepared using other OpenOffice.org applications, such as Impress, the presentation manager. You can save your work product as a GIF, JPEG, PNG, or in many other popular graphics formats.

The GIMP (Gnu Image Manipulation Program), also included in Red Hat Enterprise Linux and Fedora Core, is a more sophisticated drawing program. However, Draw integrates smoothly with other OpenOffice.org applications. So, you may find it the more convenient tool for graphics of everyday complexity.

Here are some fun operations to try out:

3D object manipulation

3D manipulations are impressive but can be challenging to create. Here's a sample 3D manipulation you can perform. On the drawing toolbar on the left edge, select the 3D tool, which resembles a cube. Don't confuse the tool with the 3D effects tool, which has a different function. The 3D effects tool appears at the bottom of the toolbar. Drag twice within the document window to create two solid cubes. Convert one solid cube to a wire frame cube by choosing Format → Area and specifying None as the Fill and by choosing Format → Line and specifying Continuous as the Style. To see other 3D effects, select the other cube and choose Format → 3D Effects. Experiment with the effects to see what's possible. The Favorites tab is particularly fun when working with solid objects.

FontWork

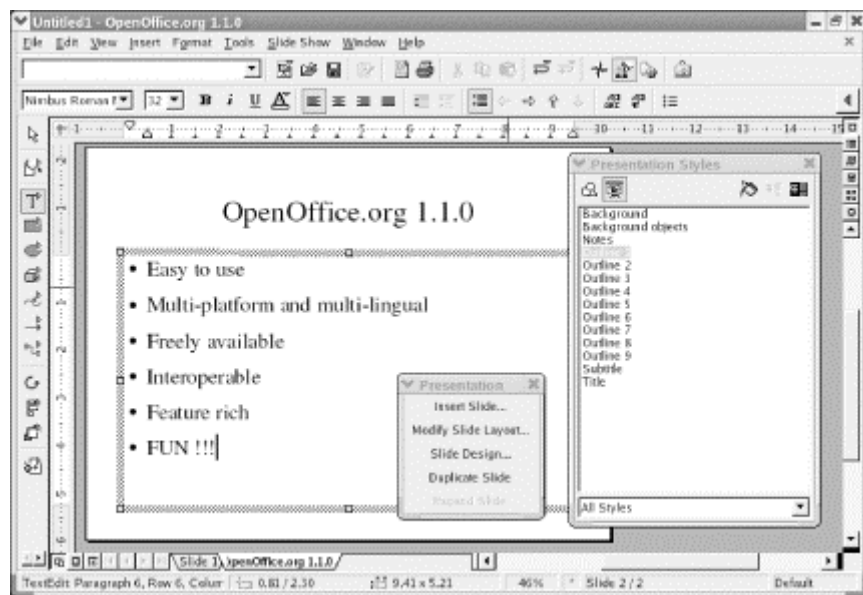
On the drawing toolbar, select the text tool, which resembles the letter T. Click in the document window and type some text. Choose Format → FontWork. Click one of the arcs visible in the list at the top of the FontWork dialog box. The text is bent

in the direction indicated by the arc. Click other buttons and specify other values as desired. As you'll quickly see, you can easily create dynamic and attention-grabbing visuals using FontWork.

6.1.4 Impress: The OpenOffice.org Presentation Manager

Impress is OpenOffice.org's presentation manager, which functionally resembles Microsoft PowerPoint. Impress is available via the Office submenu of the GNOME and KDE menus. Impress has a convenient panel icon, which resembles a slide superimposed on a bar chart. [Figure 6-5](#) shows Impress's main window.

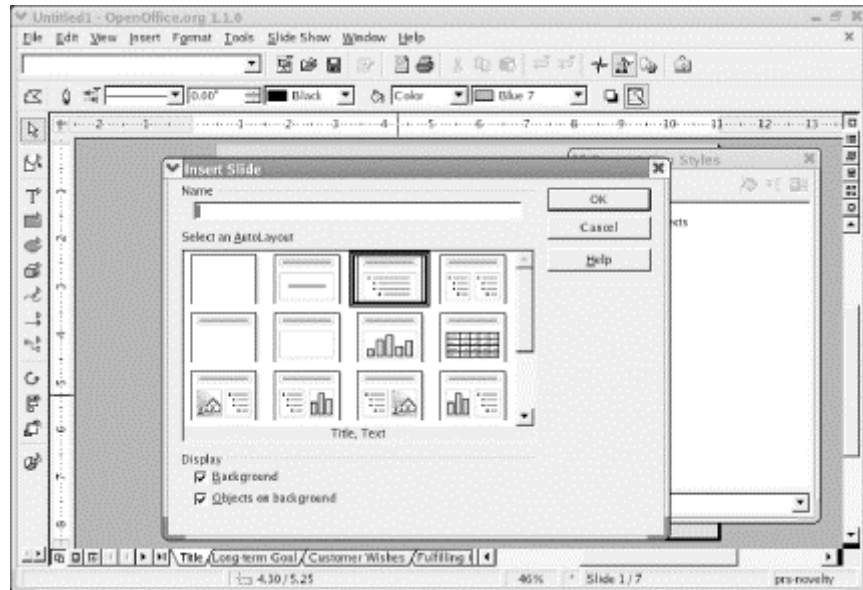
Figure 6-5. Impress's main window



When you launch Impress for the first time, it fires up AutoPilot to lead you through creating a presentation. You can manually launch AutoPilot from the File menu. AutoPilot lets you specify the presentation medium (paper, slides, and so on), and select a presentation template, slide design, and slide transition appropriate for your presentation. Currently, Impress is distributed without templates. A few templates are available on the OpenOffice.org web site, <http://www.openoffice.org>. However, Impress also lets you model a presentation on an existing presentation. Until a variety of templates is available, you'll likely find this capability useful and convenient.

When the AutoPilot is finished, Impress pops up a small dialog box titled Presentation. To create a slide, click Insert Slide, which brings up the Insert Slide dialog box shown in [Figure 6-6](#). This dialog box is the main means of creating the slides that comprise a presentation. Select the desired layout and click OK. An empty slide appears in the window, ready for your customization.

Figure 6-6. The Insert Slide dialog box



Impress is not as featureful as Microsoft PowerPoint. In particular, it lacks support for video clips. However, many experts believe that overly sophisticated presentations can work to the disadvantage of the presenter. Impress is more than adequate to create simple and clear presentations that help you communicate persuasively.

Impress includes the FontWork and 3D Effects tools explained in the section on Draw, as well as other facilities common to many OpenOffice.org applications.

Here's some fun you can have with Impress:

Effects

Create a geometric shape, such as a filled circle, within a slide. Select the shape and choose Slide Show → Effects. Choose the Favorites menu and click one of the effects that appear in the large selection box. Click the Assign button, which resembles a green checkmark. Then, click the Preview button, which appears to the right of the Assign button. A preview window appears. Click the preview window to see how your effect looks. Choose Slide Show → Slide Show to view the slide show. During the slide show, click the slide to trigger the effect. A single slide can contain multiple effects, which play sequentially when triggered, in the order they were created.

Interaction

Create a geometric shape, such as a filled circle. Choose Slide Show → Interaction. Use the Action at mouse click list to associate an action, such as playing a sound, with the shape. Depending on the action you selected, Impress may present additional options from which you can select. When you're satisfied, click OK to exit the Interaction dialog box. Choose Slide Show → Slide Show to view the slide show. During the slide show, click the object—not merely an unrelated part of the slide—to trigger the interaction.

Animation

Create a series of shapes that, when quickly viewed in sequence, will resemble a cartoon. To assemble the shapes into an animation, select the first shape and choose Slide Show → Animation. Click the Apply Object button that appears at the left of the middle row of the Animation dialog box. You can identify the Apply Object button by briefly idling the cursor over a candidate button until the button's label appears. Then, select the second shape and click Apply Object. Repeat this procedure for the third and subsequent shapes. Click the VCR Play button to view your animation. When you're satisfied with the animation, click Create to place it in your slide as an animated GIF. Delete the component shapes and choose Slide Show → Slide Show to view the slide show. When you click the slide containing your animation, the animation will play.

6.2 Evolution

Ximian's Evolution is an email client and personal scheduler. Previously, Evolution was available from Ximian as an add-on to Red Hat Linux. However, conflicts between Ximian's RPM packages and those distributed by Red Hat sometimes made life complicated for Ximian users. Evolution is now part of Red Hat Enterprise Linux and Fedora Core, so Ximian users can expect more trouble-free operation and fewer problems when upgrading to new Linux versions. You can learn more about Evolution at <http://www.ximian.com>.

Evolution has four main functions:

Email client

Receives email from POP and IMAP servers and sends email via SMTP servers.

Calendar

Provides daily and monthly calendars to help you plan your time.

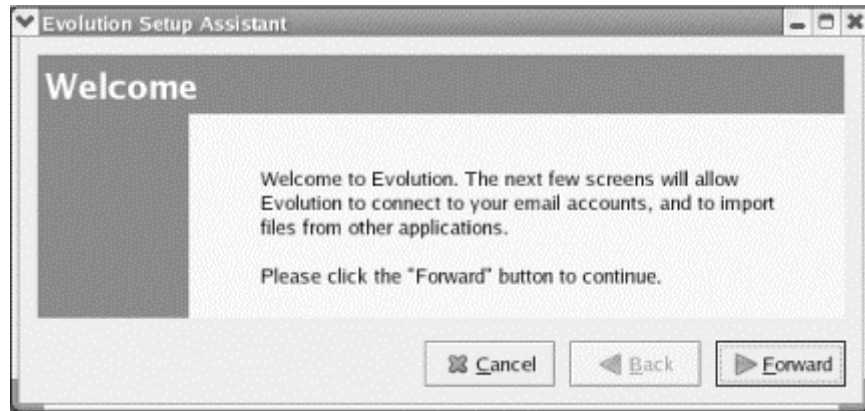
Task list

Provides a to-do list that helps you keep track of projects and deadlines.

Contact database

Provides a contact list that you can conveniently use when composing email. Evolution is available via the Internet → Email submenu of the GNOME and KDE menus. It also has a convenient panel icon, which resembles a postage stamp superimposed on an envelope. When launched for the first time, Evolution provides a wizard to help you configure its operation, as shown in [Figure 6-7](#).

Figure 6-7. The Welcome panel of the Evolution setup wizard



Click Forward to move on to the Identity panel. There, you specify your full name and email address. Optionally, you can specify the name of your organization, a plain text email signature file, and an HTML email signature file.

Click Forward to move on to the Receiving Email panel. There, you specify the type of server you use to receive email, the hostname of the server, your username, and information on the security and authentication characteristics of the server. You can obtain this information from your email provider or determine the correct values by experimentation. Click Forward to move on to a second Receiving Email panel. This panel lets you specify options about checking for new email and using folders and filters. You can specify the values according to your preferences.

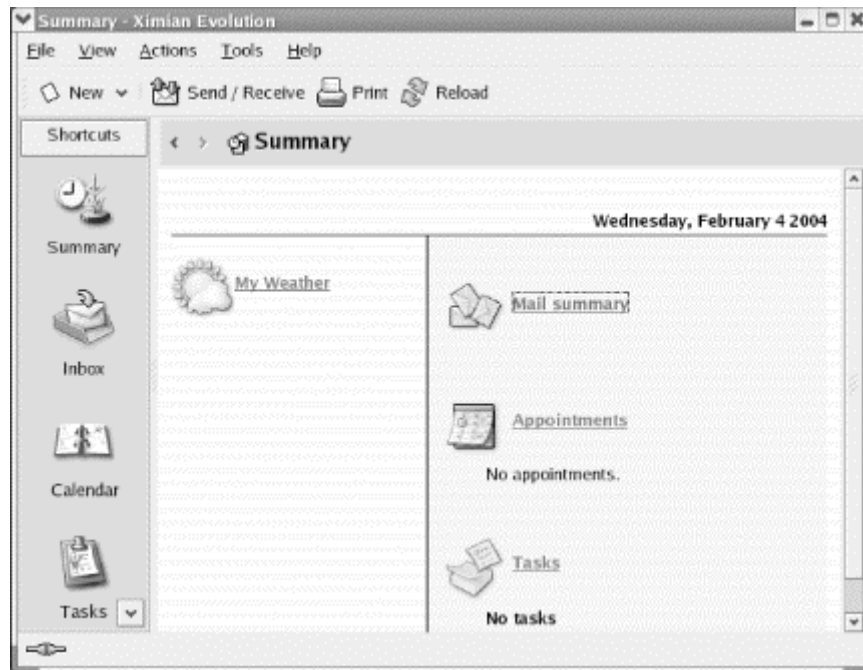
Click Forward to move on to the Sending Email panel. There, you specify the type of server you use to send mail, its hostname, and security and authentication options. You can obtain this information from your email provider or determine the correct values by experimentation.

Clicking Forward takes you to the Account Management panel. This panel lets you associate a name with the account, so that you can distinguish from accounts you may add later. You can also specify whether the account is the default account. If you have only one account, you should specify that it is the default account.

Clicking Forward takes you to the Timezone panel. Click the map or use the drop-down list to specify your time zone.

Finally, clicking Forward takes you to the Done panel. From there, click Apply to save your configuration. At last, Evolution appears as shown in [Figure 6-8](#).

Figure 6-8. Evolution's main window



The left pane of Evolution's window is a menu that lets you choose the contents of the right pane. The Summary pane, which appears initially, presents weather, news, and a status summary of Evolution. Clicking Inbox, Calendar, Tasks, or Contacts selects the corresponding Evolution function.

6.3 Pilot/Handspring Tool

Red Hat Enterprise Linux includes *gnome-pilot*, a tool that lets you synchronize your Palm or Handspring PDA (personal digital assistant) with your Linux system using your system's serial or USB port and your PDA's hot sync cradle. By default, Red Hat Linux is configured to enable you to back up your PDA to your Linux system or restore a backup from your Linux system to your PDA. You can also synchronize Evolution's calendar, task list, and contact list with your PDA.

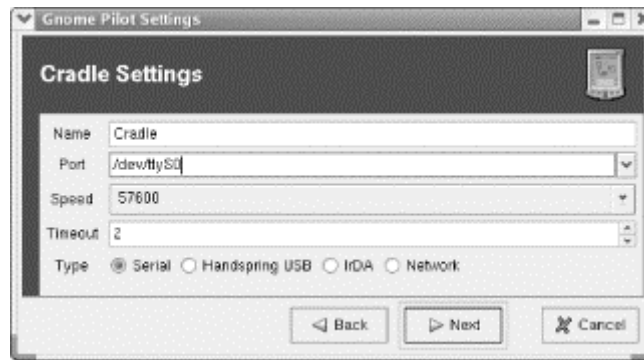
To set up your system to communicate with your PDA, choose **Tools** → **Pilot Settings** from the Evolution menu. The Welcome panel appears, as shown in [Figure 6-9](#).

Figure 6-9. The *gnome-pilot* Welcome panel



Plug your PDA into its cradle and plug the cradle into your system's serial or USB port, according to the type of cradle you're using. Click Forward to continue. The Cradle Settings panel ([Figure 6-10](#)) appears. Specify the port to which the cradle is attached.

Figure 6-10. The Cradle Settings panel

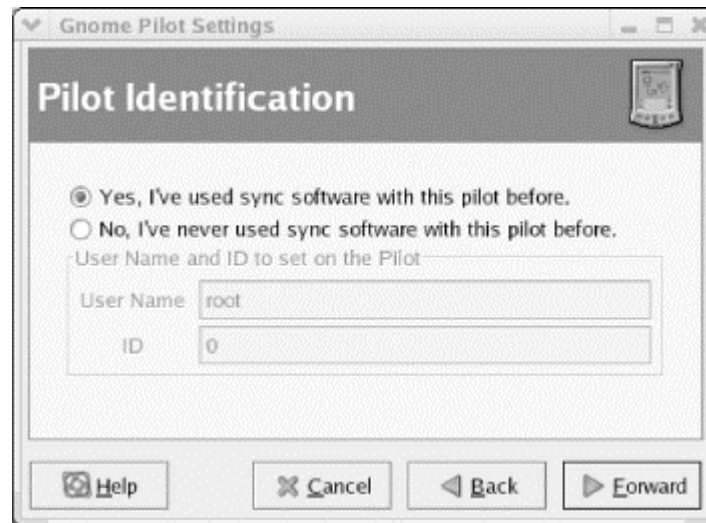


If you've synched your PDA using a Microsoft Windows host, you can use the information in [Table 6-1](#) to determine the serial port. If you're using a USB port, specify a port having the form */dev/ttyUSBn*, where n is the number of the USB port.

Table 6-1. Linux and Windows serial port designations	
Windows designation	Linux designation
COM1	/dev/ttyS0
COM2	/dev/ttyS1
COM3	/dev/ttyS2
COM4	/dev/ttyS3

Click Forward to continue. The Pilot Identification panel appears, as shown in [Figure 6-11](#). If you've not yet performed a hot sync of your PDA using Gnome Pilot, enable the No radiobutton and specify the User name and ID you want to use. Otherwise, enable the Yes radiobutton.

Figure 6-11. The Pilot Identification panel



Click Forward to continue. The Initial Sync panel appears. As directed, press the PDA's HotSync button. If the hot sync setup succeeded, the Initial Sync panel displays the user name and ID. Also, check the PDA's screen to determine if the hot sync setup succeeded. If it failed, use the Back button to return to a previous screen and correct the problem. Also, check the cable connecting the HotSync cradle to the PC.

Click Forward. The Pilot Attributes panel appears. If you like, you can change the Pilot Name and Local basedir values.

Click Forward to perform the hot sync operation and a Success panel should appear.

By default, a hot sync operation merely backs up the data in your PDA. However, you can specify that various Evolution components—such as the Contacts database—should be synchronized with your PDA whenever you perform a hot sync. To do so, select Tools → Pilot Settings → Conduits from the Evolution menu. The Pilot Conduits dialog box appears, as shown in [Figure 6-12](#). Its purpose is to show the available conduits, small programs that let you move information between your system and PDA. Note that five conduits now appear. By default, the conduits are disabled. You can enable a conduit by selecting it and clicking Enable. When enabled, some conduits present a small configuration dialog box ([Figure 6-13](#)). Generally, the dialog box lets you configure the direction of information transfer: either from the PDA to the system or from the system to the PDA. Configure each conduit as you prefer and then click OK to close the Pilot Conduits dialog box.

Figure 6-12. The Pilot Conduits dialog box

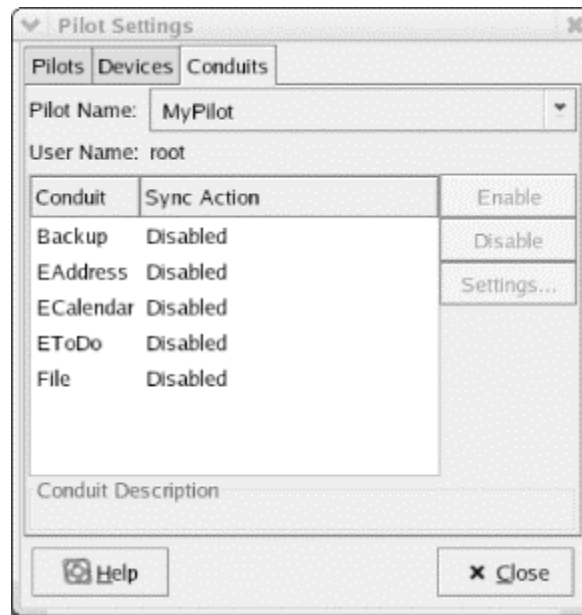
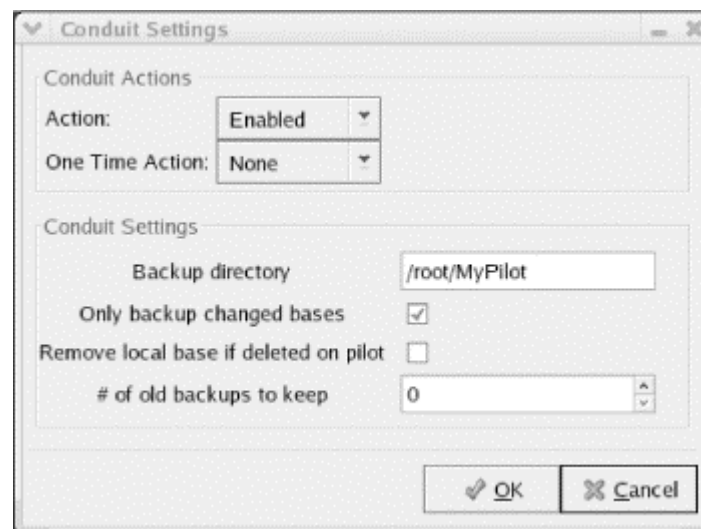


Figure 6-13. A conduit configuration dialog box



The conduits have the following functions:

Backup

Backs up your PDA contents to a file on your system or restores the PDA contents from a file on your system.

EAddress

Sends Evolution's contact list to your PDA or your PDA's contact list to Evolution.

ECalendar

Sends Evolution's calendar to your PDA or your PDA's calendar to Evolution.

EToDo

Sends Evolution's task list to your PDA or your PDA's task list to Evolution.

File

Transfers files from your system to your PDA or from your PDA to your system. You shouldn't enable this conduit unless you have a specific purpose in mind, in which case you should be careful to properly configure it.



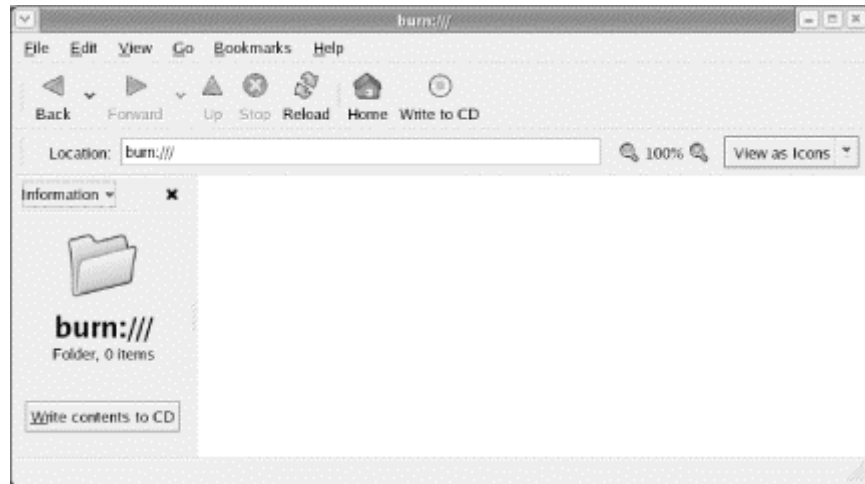
Be sure to properly configure the direction of transfer—the Action—for each conduit. Otherwise, you may destroy important information by performing a transfer in an inappropriate direction. For instance, you could inadvertently transfer old data from your PC onto your PDA, destroying more recent data in the PDA.

To perform a hot sync operation, connect your PDA's cradle to your system's serial port, place the PDA in its cradle, and press the HotSync button on the cradle. The enabled conduits run and transfer information between your PDA and system according to the conduit configurations.

6.4 CD Creator

At one time, writing a CD under Linux was a common rite of passage for new Linux users. The command used to write a CD, **cdrecord**, is a sophisticated command with many options. Getting all the options just right was a challenge for many. However, Creator, a program built into the Nautilus file manager, makes it easy to write CDs. To launch CD Creator, simply insert a blank CD into your CD-R or CD-RW drive. Alternatively, you can launch Nautilus and select Go → CD Creator from the Nautilus menu. Nautilus presents the burn window shown in [Figure 6-14](#).

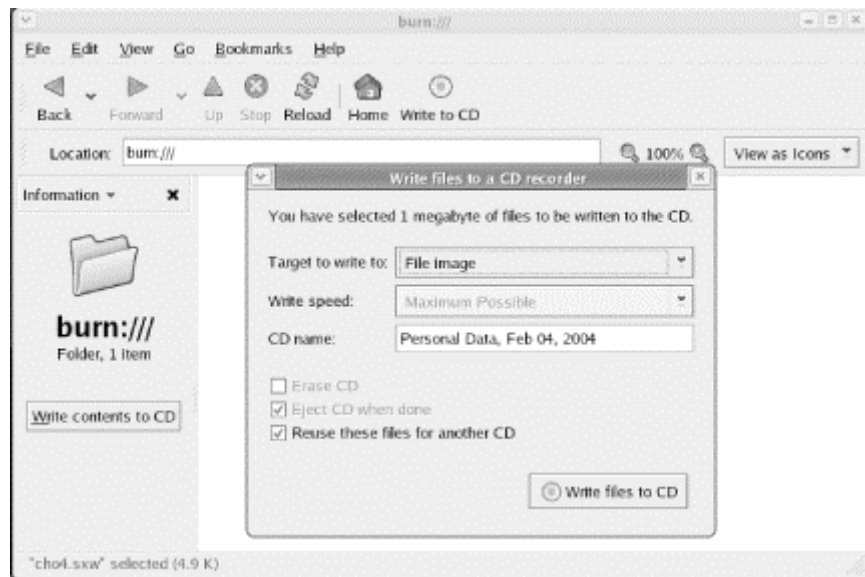
Figure 6-14. The burn window



To specify the files to be written to the CD, open a second Nautilus window. Then simply drag the desired files or directories into the CD Creator window.

When you've specified all the files to be written, click the Write to CD icon on CD Creator's toolbar. The Write Files to a CD Recorder dialog box appears ([Figure 6-15](#)).

Figure 6-15. The Write Files to a CD Recorder dialog box



Specify the target device, write speed, and other options. You can create an ISO image file—a disk file that contains the same data that would be found on a CD-ROM disk—by writing to the special File image target. However, in the more likely case that you want to burn a physical CD, you should choose your CD-R drive as the target device. Click the Write files to CD button to begin burning the CD.

Chapter 7. Conquering the bash Shell

Linux provides two user interfaces: the graphical user interface (GUI) hosted by X and an older, command-line interface (CLI) called the shell.

Those familiar with the MS-DOS command-line interface will recognize the shell, which you use by typing text commands to which the system responds by displaying text replies. But the comparison with the MS-DOS command line doesn't do justice to the Linux shell, which is vastly more powerful. And, *older* doesn't necessarily imply *inferior*.

GUIs are stylish primarily because they're easy to learn and use. But they're not always the most efficient way of operating a computer. A skilled user of the shell can often outrace a competitor using a GUI. Moreover, a GUI enables its user to perform only the functions provided by the GUI's programmers. In contrast, the shell is expandable. The shell enables users to define entirely new operations based on sequences of existing operations.

The real power of Linux lies in the shell. So, if you aspire to master Linux, you must conquer the shell. Even if your ambition falls short of gurudom, you'll find knowledge of the shell helpful. Many procedures from sources other than this book assume that you know how to use the shell. And, if X fails, you can't easily repair it without knowing how to use the shell.

Linux supports a variety of shells, but the most popular is the *bash* shell, described in this chapter. The chapter explains how to issue shell commands, and how to use shell commands to manipulate files and directories, work with removable media, and launch programs. The chapter also explains nano, a simple text editor that operates in text mode.

7.1 Issuing Shell Commands

The most common way to access the shell is via a terminal window, as explained in [Chapter 5](#) and [Chapter 6](#). However, a terminal window isn't the only way to access the shell. The [Section 4.3.3.2](#) in [Chapter 4](#) explains how to access the shell with a virtual console.

As your first Linux command, launch a terminal window, type **w**, and press **Enter**. Your contents of the terminal window should look something like this:

```
[bill@home bill]$ w
11:12am  up 6 min,  1 user,  load average: 0.00, 0.08, 0.05
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
bill      tty1                11:11am  0.00s   0.20s   0.11s   -bash
```

The **w** command tells Linux to display the system status and a list of all system users. In the example, the output of the command tells you that it's now 11:12 a.m., that the system has been up for six minutes, and that only one user—*bill*—is currently logged in. Notice that the command output is very terse, packing much information into a few lines. Such output is typical of Linux commands. At first, you may find Linux output cryptic and difficult to read, but over time you'll grow to appreciate the efficiency with which Linux communicates information.



Linux command output is not terse owing to an oversight or laziness on the part of the creators of Linux. Instead, Linux command output is designed so that it can be processed by programs as well as by humans. The structure of the output simplifies the task of programmers who write programs to process command output.

Linux provides many commands besides the **w** command—so many that you may despair of learning and recalling them. Actually, the number of commands you'll use regularly is fairly small. Soon, they will become second nature to you.

Try a second command, the **date** command:

```
[bill@home bill]$ date
Fri Oct 5 11:15:20 PST 2004
```

The **date** command displays the current date and time.

If you find working with MS-DOS distasteful or intimidating, you may not immediately enjoy working with the Linux command line. However, give yourself some time to adjust. The Linux command line has several features that make it easier to use, and more powerful, than MS-DOS.

7.1.1 Correcting Commands

Sometimes you may type a command incorrectly, causing Linux to display an error message. For example, suppose you typed **dat** instead of **date**:

```
[bill@home bill]$ dat
bash: dat: command not found
```

In such a case, carefully check the spelling of the command and try again. If you notice an error before pressing **Enter**, you can use the **Backspace** or **Left** arrow key to return to the point of the error and then type the correct characters. The **Backspace** key erases characters whereas the **Left** arrow key does not. You can also use the **Delete** key to delete unwanted characters.

Just as a web browser keeps track of recently visited sites, the *bash* shell keeps track of recently issued commands in what's known as the *history list*. You can scroll back through *bash*'s history by using the Up arrow key, or back down using the Down arrow key, just as you would with the Back and Forward buttons on a web browser. To reissue a command, scroll to it and press **Enter**. If you like, you can modify the command before reissuing it. When typing shell commands, you have access to a minieditor that resembles the DOSKEY editor of MS-DOS. This minieditor lets you revise command lines by typing key commands. [Table 7-1](#) summarizes some useful key commands interpreted by the shell. The key commands let you access a list of the 500 most recently executed commands. *bash*'s history is saved in the history file stored in the user's home directory, *~/.bash_history*.

Table 7-1. Useful editing keystrokes

Keystroke(s)	Function
Up arrow	Move back one command in the history list.

Table 7-1. Useful editing keystrokes

Keystroke(s)	Function
Down arrow	Move forward one command in the history list.
Left arrow	Move back one character.
Right arrow	Move forward one character.
Backspace	Delete previous character.
Tab	Attempt to complete the current word, interpreting it as a filename or command determined by context.
Alt-B	Move back one word.
Alt-D	Delete current word.
Alt-F	Move forward one word.
Ctrl-A	Move to beginning of line.
Ctrl-D	Delete current character.
Ctrl-E	Move to end of line.
Ctrl-K	Delete to end of line.
Ctrl-L	Clear the screen, placing the current line at the top of the screen.
Ctrl-U	Delete from beginning of line.
Ctrl-Y	Retrieve last item deleted.
Esc .	Insert last word of previous command (Esc is pressed before the dot, rather than at the same time).
Esc ? or Tab	List the possible completions (note that Esc is pressed before the question mark, not at the same time).

One of the most useful editing keystrokes, **Tab**, can also be used when typing a command. If you type the first part of a filename and press **Tab**, the shell will attempt to locate files with names matching the characters you've typed. If something exists, the shell fills out the partially typed name with the proper characters. You can then press **Enter** to execute the command or continue typing other options and arguments. This feature, called either *filename* or *command completion*, makes the shell much easier to use.

In addition to keystrokes for editing the command line, the shell interprets several keystrokes that control the operation of the currently executing program. [Table 7-2](#) summarizes these keystrokes. For example, typing **Ctrl-C** generally cancels execution of a program. This keystroke command is handy, for example, when a program is taking too long to execute and you'd prefer to try something else.

Table 7-2. Useful Control keystrokes

Keystroke	Function
Ctrl-C	Sends an interrupt signal to the currently executing command, which generally responds by terminating itself.
Ctrl-D	Sends an end-of-file to the currently executing command; use this keystroke to terminate console input.
Ctrl-Z	Suspends the currently executing program. The <code>fg</code> command resumes execution of the suspended program.

Several other special characters control the operation of the shell, as shown in [Table 7-3](#). The `#` and `;` characters are most often used in shell scripts, which you'll learn about in more detail later in this chapter. The `&` character causes the shell prompt to return immediately instead of waiting for a command to finish; the command runs in the background and you can continue to enter more commands. The pipe redirector (`|`) is explained in the section titled "Displaying Directory Contents."

Table 7-3. Other special shell characters

Character	Function
<code>#</code>	Marks the line as a comment, which the shell ignores.
<code>;</code>	Separates commands, letting you enter several commands on a single line.
<code>&</code>	Placed at the end of a command, causes the command to execute as a background process, so that a new shell prompt appears immediately after the command is entered.
<code>></code>	Stores output from the command in the file whose name follows.
<code><</code>	Takes input from the file whose name follows.
<code>\</code>	At end of line, continues command on the following line.
<code> </code>	Sends output from the command to another command.

7.2 Working with the Linux Command Prompt

Linux commands share a simple, common structure. This section describes their common structure and explains how you can obtain helpful information about the commands available to you.

7.2.1 Commands and Arguments

The general form of a shell command line is this:

```
command [options] [arguments]
```

The **command** determines what operation the shell will perform and the *options* and *arguments* customize, or fine-tune, the operation. The *options* and *arguments* may or may not appear, as indicated by the square brackets. Sometimes the **command** specifies a program file that will be launched and run; such a command is called an *external command*. Linux generally stores these files in `/bin`, `/usr/bin`, or `/usr/local/bin`. System administration commands are generally stored in `/sbin` or `/usr/sbin`, which are included by default in the path of the root user. When a command specifies a program file, the shell passes any specified arguments to the program, which scans and interprets them, adjusting its operation accordingly.

Some commands are not external program files; instead they are built-in commands interpreted by the shell itself. One important way in which shells differ is in the built-in commands that they support. Later in this section, you'll learn about some of *bash*'s built-in commands.

The name of a Linux command almost always consists of lowercase letters and digits. Most commands let you specify options or arguments. However, in any given case, you may not need to do so. For example, typing the **w** command without options and arguments causes Linux to display a list of current users.



Remember, Linux commands are case sensitive; be sure to type each character of a command in the proper case.

Options modify the way that a command works. Many options consist of a single letter, prefixed by a dash. Often, you can specify more than one option; when you do so, you separate each option with one or more spaces. For example, the **-h** option of the **w** command causes the output of the command to omit the header lines that give the time and the names of the fields:

```
[bill@home bill]$ w -h
```

Arguments specify filenames or other targets that direct the action of the command. For example, the **w** command lets you specify a username as an argument, which causes the command to list logins that pertain only to the specified user:

```
[bill@home bill]$ w bill
```

Some commands let you specify a series of arguments; you must separate each argument with a space between them. For example, the following command prints a list of logins by the root user, without header lines:

```
[bill@home bill]$ w -h bill
```

When a command includes several arguments, a command may not fit on a single line. However, you can continue typing when you reach the end of a line, because the shell will automatically wrap your input to the next line. If you find line wrapping disconcerting, you can type a backslash (`\`) at the end of a line, press **Enter**, and continue typing on the next line. The backslash is the shell's line continuation character; the shell sees lines joined by a backslash as though they were a single line. Don't type anything after the backslash or the continuation feature won't work correctly.

7.2.2 Getting Help

Because Linux provides so many commands and because Linux commands provide so many possible options, you can't expect to recall all of them. To help you, Linux provides the **man** and **apropos** commands, which let you access a help database that describes commands and their options.

7.2.2.1 Using man

Each Linux command is described by a special file called a *manual page*. The manual pages (or *manpages*) are stored in a group of subdirectories comprising a help database. To access this database, you use the **man** command, which resembles the MS-DOS **help** command. For example, to get help on using the **w** command, type:

```
[bill@home bill]$ man w
```

[Figure 7-1](#) shows the resulting output, which the command displays one page at a time. Notice the colon prompt that appears at the bottom left of the screen. To page forward, press the **Space** key; to page backward, press the **b** key. To exit the **man** program, press the **q** key.

Manpages are organized according to a common format. At the beginning of a manpage, you'll find the name of the page and the section of the database from which the page comes, shown in parentheses. For example, [Figure 7-1](#) shows "W(1)" in the upper-left and -right corners. This means that you're looking in section 1 of the manpage (the section pertaining to commands) for the **w** command.

Figure 7-1. A typical manpage

```

W(1)                                Linux User's Manual                                W(1)

NAME
    w - Show who is logged on and what they are doing.

SYNOPSIS
    w - [hufV] [user]

DESCRIPTION
    w displays information about the users currently on the machine, and their
    processes. The header shows, in this order, the current time, how long
    the system has been running, how many users are currently logged on, and
    the system load averages for the past 1, 5, and 15 minutes.

    The following entries are displayed for each user: login name, the tty
    name, the remote host, login time, idle time, JCPU, PCPU, and the command
    line of their current process.

    The JCPU time is the time used by all processes attached to the tty. It
    does not include past background jobs, but does include currently running
    background jobs.

    The PCPU time is the time used by the current process, named in the "what"
    field.

COMMAND-LINE OPTIONS
    -h    Don't print the header.

    -u    Ignores the username while figuring out the current process and cpu
    times. To demonstrate this, do a "su" and do a "w" and a "w -u".

    -s    Use the short format. Don't print the login time, JCPU or PCPU
    times.

    -f    Toggle printing the from (remote hostname) field. The default as
    released is for the from field to not be printed, although your sys-
    tem administrator or distribution maintainer may have compiled a ver-
    sion in which the from field is shown by default.

    -V    Display version information.

    user  Show information about the specified user only.

:

```

Next in the output comes the name and a brief description of the command. Then comes a synopsis of the command, which shows the options and arguments that you can specify. Brackets enclose parts of a command that you can choose to include or omit. Next comes a detailed description of the operation of the command, followed by a description of its options.

As you're learning your way around Linux, you may find it convenient to reserve a terminal window or virtual console for running the **man** command. Alternatively, you can browse manpages using Konqueror. Browsing the URI *man://index* will present the Unix manual index, from which you can jump to other manpages. That way, you can enter commands in a separate virtual console, switching between windows or consoles to refresh your recollection of the options and arguments of commands as you type them.

[Table 7-4](#) describes the sections of the manual page database; most sections are primarily of interest to programmers. As a user and system administrator, you'll be interested primarily in sections 1 and 8.

Table 7-4. Manual page sections

Section	Description
1	Executable programs and shell commands

Table 7-4. Manual page sections	
Section	Description
2	System calls (provided by the kernel)
3	Library calls (provided by system libraries)
4	Special files (for example, device files)
5	File formats and conventions
6	Games
7	Macro packages and conventions
8	System administration commands
9	Nonstandard kernel routines

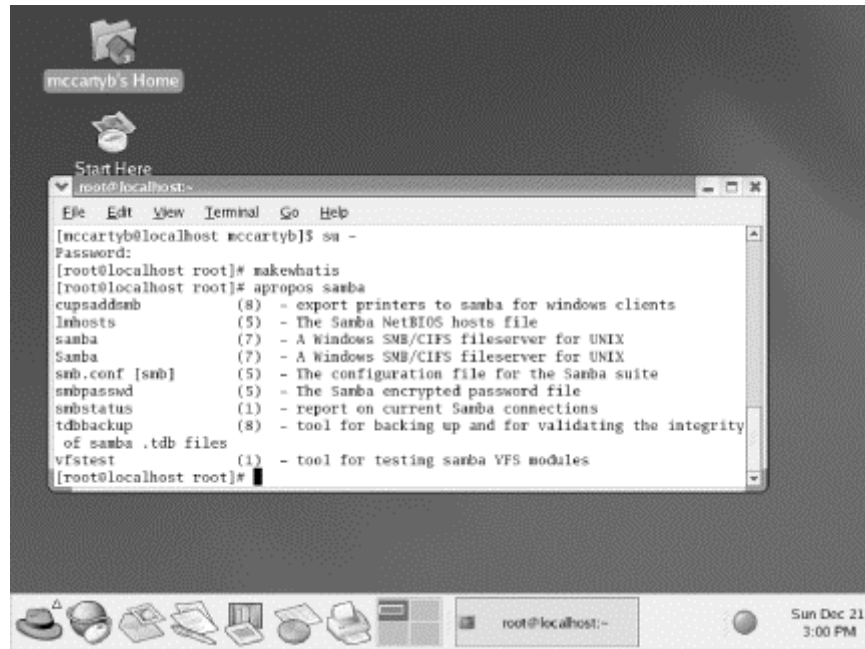
7.2.2.2 Using apropos

The **man** command searches the manual pages and displays detailed information about a specified command. The **apropos** command also searches the manual pages, but it displays just a one-line summary of each. You have to supply **apropos** with a keyword that indicates the information you're looking for; it simply displays each summary line that contains the keyword. For example, typing the command:

```
[bill@home bill]$ apropos samba
```

will display a list of manpages containing the word *samba*, as shown in [Figure 7-2](#).

Figure 7-2. Output of the apropos command



The **apropos** command is useful when you don't recall the name of a Linux command. By typing a related keyword, you can obtain a list of commands and search the list for the command you need.



The **apropos** command uses a special database to store information about commands. Before using the command for the first time, you must create the database by logging in as root and issuing the command:

```
makewhatis
```

The command may require several minutes to complete. After installing a package that includes commands, you can update the database to include information about the new commands by issuing the same command.

7.2.3 Using Commands That Work with Directories

Now that you understand the fundamentals of issuing Linux commands, you're ready to learn some commands that work with directories. Rather than simply reading this section, you should log in to your Linux system and try the commands for yourself. By doing so, you will begin to develop skill in working with shell commands.

7.2.3.1 Displaying the working directory

To display the current working directory, issue the **pwd** (print working directory) command. The **pwd** command requires no options or arguments:

```
[bill@home bill]$ pwd
/root
```

The **pwd** command displays the absolute pathname of the current working directory.

7.2.3.2 Changing the working directory

To change the working directory, issue the **cd** (change directory) command, specifying the pathname of the new working directory as an argument. You can use an absolute or relative pathname. For example, to change the working directory to the */bin* directory, type:

```
[bill@home bill]$ cd /bin
[bill@home /bin]$
```

Notice how the prompt changes to indicate that */bin* is now the working directory. You can quickly return to your home directory by issuing the **cd** command without an argument:

```
[bill@home /bin]$ cd
[bill@home bill]$
```

Again, notice how the prompt changes to indicate the new working directory. If you attempt to change the working directory to a directory that doesn't exist, Linux displays an error message:

```
[bill@home bill]$ cd nowhere
bash: nowhere: No such file or directory
```

7.2.3.3 Displaying directory contents

To display the contents of a directory, you use the **ls** (list) command. The **ls** command provides many useful options that let you tailor its operation and output to your liking. The simplest form of the **ls** command takes no options or arguments. It simply lists the contents of the working directory, including files and subdirectories (your own output will differ, reflecting the files present in your working directory):

```
[bill@home bill]$ ls
GNUstep          firewall          sniff
Xrootenv.0       linux            ssh-1.2.26
audio.cddb       mail            ssh-1.2.26.tar.gz
audio.wav        mirror          support
axhome           mirror-2.8.tar.gz temp
conf            nlxb3181.tar    test
corel            penn            test.doc
drivec.img       scan            tulip.c
dynip_2.00.tar.gz screen-3.7.6-0.i386.rpm win98
```

Here, the output is presented in lexical (dictionary) order, as three columns of data. Notice that filenames beginning with uppercase letters appear before those beginning with lowercase letters.

A more sophisticated form of the **ls** command that includes the **-l** option displays descriptive information along with the filenames, as shown in [Figure 7-3](#).

Figure 7-3. Output of the ls command

```
[root@desktop /root] # ls -l
total 558414
d rwxr-xr-x  5 root  root    1024 Dec 23 13:48 GNUstep
- rw-r--r--  1 root  root     331 Feb 11 10:19 Xrootenv.0
- rw-rw-r--  1 root  root     490 Jan  6 15:07 audio.cddb
- rw-r--r--  1 root  root  45254876 Jan  6 15:08 audio.wav
d rwxr-xr-x  2 root  root    1024 Feb 20 16:41 axhome
- rw-r--r--  1 root  root     900 Jan 18 20:15 conf
d rwxr-xr-x  2 root  root    1024 Dec 25 10:03 corel
- rw-r--r--  1 root  root     915 Jan 18 20:57 firewall
d rwxrwxr-x  2 root  root    1024 Jan  6 15:42 linux
d rwx-----  2 root  root    1024 Jan  4 02:19 mail
d rwxr-xr-x  3 root  root    1024 Jan  4 01:49 mirror
- rwxr--r--  1 root  root      29 Dec 27 15:07 openn
d rwxr-xr-x  3 root  root    1024 Dec 26 13:24 scan
d rwxrwxr-x  3 root  root    1024 Jan  4 02:34 sniff
```

Type	Access modes	# of links	Owner	Group	Size (bytes)	Modification date and time	Name
d	rwxr-xr-x	5	root	root	1024	Dec 23 13:48	GNUstep
-	rw-r--r--	1	root	root	331	Feb 11 10:19	Xrootenv.0
-	rw-rw-r--	1	root	root	490	Jan 6 15:07	audio.cddb
-	rw-r--r--	1	root	root	45254876	Jan 6 15:08	audio.wav
d	rwxr-xr-x	2	root	root	1024	Feb 20 16:41	axhome
-	rw-r--r--	1	root	root	900	Jan 18 20:15	conf
d	rwxr-xr-x	2	root	root	1024	Dec 25 10:03	corel
-	rw-r--r--	1	root	root	915	Jan 18 20:57	firewall
d	rwxrwxr-x	2	root	root	1024	Jan 6 15:42	linux
d	rwx-----	2	root	root	1024	Jan 4 02:19	mail
d	rwxr-xr-x	3	root	root	1024	Jan 4 01:49	mirror
-	rwxr--r--	1	root	root	29	Dec 27 15:07	openn
d	rwxr-xr-x	3	root	root	1024	Dec 26 13:24	scan
d	rwxrwxr-x	3	root	root	1024	Jan 4 02:34	sniff

The first line of the output shows the amount of disk space used by the working directory and its subdirectories, measured in 1 KB blocks. Each remaining line describes a single file or directory. The columns are:

Type

The type of file: a directory (d), or an ordinary file (-). If your system supports color, Linux displays output lines that pertain to directories in blue and lines that pertain to files in white.

Access modes

The access mode, which determines which users can access the file or directory. You'll learn more about access modes, links, and groups in subsequent sections of this chapter.

Links

The number of files or directories linked to this one.

Owner

The user who owns the file or directory.

Group

The group that owns the file or directory.

Size

The size of the file or directory, in bytes.

Modification date

The date and time when the file or directory was last modified.

Name

The name of the file or directory.

If a directory contains many files, the listing will fill more than one screen. To view the output one screen at a time, use the following command:

```
[bill@home bill]$ ls | less
```

This command employs the pipe redirector, sending output of the **ls** subcommand to the **less** subcommand, which presents the output one screen at a time. You can control the operation of the **less** command by using the following keys:

- **Space** moves you one page forward.
- **b** moves you one page back.
- **q** or **Q** exits the program and returns you to the command prompt.

If you want to list a directory other than the working directory, you can type the name of the directory as an argument of the **ls** command, like so:

```
ls /bin
```

Linux displays the contents of the directory but does not change the working directory. Similarly, you can display information about a file by typing its name as an argument of the **ls** command. Moreover, the **ls** command accepts an indefinite number of arguments, so you can type a series of directories and filenames as arguments, separating each with one or more spaces.

When the name of a directory or file begins with a dot (**.**), the output of the **ls** command does not normally include the directory or file, because the file is *hidden*. To cause the output of the **ls** command to include hidden directories and files, use the **-a** option. For example, to list all the files and subdirectories in the current directory—including hidden ones—type:

```
[bill@home bill]$ ls -a -l
```

If you prefer, you can combine the **-a** and **-l** options, typing the command like this:

```
[bill@home bill]$ ls -al
```

A user's home directory generally includes several hidden files containing configuration information for various programs. For example, the *.profile* file contains configuration information for the Linux *bash* shell.

The **ls** command provides a host of additional useful options; see its manual page for details.

7.2.3.4 Creating a directory

You can create directories by using the **mkdir** (make directory) command. Just type the name of the new directory as an argument of the command. Linux creates the directory as a subdirectory of the working directory. For example, this command creates a subdirectory named *office*:

```
[bill@home bill]$ mkdir office
```

If you don't want to create the new directory as a subdirectory of the working directory, type an absolute or relative pathname as the argument. For example, to create a directory named */tmp/documents*, type:

```
[bill@home bill]# mkdir /tmp/documents
```

The name of a directory or file must follow certain rules. For example, it must not contain a slash (/) character. Directory names and filenames usually include letters (either upper- or lowercase), digits, dots, and underscores (_). You can use other characters, such as spaces and hyphens, but such names present problems, because the shell gives them special meaning. If you simply must use a name containing special characters, enclose the name within single quotes (''). The quotes don't become part of the name that is stored on the disk. This technique is useful when accessing files on a Windows filesystem; otherwise, you'll have trouble working with files in directories such as *My Documents*, which have pathnames containing spaces.

Most MS-DOS filenames contain a dot, but most Linux filenames do not. In MS-DOS, the dot separates the main part of the filename from a part known as the extension, which denotes the type of the file. For example, the MS-DOS file *memo.txt* would contain text. Most Linux programs ignore file extensions, so Linux filenames don't require an extension. However, if you plan to send a file to someone using an operating system other than Linux, you should include an appropriate file extension, such as *.txt* for a text file.

7.2.3.5 Removing a directory

To remove a directory, use the **rmdir** command. For example, to remove *unwanted*, a subdirectory of the working directory, type:

```
[bill@home bill]$ rmdir unwanted
```

If the directory you want to delete is not a subdirectory of the working directory, remove it by typing an absolute or relative pathname.

You cannot use **rmdir** to remove a directory that contains files or subdirectories; you must first delete the files in the directory and then remove the directory itself. Hidden files can present a puzzle, because a directory containing hidden files may look empty even though it's not. You must delete hidden files before you can delete the directory containing them.

7.2.4 Working with Files

Directories contain files and other directories. You use files to store data. This section introduces you to several useful commands for working with files.

7.2.4.1 Displaying the contents of a file

Linux files, like most Windows files, can contain text or binary information. The contents of a binary file are meaningful only to skilled programmers, but you can easily view the contents of a text file. Simply type the **cat** command, specifying the name of the text file as an argument. For example:

```
[root@desktop /root]# cat /etc/passwd
```

displays the contents of the `/etc/passwd` file, which lists the accounts on the system. If a file is too large to be displayed on a single screen, the first part of the file will whiz past you and you'll see only the last few screenfuls of lines of the file. To avoid this, you can use the **less** command:

```
[root@desktop /root]# less /etc/passwd
```

This command displays the contents of a file in the same way the **man** command displays a manual page. You can use **Space** and the **b** key to page forward and backward through the file, and the **q** or **Q** key to exit the command.

7.2.4.2 Removing a file

To delete a file, type the **rm** (remove) command, specifying the name of the file as an argument. For example:

```
[bill@home bill]$ rm badfile
```

removes the file named *badfile* contained in the working directory. If a file is located elsewhere, you can remove it by specifying an absolute or relative pathname.



Once you remove a Linux file, its contents are likely lost forever. Be careful to avoid removing a file that contains needed information. Better yet, be sure to have a backup copy of any important data.

The **-i** option causes the **rm** command to prompt you to verify your decision to remove a file. If you don't trust your typing skills, you may find this option helpful. If you log in as the root user, Linux automatically supplies the **-i** option even if you don't type it.

7.2.4.3 Copying a file

To copy a file, use the **cp** command, specifying the name (or path) of the file you want to copy and the name (or path) to which you want to copy it. For example:

```
[root@desktop /root]# cp /etc/passwd sample
```

copies the `/etc/passwd` file to a file named *sample* in the working directory.

If the destination file already exists, Linux overwrites it. You must therefore be careful to avoid overwriting a file that contains needed data. Before copying a file, use the **ls** command to ensure that no file will be overwritten; alternatively, use the **-i** option of the **cp** command, which prompts you to verify that you want to overwrite an existing file. If you log in as the root user, Linux automatically supplies the **-i** option even if you don't type it.

7.2.4.4 Renaming or moving a file

To rename a file, use the **mv** command, specifying the name (or path) of the file and the new name (or path). For example:

```
[bill@home bill]$ mv old new
```

renames the file named *old* as *new*. If the destination file already exists, Linux overwrites it, so you must be careful. Before moving a file, use the **ls** command to ensure that no file will be overwritten or use the **-i** option of the **mv** command, which prompts you to verify that you want to overwrite an existing file. If you log in as the root user, Linux automatically supplies the **-i** option even if you don't type it.

The **mv** command can rename a directory but sometimes cannot move a directory from one device to another. If you encounter problems when moving a directory to a new device, first copy the directory and its contents and then remove the original.

7.2.4.5 Finding a file

If you know the name of a file but do not know which directory contains it, you can use the **find** command to locate the file. For example:

```
[bill@home bill]$ find . -name 'missing' -print
```

attempts to find a file named *missing*, located in (or beneath) the current working directory (.). If the command finds the file, it displays its absolute pathname.

If you know only part of the filename, you can surround the part you know with asterisks (*):

```
[bill@home bill]$ find / -name '*iss*' -print
```

This command will find any file whose name includes the characters *iss*, searching every subdirectory of the root directory (that is, the entire system).

Another command useful for finding files is **locate**. The **locate** command uses a database that is updated only daily. So it can't find recently created files and it shows files that may have been recently deleted. But it operates much more quickly than the **find** command. To use the **locate** command, specify as the command's argument a string of characters, which need not be enclosed in quotes. The command will list all filenames in its database that contain the specified characters. For example, the command:

```
locate pass
```

lists all files containing the characters *pass*.



The **locate** command depends on a database built by the **updatedb** command. You must login as root and run **updatedb** before running **locate** for the first time and as often thereafter as you feel that your system's file structure has changed significantly. The *cron* and *anacron* services will generally take care of this for you. However, you can run the **updatedb** command manually if the results of the **locate** command seem out of date.

7.2.4.6 Printing a file

If your system includes a configured printer, you can print a file by using the **lpr** command. For example:

```
[root@desktop /root]# lpr /etc/passwd
```

/etc/passwd to the printer. See [Chapter 9](#) for information on configuring a printer. You can send other files to the printer while a file is printing. The **lpq** command lets you see what files are queued to be printed:

```
[root@desktop /root]# lpq
lp is ready and printing
Rank   Owner      Job   Files              Total Size
active root          155  /etc/passwd        1030 bytes
```

Each waiting or active file has an assigned print job number. You can use **lprm** to cancel printing of a file, by specifying the print job number. For example:

```
[root@desktop /root]# lprm 155
```

cancels printing of job number 155. However, only the user who requested that a file be printed (or the root user) can cancel printing of the file.

7.2.4.7 Working with compressed files

To save disk space and expedite downloads, you can compress a data file. By convention, compressed files are named ending in *.gz*; however, Linux doesn't require or enforce this convention.

To expand a compressed file, use the **gunzip** command. For example, suppose the file *bigfile.gz* has been compressed. Typing the command:

```
[bill@home bill]$ gunzip bigfile.gz
```

extracts the file *bigfile* and removes the file *bigfile.gz*.

To compress a file, use the **gzip** command. For example, to compress the file *bigfile*, type the command:

```
[bill@home bill]$ gzip bigfile
```

The command creates the file *bigfile.gz* and removes the file *bigfile*.

A different compression format is provided by the **bzip2** command; files compressed this way usually have *.bz2* extensions and can be uncompressed with **bunzip2**.

Sometimes it's convenient to store several files (or the contents of several subdirectories) in a single file. This is useful, for example, in creating a backup or archive copy of files. The Linux **tar** command creates a single file that contains data from several files. Unlike the **gzip** command, the **tar** command doesn't disturb the original files. To create a tarfile, as a file created by the **tar** command is called, issue a command like this:

```
tar -cvf tarfile files-or-directories
```

Substitute *tarfile* with the name of the tarfile you want to create and *files-or-directories* with a list of files and directories, separating the list elements by one or more spaces. You can use absolute or relative pathnames to specify the files or directories. By convention, the name of a tarfile ends with *.tar*, but Linux does not require or enforce this convention.

Some people refer to tarfiles as *tarballs*, because they often contain multiple files.

For example, to create a tarfile named *backup.tar* that contains all the files in all subdirectories of the directory */home/bill*, type:

```
[bill@home bill]$ tar -cvf backup.tar /home/bill
```

The command creates the file *backup.tar* in the current working directory. You can list the contents of a tarfile by using a command that follows this pattern:

```
tar -tvf tarfile | less
```

The `| less` causes the output to be sent to the `less` command, so that you can page through multiple pages. If the tarfile holds only a few files, you can omit `| less`.

To extract the contents of a tarfile, use a command that follows this pattern:

```
tar -xvf tarfile
```

This command expands the files and directories contained within the tarfile as files and subdirectories of the working directory.



If a file or subdirectory already exists, the **tar** command silently overwrites it.

The **tar** command provides a host of useful options; see its manpage for details.

It's common to compress a tarfile, which you can do by specifying the options **-czvf** instead of **-cvf**. Compressed tarfiles are conventionally named ending with *.tgz*. To expand a compressed tarfile, specify the options **-xzvf** instead of **-xvf**.

The **tar** command doesn't use the ZIP method of compression common in the Windows world. However, Linux can easily work with, or even create, ZIP files.

To create a ZIP file that holds compressed files or directories, issue a command like this one:

```
zip -r zipfile files_to_zip
```

where *zipfile* names the ZIP file that will be created and *files_to_zip* specifies the files and directories to be included in the ZIP file.

To expand an existing ZIP file, issue a command like this one:

```
unzip zipfile
```

7.2.4.8 Working with links

Windows supports shortcuts, which let you refer to a file or directory (folder) by several names. Shortcuts also let you include a file in several directories or a subdirectory within multiple parent directories. In Linux, you accomplish these results by using the **ln** command, which links multiple names to a single file or directory. These names are called *symbolic links*, *soft links*, *symlinks*, or simply *links*.

To link a new name to an existing file or directory, type a command that follows this pattern:

```
ln -s old new
```

For example, suppose that the current working directory contains the file *william*. To be able to refer to this same file by the alternative name *bill*, type the command:

```
[bill@home bill]$ ln -s william bill
```

The **ls** command shows the result:


```
[bill@home bill]$ ls -l
lrwxrwxrwx  1 root    root      7 Feb 27 13:58 bill->william
-rw-r--r--  1 root    root    1030 Feb 27 13:26 william
```

The new file (*bill*) has type *l*, which indicates it's a link rather than a file or directory. Moreover, the **ls** command helpfully shows the name of the file to which the link refers (*william*). Notice the file size of the link *bill*. Creating a link merely creates a pointer to a file rather than a duplicate of the file, thereby saving disk space.

If you omit the **-s** option, Linux creates what's called a *hard link*. A hard link must be stored on the same filesystem as the file to which it refers, a restriction that does not apply to symbolic links. The link count displayed by the **ls** command reflects only hard links; symbolic links are ignored. Hard links are seldom used, because soft links are more flexible.

7.2.4.9 Working with file permissions

As explained in [Chapter 4](#), access permissions determine which operations a user can perform on a directory or file. [Table 7-5](#) lists the possible permissions and explains the meaning of each. Recall from [Chapter 4](#) that permissions work differently for directories than for files. For example, permission *r* denotes the ability to list the contents of a directory or *read* the contents of a file. A directory or file can have more than one permission. Only the listed permissions are granted; any other operations are prohibited. For example, a user who had file permission *rw* could *read* or *write* the *file* but could not execute it, as indicated by the absence of the execute permission, *x*. Look back to [Figure 7-3](#) to see how the **ls** command displays permissions.

Table 7-5. Access permissions		
Permission	Meaning for a directory	Meaning for a file
<i>r</i>	List the directory	Read contents
<i>w</i>	Create or remove files	Write contents
<i>x</i>	Access files and subdirectories	Execute

The access modes of a directory or file consist of three permissions:

User/Owner

Applies to the owner of the file

Group

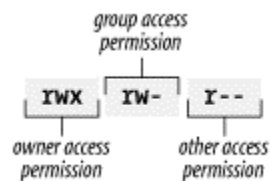
Applies to users who are members of the group assigned to the file

Other

Applies to other users

The **ls** command lists the file access modes in the second column of its long output format, as shown in [Figure 7-3](#). The column contains nine characters: the first three specify the access allowed the owner of the directory or file, the second three specify the access allowed users in the same group as the directory or file, and the final three specify the access allowed to other users (see [Figure 7-4](#)).

Figure 7-4. Access modes specify three permissions



You set the access modes of a directory or file by using the **chmod** command, which has the following pattern:

chmod *nnn* *directory-or-file*

The argument *nnn* is a three-digit number, which gives the access mode for the owner, group, and other users. [Table 7-6](#) shows each possible digit and the equivalent access permission. For example, the argument **751** is equivalent to **rw-r-x--x**, which gives the owner every possible permission, gives the group read and execute permission, and gives other users execute permission.

Table 7-6. Numerical access mode values	
Value	Meaning
0	---
1	--x
2	-w-
3	-wx
4	r--
5	r-x
6	rw-
7	rwx

If you're the owner of a file or directory (or if you're the root user), you can change the ownership of the file or directory by using the **chown** command. For example, the following command assigns *newuser* as the owner of the file *hotpotato*:

```
[bill@home bill]$ chown newuser hotpotato
```

The owner of a file or directory (and the root user) can also change the group of a file. For example, the following command assigns *newgroup* as the new group of the file *hotpotato*:

```
[bill@home bill]$ chgrp newgroup hotpotato
```

The group you assign to a file or directory must have been previously established by the root user. And, unless the command is issued by the root user, the user must be a member of the new group to which the file is assigned. The valid groups appear in the file */etc/group*, which only the root user can alter. The root user can assign each user to one or more groups. When you log on to the system, you are assigned to one of these groups—your *login group*—by default. To change to another of your assigned groups, you can use the **newgrp** command. For example, to change to the group named *secondgroup*, use the following command:

```
[root@desktop /root]# newgrp secondgroup
```

If you attempt to change to a group that does not exist or to which you have not been assigned, your command will fail. When you create a file or directory, it is automatically assigned your current group as its owning group.

7.2.4.10 Running programs

In Linux, as in MS-DOS and Windows, programs are stored in files. Often, you can launch a program by simply typing its filename. However, this assumes that the file is stored in one of a series of directories known as the *path*. A directory included in this series is said to be *in the path* or *on the path*. If you've worked with MS-DOS, you're familiar with the MS-DOS path, and the Linux path works much like it.

If the file you want to launch is not stored in a directory on the path, you can simply type the pathname of the file. Linux will then launch the program even though it's not on the path. If the file you want to launch is stored in the current working directory, type *./* followed by the name of the program file. Again, Linux will launch the program even though it's not on the path.

For example, suppose the program **bigdeal** is stored in the directory */home/bob*, which is the current directory and which happens to be on the path. You could launch the program with any of these commands:

```
bigdeal
./bigdeal
/home/bob/bigdeal
```

The first command assumes that the program is on the path. The second assumes that the program resides in the current working directory. The third explicitly specifies the location of the file.

7.2.5 Mounting and Unmounting Drives

To mount a device or partition, you use the **mount** command, which has the following pattern:

```
mount options device directory
```

The **mount** command provides many options. However, you can generally use the **mount** command without any options; consult **mount**'s manpage to learn about the available options.



The reason you can often use the **mount** command without options is that the file */etc/fstab* describes your system's devices and the type of filesystem each is likely to contain. When you reboot your system after adding a new device, the *kudzu* service detects the new device and updates */etc/fstab* to include an entry for the device, if necessary. However, you can edit the */etc/fstab* by hand to add new entries or revise entries placed there by *kudzu*.

You must specify the device that you want to mount and a directory, known as the *mount point*. To make it convenient to access various devices, Linux treats a mounted device as a directory; mounting the device associates it with the named directory. For example, the following command is used to mount a CD-ROM:

```
[root@desktop /root]# mount -t iso9660 /dev/cdrom /mnt/cdrom -o ro
```

The file */dev/cdrom* is a link that points to the actual device file associated with your system's CD-ROM drive. The directory */mnt/cdrom* is a directory created by the install program; this directory is conventionally used as the mounting point for CD-ROMs. The type of filesystem found on most CD-ROMs is **iso9660**, the value of the **-t** argument. The **-o** argument, **ro**, specifies that the filesystem is read-only; that is, it can be read but not written. The file */etc/fstab* can supply most of these arguments if they're omitted. Generally, you can mount a CD-ROM by issuing the abbreviated command:

```
[root@desktop /root]# mount /dev/cdrom
```

After the command has completed, you can access files and directories on the CD-ROM just as you would access ordinary files and directories on the path */mnt/cdrom*. For example, to list the top-level files and directories of the CD-ROM, simply type:

```
[root@desktop /root]# ls /mnt/cdrom
```

To mount an MS-DOS floppy disk in your *a:* drive, type:

```
[root@desktop /root]# mount -t msdos /dev/fd0 /mnt/floppy
```

To unmount a device, specify its mount point as an argument of the **umount** command (note the missing *n* in *umount*). For example, to unmount a CD-ROM diskette, type:

```
[root@desktop /root]# umount /mnt/cdrom
```

Generally, only the root user can unmount a device. However, Red Hat Linux allows ordinary users to mount and unmount devices when logged in locally. Nevertheless, a device can be unmounted only if it's not in use. For example, if a user's working directory is a directory of the device, the device cannot be unmounted.



If you can't unmount a device, check each terminal window and virtual console to see if one of them has a session that's using the device as its working directory. If so, either exit the session or change to a working directory that isn't associated with the device.

7.2.6 Formatting a Floppy Disk

Before you can write data on a floppy disk, you must format it. The Linux command to format a floppy disk is **fdformat**. Simply follow the command with an argument that specifies the floppy drive and the capacity of the floppy disk; the available arguments are listed in [Table 7-7](#).

Table 7-7. Floppy drive designators

Designation	Meaning
/dev/fd0	3.5-inch disk in <i>a</i> : (1.44 MB)
/dev/fd0H1440	3.5-inch disk in <i>a</i> : (1.44 MB)
/dev/fd1	3.5-inch disk in <i>b</i> : (1.44 MB)
/dev/fd1H1440	3.5-inch disk in <i>b</i> : (1.44 MB)
/dev/fd1H2880	3.5-inch disk in <i>b</i> : (2.88 MB)

For example, to format a 1.44 MB floppy disk, log in as *root* and issue the command:

```
[root@desktop /root]# fdformat /dev/fd0H1440
```

The **fdformat** command performs only a low-level format. Before the floppy disk can be used, you must place a filesystem on it. Floppy disks containing an MS-DOS filesystem are useful for transferring data between Windows and Linux. To place an MS-DOS filesystem on a formatted floppy disk, issue the command:

```
[root@desktop /root]# mkdosfs /dev/fd0
```

Once the floppy disk has been formatted and given a filesystem, you can mount it and then read and write it.



Be sure you unmount the floppy disk before you remove it. Unmounting the floppy disk ensures that all pending data has been written to it; otherwise, the floppy disk may be unusable due to corrupt data.

7.3 Useful Linux Programs

This section presents several programs you may find helpful in working with your Linux system. You'll learn several commands that report system status and you'll learn how to use *nano*, a simple text editor.

7.3.1 Viewing System Information

Linux provides a number of commands that report system status. The most commonly used commands are shown in [Table 7-8](#). These commands can help you troubleshoot system problems and identify resource bottlenecks. Although each command can be used without options or arguments, each supports options and arguments that let you customize operation and output; consult the appropriate manpage for details.

Table 7-8. Useful system commands	
Command	Function
df	Shows the amount of free disk space (in 1 KB blocks) on each mounted filesystem.
du	Shows the amount of disk space (in 1 KB blocks) used by the working directory and its subdirectories. With the -s option, displays just a summary without listing all the subdirectories and files.
free	Shows memory usage statistics, including total free memory, memory used, physical memory, swap memory, shared memory, and buffers used by the kernel.
ps	Shows the active processes (instances of running programs) associated with this login session. Use the -a option to list all processes.
top	Shows a continually updated display of active processes, and the resources they are using. Type the q key to exit.
uptime	Shows the current time, the amount of time logged in, the number of users logged in, and system load averages.
users	Shows each login session.
w	Shows a summary of system usage, currently logged-in users, and active processes.
who	Shows the names of users currently logged in, the terminal each is using, the time each has been logged in, and the name of the host from which each logged in (if any).

7.3.2 Using the nano Editor

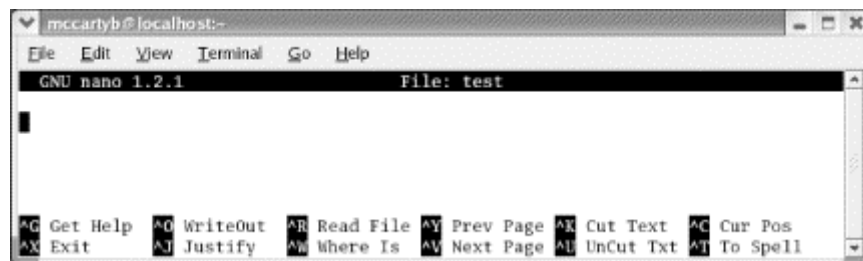
If you're working under X, you have access to a variety of GUI text editors. However, GUI text editors cannot be used from a virtual console. The *nano* editor is a simple text editor that you can think of as the Linux equivalent of the Windows program named Edit, because it can be used in graphical or text mode.

To start *nano*, simply type **nano** at the shell prompt, or if you want to create or edit a particular file, type **nano** followed by the name of the file (or the file's path, if the file is not in the working directory). For example, to create or edit the file *mydata*, type:

```
[bill@home bill]$ nano mydata
```

[Figure 7-5](#) shows *nano*'s standard display. At the top of the display is a status line, which shows the version of the program and the name of the file being edited (or New Buffer, if the file is new). If the file has been modified, the upper-right corner of the display contains the word *Modified*. The bottom two lines of the display list the available editing commands. Most of the commands require you to type a control character, so that commands can be distinguished from characters you want to add to the buffer, as *nano*'s work area is termed. Typing an ordinary character inserts it at the current cursor position. You can use the cursor keys to move around the display; you can use the **Delete** or **Backspace** key to erase unwanted characters. Some commands use the third line from the bottom to report status and obtain additional input.

Figure 7-5. The nano editor



[Table 7-9](#) summarizes *nano*'s commands. Notice that the command **Ctrl-G** accesses *nano*'s help system. You can access several of the commands by using function keys; for example, pressing **F1** has the same result as typing **Ctrl-G**.

Table 7-9. Summary of nano commands

Command	Description
Ctrl-^	Mark the cursor position as beginning of selected text.
Ctrl-A	Move to the beginning of the current line.
Ctrl-B	Move backward one character.
Ctrl-C (F11)	Report the current cursor position.
Ctrl-D	Delete the character at the cursor position.
Ctrl-E	Move to the end of the current line.
Ctrl-F	Move forward one character.

Table 7-9. Summary of nano commands	
Command	Description
Ctrl-G (F1)	Display help.
Ctrl-I	Insert a tab at the current cursor position.
Ctrl-J (F4)	Format the current paragraph.
Ctrl-K (F9)	Cut selected text.
Ctrl-L	Refresh the display.
Ctrl-N	Move to the next line.
Ctrl-O (F3)	Save the current buffer to a file.
Ctrl-P	Move to the previous line.
Ctrl-R (F5)	Insert an external file at the current cursor position.
Ctrl-T (F12)	Invoke the spelling checker.
Ctrl-U (F10)	Paste text at the current cursor position.
Ctrl-V (F8)	Move forward one page of text.
Ctrl-W (F6)	Search for text, neglecting case.
Ctrl-X (F2)	Exit <i>nano</i> , saving the edit buffer.
Ctrl-Y (F7)	Move backward one page of text.

Here's a simple exercise that will give you a quick tour of *nano*. Start *nano* by issuing the command:

```
nano
```

Then type the following short paragraph of text, including the typographical errors:

```
Nano is is a greet editor. I use it for my light-duty tasks editing.
However, when I
need to really get down to business, I prefer vi.
```

Notice that three errors appear:

- The word *is* appears twice
- *Great* is misspelled as *greet*
- The words *editing* and *tasks* appear in the wrong order.

To correct the first error, use the up and left arrow keys to position the cursor on the letter *i* of the first instance of the word *is*. Then, press **Delete** three times to erase the word and the following space.

Next, use the right arrow key to place the cursor on the second (incorrect) letter *e* in the misspelled word *greet*. Type an *a* and press **Delete** to correct the error by replacing the *e* with an *a*.

Now, let's revise the phrase "for my" to read "for all my." Simply use the right arrow keys to move to the letter *m* in the word *my*. Type *all* and a space.

Finally, let's cut and paste to move the words *tasks* and *editing* into their proper sequence. Use the arrow keys to position the cursor on the letter *t* in the word *tasks*. Type **Ctrl-^** to mark the beginning of a selection. Use the right arrow key to select the remainder of the word and the following text, positioning the cursor on the letter *e* in the word *editing*. Cut the selected word by typing **Ctrl-K**. Now, use the right arrow key to move the cursor past the word *editing* and type **Ctrl-U** to paste the text that was cut. The paragraph is now error-free.

Save the paragraph to a file by typing **Ctrl-X**, responding *y* to the question "Save modified buffer?", typing the desired filename, and pressing **Enter**. Use the **less** command to verify that the file was created and has the proper contents.

Chapter 8. Installing Software Using the RPM Package Manager

This chapter explains the Red Hat Package Manager (RPM), a tool that facilitates installing, uninstalling, and upgrading software for your Red Hat Linux system. The chapter explains how to use RPM to find the package associated with an application and how to quickly and easily install the package. It also explains how to use RPM to upgrade packages and query the status of installed packages.

An RPM package (or more simply, an *RPM* or a *package*) is a file that contains executable programs, scripts, documentation, and other files needed by an application or software unit. RPM packages are generally named using a convention that lets you determine the name of the package, the version of the software, the release number of the software, and the system architecture for which the application is intended. [Figure 8-1](#) shows how the components of a package name are arranged.

Figure 8-1. The structure of a package name



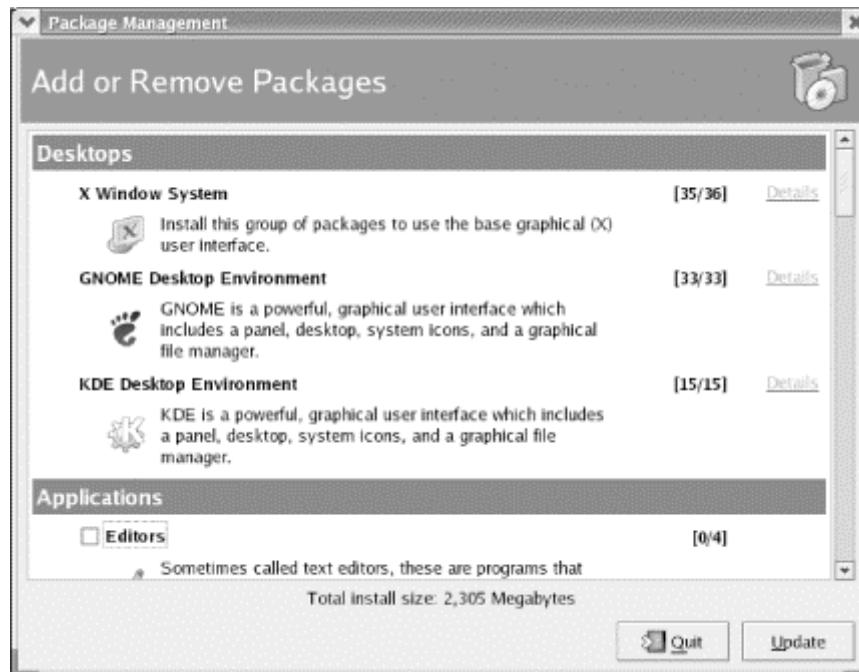
8.1 The Package Management Tool

Red Hat Enterprise Linux and Fedora Core feature a package management tool that's much easier to use than GnoRPM, the tool included in previous versions of Red Hat Linux. To launch the package management tool, choose System Settings → Add/Remove Applications from the main menu. Only the root user can manipulate packages. If you're not logged in as the root user, the tool helpfully prompts you for the root user's password. The tool checks the status of installed packages, as shown in [Figure 8-2](#). This process may require a minute or two. Then, the Add or Remove Packages window ([Figure 8-3](#)) appears.

Figure 8-2. Checking system package status



Figure 8-3. The Add or Remove Packages window



8.1.1 Installing Package Groups and Packages

To install a package package group, select the checkbox next to the name of an uninstalled package group. In [Figure 8-3](#), the Editors package group is not installed, as indicated by the unchecked checkbox to the left of the package group's name. By enabling the checkbox, you can specify that you want to install the package group. Installation does not begin until you click the Update button.

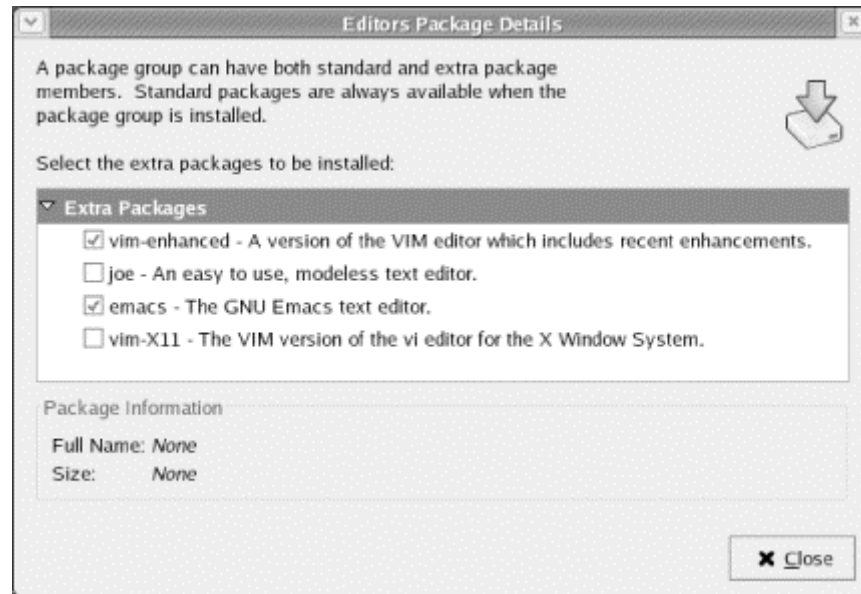
Most package groups contain optional packages that you can install or omit, according to your preference. The numbers to the right of the package group name indicate how many of the optional packages are installed. If you've just specified that a package group should be installed, the numbers indicate how many of the optional packages will be installed when you click the Update button. For example, in [Figure 8-3](#), the 0/4 associated with the Editors package group indicates that none of the four associated packages is currently installed. If you enable the checkbox associated with the Editors package group, the numbers immediately change to 3/4, meaning that three of the four associated packages will be installed when you click the Update button.

To learn more about the optional packages associated with a package group, click the Details link. The Package Details window appears, as shown in [Figure 8-4](#). This window provides checkboxes that you can use to specify which optional packages you want installed. If you can't see the list of package names, expand the tree by clicking the triangle next to the text Extra Packages. When you're done viewing the package details, close the window by clicking Close. You can then install the selected packages by clicking Update. The package management tool will prompt you to insert the appropriate installation CD, in necessary. To close the Add or Remove Packages window without installing or removing any packages, click the Quit button.



The package management tool only installs official Red Hat Enterprise Linux packages that are distributed on the installation media or official Fedora Core packages. However, you can use the **rpm** command to install packages from other sources, as described later in this chapter.

Figure 8-4. The Editors Package Details window



8.1.2 Removing Package Groups and Packages

You can also delete package groups and packages by using the package management tool. To remove a package group, disable the checkbox associated with the package group. To remove a package, open the Package Details window and disable the checkbox associated with the package. No changes are made until you click the Update button of the Add or Remove Packages window.



You can't use the package management tool to remove the package groups for the X Window System, GNOME desktop environment, or KDE desktop environment. Neither can you use the package management tool to delete standard packages associated with a package group. These restrictions protect you against unintended deletion of packages necessary for system operation. If, however, you do want to delete one or more such packages, you can do so using the **rpm** command. If you do use the **rpm** command to remove packages, be sure you avoid removing packages necessary for system operation.

8.1.3 Installing Packages from the Hard Disk

If you have several free gigabytes of disk storage, you may prefer to copy the Red Hat Linux installation media to your hard disk. Doing so avoids the need to insert CDs when installing packages.

To set up your system for installing packages from the hard disk, open a terminal window, become the root user, and issue the following command to create the directory */var/redhat-tree/RedHat*:

```
mkdir -p /var/redhat-tree/RedHat
```

The **-p** flag enables you to create the directory and its parent directory in a single command. Without the flag, the command will fail because the */var/redhat-tree* directory does not exist.

Next, mount installation CD 1 and copy the contents of the *RedHat* directory (*/mnt/cdrom/RedHat*) to the new directory:

```
cp -a /mnt/cdrom/RedHat/* /var/redhat-tree/RedHat
```

Finally, copy the files from installation CD 2, CD 3, and CD 4 to the same location.



You can create the directories and copy the files using a file manager, if you prefer.

To install packages, launch the package management tool by issuing the command:

```
redhat-config-packages --tree=/var/redhat-tree
```

The tool will no longer prompt you to insert a CD; instead, it will obtain package files from the */redhat-tree* directory. To avoid typing this command to install packages, you can create a panel launcher or menu item, as explained in [Chapter 5](#) and [Chapter 6](#).

8.2 The redhat-install-packages Command

Many GUI tools use shell commands behind the scenes. The package management tool uses a special command, **redhat-install-packages**, to install packages. Although the package management tool can install only packages that reside on the installation media, the **redhat-install-packages** command can install RPM packages from any source. For example, you can use this command to install packages that you've downloaded from a web or FTP server.

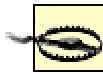
RPM package names generally end with a *.rpm* extension. To install a package using the **redhat-install-packages** command, issue the command:

```
redhat-install-packages package-file.rpm
```

where *package-file.rpm* is the name of the package you want to install. You can install multiple packages by listing several filenames, separating each from the next with a space, like so:

```
redhat-install-packages package-file1.rpm package-file2.rpm package-  
file3.rpm
```

If a desired package requires other packages in order to operate correctly, the other packages must already be installed or must be installed at the same time as the desired package. Otherwise, installation of the desired package will fail. See the upcoming [Section 8.5.1](#) for more information on this potential problem. However, the **redhat-install-packages** command will attempt to automatically install necessary packages that reside on the installation media.



Several Linux distributions publish their software as RPMs. Although the **redhat-install-packages** command is generally capable of installing foreign packages—that is, packages not created for Red Hat Enterprise Linux or Fedora Core—installing foreign packages may leave your system in an unusable state. If you're using Red Hat Enterprise Linux, you should generally install only packages created for Red Hat Enterprise Linux, Version 3. Similarly, if you're using Fedora Core, you should generally install only packages created for Fedora Core 1. You can safely use the **redhat-install-packages** command to install such packages.

8.3 The rpm Command

RPM packages are built, installed, uninstalled, and queried with the **rpm** command. Like other command-line facilities, the **rpm** command gives you almost complete control over its operation. So, you can use the **rpm** command to accomplish feats not possible using the package management tool. For instance, you can use the **rpm** command to install or remove individual packages, install packages from non-Red Hat media, and install packages other than those built by Red Hat. On the other hand, the **rpm** command is sometimes much less convenient to use than the package management tool. So, you use can either the GUI or command line, according to your needs and preferences.

rpm has several modes, each with its own options. The general format of the **rpm** command is:

```
rpm [ options ] [ packages ]
```

The first option generally specifies the **rpm** mode (e.g., install, query, update, build, etc.); any remaining options pertain to the specified mode.

The **rpm** command has built-in FTP and HTTP clients. So, you can specify an *ftp://* or *http://* URL to identify an RPM package stored on a remote host.

Unless the system administrator has specially configured the system, any user can query the RPM database. Most other RPM functions require root privileges. Strictly speaking, it's not necessary for you log in as root to install an RPM package; however, your user account must be authorized to access and modify the files and directories required by the package, including the RPM database itself. Generally, the easiest way to ensure such access is by logging in as root.

8.3.1 General rpm Options

The following **rpm** options can be used with all modes:

--ftpport *port*

Use *port* as the FTP port.

--ftpproxy *host*

Use *host* as a proxy server for all transfers. Specified if you are FTPing through a firewall system that uses a proxy.

--help

Print a long usage message (running **rpm** with no options gives a shorter usage message).

--quiet

Display only error messages.

--version

Print the version number of **rpm**.

-vv

Print debugging information.

8.4 Querying the RPM Database

You can query RPM's database, which lists the packages installed on your system. For example, to display a simple description of an installed package, use a command like this one:

```
rpm -q package
```

In this command, *package* is the name of the package you want RPM to describe. In response, RPM prints the package name, version, and release number.

Rather than use the **-q** option and the package name, you can use any of the following alternative options:

-a

Causes RPM to display information about all installed packages

-f *file*

Causes RPM to display information about the package that owns *file*

-p *packagefile*

Causes RPM to display information about the package contained in *packagefile*

You can also tailor the output of an RPM query, by specifying one or more of the following options:

- c**
Causes RPM to display a list of configuration files included in the package
- d**
Causes RPM to display a list of documentation files included in the package
- i**
Causes RPM to display the package name, description, release number, size, build date, installation date, vendor, and other information
- l**
Causes RPM to display the list of files that the package owns
- s**
Causes RPM to display the state of all the files in the package—normal, not installed, or replaced

For example, the command:

```
rpm -qid rhide
```

displays information about the *rhide* package, including a list of documentation files included in the package.

8.5 Installing a Package

To install a package, log in as root and issue the following command from a shell prompt:

```
rpm -ivh package
```

where *package* specifies the name of the file that contains the package. You can specify multiple packages, as long as you include a space to separate each package name from its neighbor. For example, the following command installs both the *nano* and *mutt* packages from files in the current directory:

```
rpm -ivh nano-1.2.1-4.i386.rpm mutt-1.4.1-3.i386.rpm
```

The options used with the **rpm** command include:

- i**
This option specifies that RPM should install the package or packages given as arguments.
- h**
This option specifies that RPM should print hash marks (#) as it installs the package as a visible indication of progress.

-v

The verbose option specifies that RPM should print messages that summarize its actions and progress.

Generally, RPM will successfully install the specified package. However, errors can occur. RPM may report:

- That the package is already installed
- That a package file conflicts with a file from another package
- A failed dependency

The next three sections explain how to resolve these errors.

8.5.1 Package Is Already Installed

If a package has already been installed, RPM will not overwrite the package without your permission:

```
# rpm -ivh bad-1.0-1.i386.rpm
bad package bad-1.0-1 is already installed
```

If you want to overwrite the package, add the **—replacepks** option to your command:

```
rpm -ivh --replacepks bad-1.0-1.i386.rpm
```

It may be more appropriate to update the package. Updating the package leaves its configuration files intact, whereas overwriting the package replaces the configuration files with files containing default options. An upcoming section shows you how to update a package.

Of course, it's also possible that you should do nothing. You may have attempted to install the package without first checking whether it's already installed and operational. In that case, you can use RPM to verify that the package is installed correctly and update or overwrite the package only if RPM reports problems.

To verify a package against a package file, issue the following command from a shell prompt:

```
rpm -Vp package
```

In the command, *package* specifies the name of the file that contains the package; for example, *basesystem-8.0-1.rpm*. In verifying a package, RPM compares the installed files with the original package contents. If RPM detects no discrepancies, no output will appear. Otherwise, RPM displays a line for each file that differs from the original package contents.

[Figure 8-5](#) shows the structure of such a line.

Figure 8-5. The structure of RPM's discrepancy report



The first eight characters report discrepancies; each character has the meaning described in [Table 8-1](#). Following the list of discrepancies, you may see the letter *c*, which denotes that this is a configuration file. Finally, the filename appears.

Table 8-1. Package file discrepancy codes	
Code	Meaning
.	No discrepancy
5	MD5 checksum discrepancy
D	Device discrepancy
G	Group discrepancy
L	Symbolic link discrepancy
M	Access mode or file type discrepancy
S	File size discrepancy
T	File modification time discrepancy
U	User discrepancy

The configuration files associated with a package are generally modified during installation and use, so it's not unusual for the content of configuration files to differ from that of the original files. You should generally ignore MD5 checksum and file size discrepancies that pertain to configuration files.

You can verify a package against the information recorded in the RPM database when the package was installed. To do so, issue a command of the form:

```
rpm -V package
```

Here, *package* specifies the name the package, for example, *basesystem*. Another form of the **rpm** command lets you verify packages that contain a specified file:

```
rpm -vf path
```

In this form, *path* specifies the absolute pathname of the file. The output of this command is the same as that given earlier.

8.5.2 Conflicting File

Conflict is RPM's term for a situation in which two packages include one or more identically named files that have different contents. For example, suppose that package **a** contains a file named `/etc/superconfig` having 12 lines and that package **b** contains a file named `/etc/superconfig` having 13 lines. The two packages conflict, because the two instances of `/etc/superconfig` are inconsistent.

In the event of a conflict, only one of the two conflicting packages is likely to work properly. If you instruct RPM to install a package and RPM finds that one or more of the package files conflict with existing files, RPM reports the conflict and terminates without installing the package:

```
# rpm -ivh bad-1.0-1.i386.rpm
bad /bin/badfile conflicts with file from good-1.0-1
```

In its report, RPM gives the name of the file and the name of the package that originally installed the file. You can force RPM to install the package, by using the **—replacefiles** option:

```
rpm -ivh --replacefiles package
```

In response, RPM saves and then replaces any conflicting configuration files; it overwrites other types of files. However, using the **—replacefiles** option is a bit like hitting a malfunctioning mechanical device with a hammer. Sometimes, it's what you must do; but, more often, it merely causes damage. The better way to resolve a conflict is to decide which of the two conflicting packages you prefer. Then, delete the unwanted package and install the preferred one.

8.5.3 Failed Dependency

Packages are not always self-contained: some packages require that other packages be installed before they operate correctly. RPM can identify such dependencies. If you attempt to install a package before you install other packages it requires, RPM reports a "failed dependency" and terminates without installing the package:

```
# rpm -ivh bad-1.0-1.i386.rpm
failed dependencies:
  mefirst is needed by bad-1.0-1
```

To resolve a failed dependency, you should install the missing package (or packages) and then install the desired package. If you prefer, you can force RPM to install the package; however, the package may not operate correctly. To force package installation, specify the **—nodeps** option:

```
rpm -ivh --nodeps bad-1.0-1.i386.rpm
```

However, a forcibly installed package is unlikely to work correctly.

8.6 Uninstalling a Package

To uninstall a package, type:

```
rpm -e package
```

In this command, *package* is the name of the package, not the name of the package file. The name should omit the architecture; it can also omit the package version or package version and release number. For example, you can erase the **nano** package by issuing either of the following commands:

```
rpm -e nano-1.2.1-4rpm -e nano
```

If you attempt to uninstall a package on which another package depends, RPM will report a dependency error and terminate without uninstalling the package. You can force RPM to uninstall the package by using the **—nodeps** option:

```
rpm -e --nodeps package
```

However, doing so will probably cause the dependent package to cease working properly. Therefore, you shouldn't use the **—nodeps** option very often.

8.7 Updating a Package

When you update (upgrade) a package, RPM installs the new version of the software but attempts to leave your existing configuration files intact. You can update a package by using the **-U** option of the **rpm** command:

```
rpm -Uvh package
```

When you update a package, RPM automatically uninstalls the old version of the package before installing the new one.



If no old version of the specified package exists, RPM simply installs the new version. Therefore, you can use the **-U** option to install or update a package; many Linux users avoid the **-i** (install) option, always using the **-U** option instead. However, you should use the **-i** option rather than the **-U** option when installing a package containing an updated Linux kernel. By using the **-i** option, you leave the original kernel in place, so that you can use it to boot your system if the new kernel does not work correctly with your system.

If RPM determines that your existing configuration files may be incompatible with those of the new version of the package, RPM will save a copy of the existing files. In that case, you need to examine the old and new files and determine what the proper configuration should be. The documentation that accompanies the package should assist you in this process.

If you attempt to update an existing package using an older version of the package, RPM will report an error and terminate without performing the update. To force RPM to perform the update, use the **—oldpackage** option:

```
rpm -Uvh --oldpackage package
```

8.8 Freshening a Package

From time to time, Red Hat issues updated packages that correct functional or security-related problems with released packages. You can use Red Hat Network to obtain and install updated packages. However, if you prefer, you can download updated packages via FTP and install them using the **rpm** command. Red Hat makes updated packages available on the public FTP servers *ftp.redhat.com* and *updates.redhat.com*.

To install an updated package, use the **-F** flag, which stands for *freshen*. For instance, to install an updated version of the **gnupg** package, issue the command:

```
rpm -Fvh gnupg-1.2.1-10.i386.rpm
```

By specifying **-F** rather than **-U**, you instruct RPM to install the updated package only if an earlier version of the package is already present. This lets you use wildcards to specify entire sets of updated packages:

```
rpm -Fvh *.rpm
```

This command will not install packages that aren't already installed. But, it will install updated versions of any existing packages.

8.9 Advanced RPM Techniques

Because you invoke the **rpm** command by using the shell, just as you do any other program, you can combine options and arguments to perform a variety of useful tasks. Consider the following examples:

```
rpm -Va
```

Verifies every installed package. You might find this command useful if you accidentally deleted some files. The output of the command would help you determine what packages, if any, suffered damage.

```
rpm -qf /usr/bin/mystery
```

Displays the name of the package that owns the specified file.

```
rpm -Vf /usr/bin/mystery
```

Verifies the package that owns the file */usr/bin/mystery*.

```
rpm -qdf /usr/bin/puzzle
```

Lists the documentation files associated with the package that owns the file */usr/bin/puzzle*. This could be helpful, for example, if */usr/bin/puzzle* is a program you're having difficulty using.

8.10 Keeping Your Red Hat Enterprise Linux System Up to Date

Red Hat provides a service known as Red Hat Network (<http://rhn.redhat.com>), designed to help you keep your Red Hat Enterprise Linux system up-to-date and secure. Access to Red Hat Network is included with your subscription to Red Hat Enterprise Linux. If you purchase Red Hat Professional Workstation, you receive a one-year subscription to Red Hat Network. If you're using Fedora Core, you can keep your system up to date by using Yum, which is described in [Section 8.11](#).

The Red Hat Network provides access to security alerts, bug fix alerts, and enhancement alerts published by Red Hat. Updated packages can be downloaded or automatically installed via the Red Hat Update Agent. The Red Hat Network is of significant benefit to administrators of multiple systems, who might otherwise have difficulty applying patches to close security loopholes and fix problems quickly.

8.10.1 Using Red Hat Network

To use Red Hat Network, your computer must be able to access the Internet. To set up your computer, you must perform two steps:

1. Sign up for Red Hat Network.
2. Create a system profile for your computer.

Once your computer is set up to use Red Hat Network, updates can be accomplished automatically based on a schedule or manually, by using the Red Hat Update Agent. The following sections describe these procedures as they existed at the time of writing. However, Red Hat may revise the operation of Red Hat Network at any time. For more information on using Red Hat Network, see the *Red Hat Network User Reference Guide*, available at <http://www.redhat.com/docs/manuals/RHNetwork>.

8.10.1.1 Signing up

To sign up for Red Hat Network, point your browser to the main Red Hat Network web page, <http://rhn.redhat.com>. Then click the link marked "Create Account." The web page shown in [Figure 8-6](#) appears.

Figure 8-6. The Red Hat Network Create Account web page



Click the link marked "Create a Personal Account." The web page shown in [Figure 8-7](#) appears. Fill in the requested information and then click the Create Account button at the bottom of the form.

Figure 8-7. The Red Hat Network personal account sign-up page

8.10.1.2 Activating your product

When you purchase Red Hat Enterprise Linux WS, you should receive an email or printed card explaining how to activate your product. The instructions include a product ID that uniquely identifies your license. Before creating a system profile, you should activate your product by following the instructions you received.



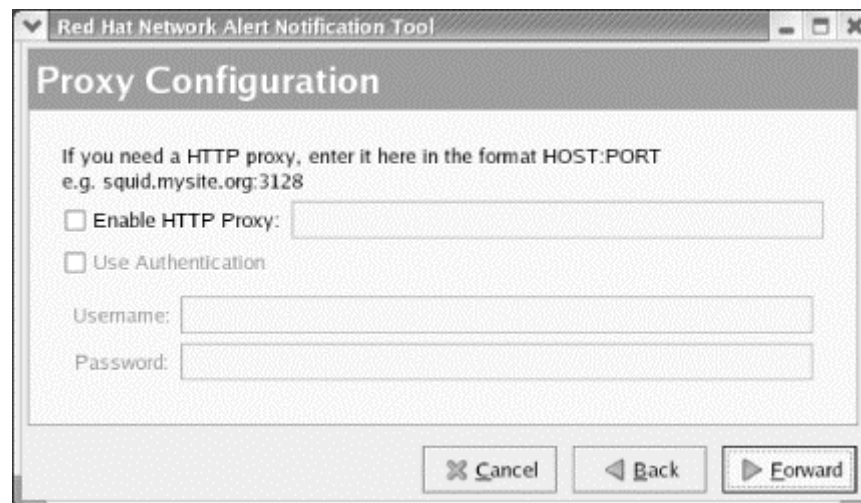
Be sure to keep a permanent record of your Red Hat Network user ID and password, and your product ID. You may need these if your system's data becomes damaged or corrupted.

8.10.1.3 Creating a system profile

Once you've activated your product, you can create a profile describing your system. To do so, click the Red Hat Network Alert Notification Tool icon that appears near the right of the desktop panel. If you've not yet created a profile, the icon will resemble a red ball with a superimposed exclamation mark. Otherwise, it may take the form of a green, blue, or gray ball.

Clicking the Notification Tool icon starts a configuration dialogue. First, the dialogue asks you to read and accept the Red Hat Network terms of service. Then, it presents the Proxy Configuration dialog box ([Figure 8-8](#)).

Figure 8-8. The Proxy Configuration dialog box



If your computer is behind a firewall that requires you to use a proxy sever to access the Web, you must enable the HTTP proxy and specify its hostname and port. Most users don't require a proxy. If you suspect that you do, you can likely obtain the proper information by inspecting the configuration of your Windows web browser, if any. Otherwise, you can obtain the necessary configuration information from your network administrator.

After you complete the dialogue, the Configuration Complete dialog box appears. Click Apply and the Notification Tool will attempt to access Red Hat Network. Unless you've already configured a profile for your system, the Red Hat Network Registration dialog box shown in [Figure 8-9](#) appears.

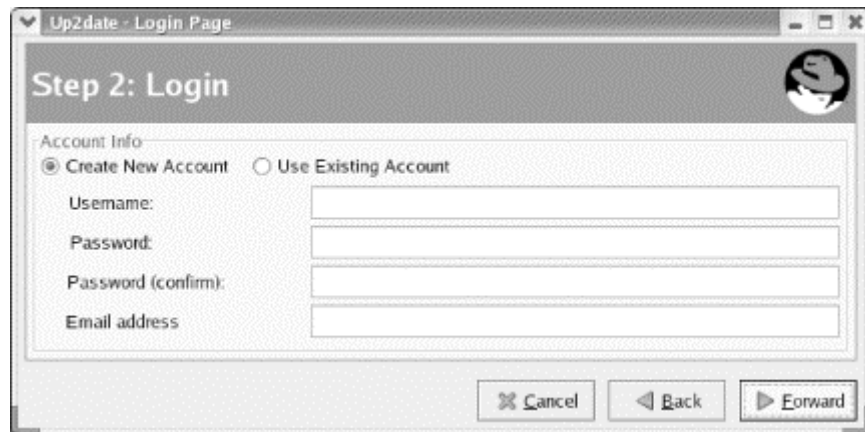
Figure 8-9. The Red Hat Update Agent dialog box



If the dialog box does not appear, you can manually initiate the registration process. To do so, open a terminal window, become root, and issue the command:
`rhn_register`

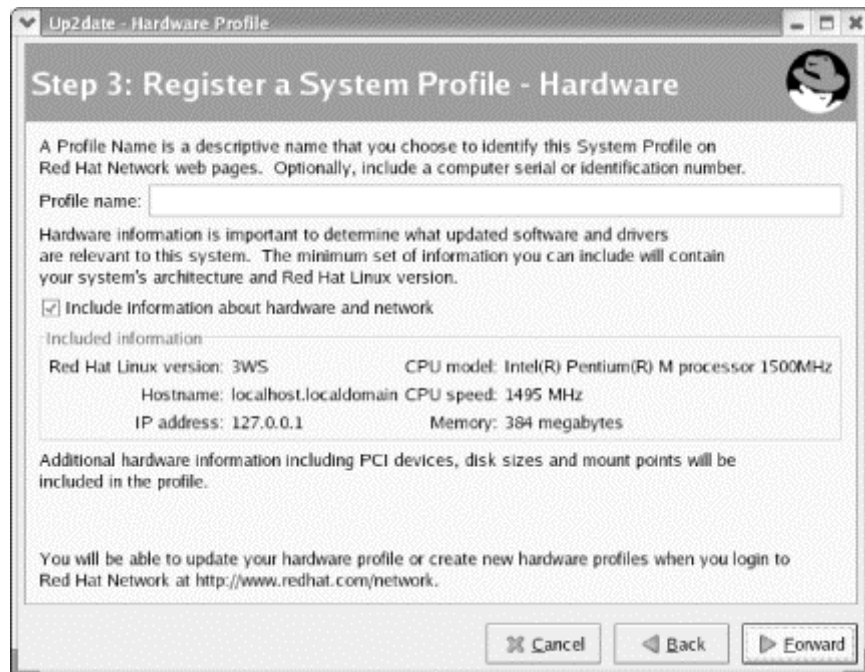
To register your computer with Red Hat Network, click Forward, review Red Hat's privacy statement and, if you concur with the statement, click Forward again. The Login dialog box ([Figure 8-10](#)) appears. If you created a Red Hat Network account as instructed earlier, enable the radiobutton labeled Use Existing account; otherwise enable the radiobutton labeled Create New Account. Specify your account information and click Forward.

Figure 8-10. The Login dialog box



The Register a System Profile—Hardware dialog box appears ([Figure 8-11](#)). Specify a name for your PC and click Forward.

Figure 8-11. The Register a System Profile—Hardware dialog box



The Register a System Profile—Packages dialog box appears, as shown in [Figure 8-12](#). This dialog box allows you to conceal from Red Hat the identities of packages installed on your system. Since you can't use Red Hat Network to manage packages it isn't aware of, you should not generally conceal any package information from Red Hat Network. Click Forward to proceed. The Send Profile Information to Red Hat Network dialog box appears. Click Forward to transmit your system's profile.

Figure 8-12. The Register a System Profile—Packages dialog box

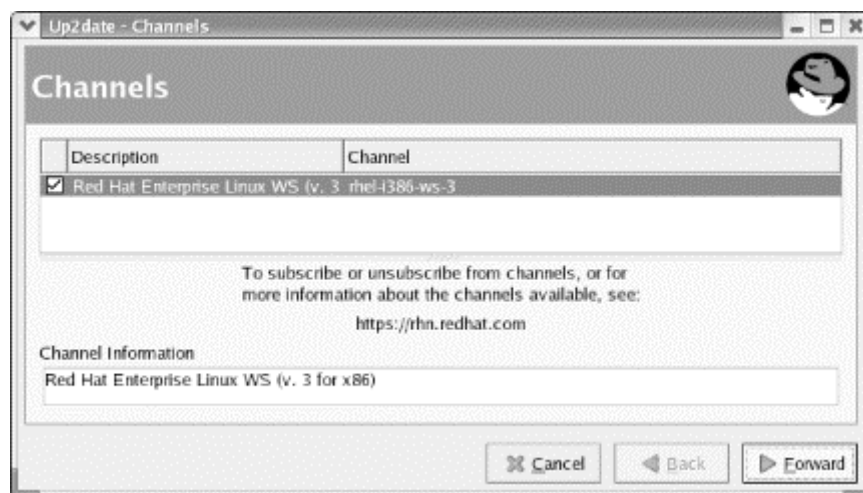


The Channels dialog box appears, as shown in [Figure 8-13](#). This dialog box enables you to subscribe to the channel that provides updates to Red Hat Enterprise Linux WS. Normally, only one channel should appear and you should not need to make changes to the defaults. Click Forward to continue.



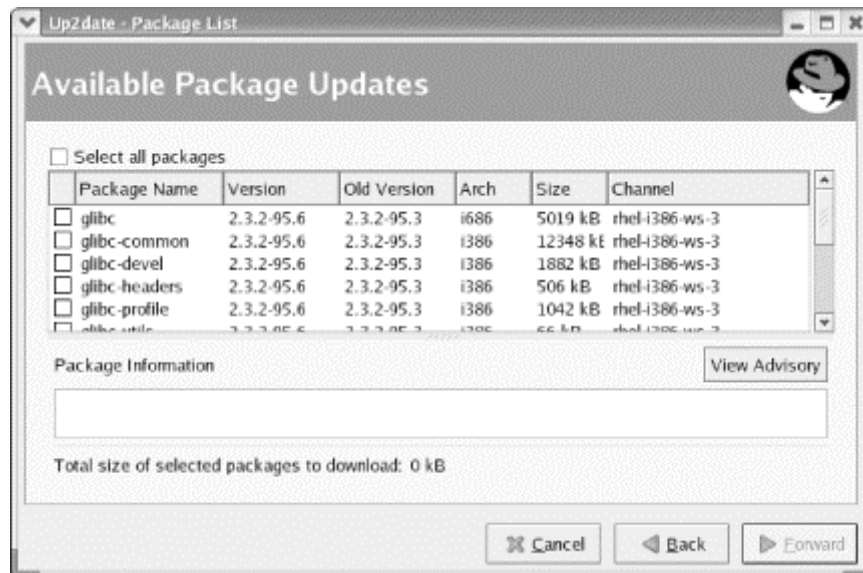
If the Channels dialog box does not include the Red Hat Enterprise Linux WS channel, it's likely that your product hasn't been properly activated. Using the information on your product activation email or card, contact Red Hat technical support for assistance.

Figure 8-13. The Channels dialog box



As shown in [Figure 8-14](#), the Update Agent obtains and displays a list of installed packages for which updates are available. To update a package, enable the associated checkbox. To update all available packages, enable the checkbox labeled Select all packages. You can view information about the updates associated with a package by selecting the package name and clicking View Advisory. Select the packages you want to update and then click Forward. If you do not want to update any packages at this time, click Cancel.

Figure 8-14. The available package updates



The Update Agent may first present a similar dialog box titled **Packages Flagged to be Skipped**. By default, the Update Agent does not automatically update the Linux kernel, since a failed kernel update can result in an unbootable system. If you want to update a kernel package or other package that is flagged to be skipped, enable the associated checkbox and click Forward.

After you click Forward, the Update Agent will retrieve any specified packages and update your system. Separate dialog boxes present the progress and completion of package retrieval and installation.

8.10.1.4 Updating your system

After you initially set up your system to use Red Hat Network, you can update it at any time. To do so, choose **System Tools** → **Red Hat Network** from the main menu. Or, click the ball icon that represents the Notification Tool. If the icon is red, new updates are available. To avoid a security breach, you should download and install the updates right away. Timely installation of updates is one of the most important measures in defending your system against possible attack.

8.11 Keeping Your Fedora Core System Up to Date

Since Fedora Core isn't officially supported by Red Hat, Fedora Core users don't have access to Red Hat Network. However, Fedora Core includes the Alert Notification and up2date tools, which can automatically install software updates.

You don't need to activate a software license in order to update your Fedora Core system, but you do need to configure your system. The procedure used to configure and update your system closely resembles that appropriate for updating Red Hat Enterprise Linux. Here are the steps you should follow:

1. Launch the Alert Notification Tool configuration sequence by right-clicking the Alert Notification Tool icon. The series of dialog boxes that appears closely resembles that related to Red Hat Enterprise Linux, described in the preceding section.
2. Launch the up2date tool by clicking the Alert Notification Tool icon. Complete the series of dialog boxes, beginning with the Welcome to Red Hat Update Agent dialog box. The series closely resembles that of Red Hat Enterprise Linux, described in the preceding section. However, it omits several dialog boxes that relate to Red Hat Network.



If you're prompted to install the Red Hat, Inc. public GPG key, be sure to click Yes. Otherwise, system updates will fail.

Once you've configured the Alert Notification tool, its color indicates whether updates are available. If the tool is red, click the tool icon, review the available updates, and install those you consider appropriate. In general, you should install all available updates as soon as they become available. However, you may prefer to postpone installing available updates—especially large updates that entail time-consuming downloads—to a convenient time.

Chapter 9. Configuring and Administering Linux

GNOME and KDE include two menus that provide access to utilities for configuring and administering your system: System Settings and System Tools. Most of these utilities are also available via the Start Here folder. This chapter explains the utilities, equipping you to perform common system administration tasks. In particular, the chapter explains how to manage user accounts, how to configure a printer, how to configure your system's sound adapter, how to view system log files, and how to administer services. Many of these operations require root access. If you launch one of the tools when not logged in as root, the

tool will conveniently ask you for the root password. Once you've logged in as the root user, the desktop will display an icon resembling a set of keys. So long as this icon appears, the system will automatically extend you root privileges when necessary. You can cancel this authorization by clicking on the icon and then clicking Forget Authorization.

9.1 Configuring Your System by Using the System Settings Menu

The GNOME and KDE System Settings menu provides access to 15 utilities that help you configure your system. [Chapter 8](#) explained one of these, the package management tool. The tools available are:

Add/Remove Applications

Provides access to the Package Management tool, which lets you install and remove RPM packages

Authentication

Provides access to the Authentication Configuration tool, which lets you configure password settings

Date & Time

Provides access to the Date/Time Properties tool, which lets you configure the time zone, date, and time

Display

Provides access to the Display Settings tool, which lets you configure monitor resolution, color depth, and other display characteristics

Keyboard

Provides access to the Keyboard tool, which lets you choose the keyboard appropriate to your system

Language

Provides access to the Language Selection tool, which lets you choose the current language from among those for which you've installed support

Login Screen

Provides access to the GDM (GNOME Display Manager) Setup tool, which lets you configure how people log in under X

Mouse

Provides access to the Mouse Configuration tool, which lets you choose the mouse appropriate to your system and enable or disable 3-button emulation

Network

Provides access to the Network Configuration tool, which lets you configure network, modem, virtual private network (VPN), and wireless connections and related settings

Printing

Provides access to the Red Hat Printer Configuration tool, which lets you configure printers and control print queues

Root Password

Provides access to a tool that enables you to change the root password

Security Level

Provides access to the Security Level Configuration tool, which lets you configure a firewall to protect your system from network attacks

Soundcard Detection

Provides access to the Sound Card Detection tool, which configures, and enables you to verify the operation of, your sound card

Users and Groups

Provides access to the Red Hat User Manager, which lets you configure user accounts and groups. Depending on the packages installed on your system, you may see fewer or more items on the System Settings menu.

In addition, the System Settings menu contains a More System Settings submenu, on which you can find one more tool: the Desktop Switching Tool, which lets you choose between the GNOME and KDE desktop if you've installed both desktops. You may also find a Server Settings submenu, which you can use to configure servers you've installed. This chapter explains how to control services. [Chapter 12](#) explains how to install and configure commonly used servers.

Most of the System Settings tools function much like the Red Hat Linux installation procedure. This section focuses on tools that do not resemble installation procedure steps, namely:

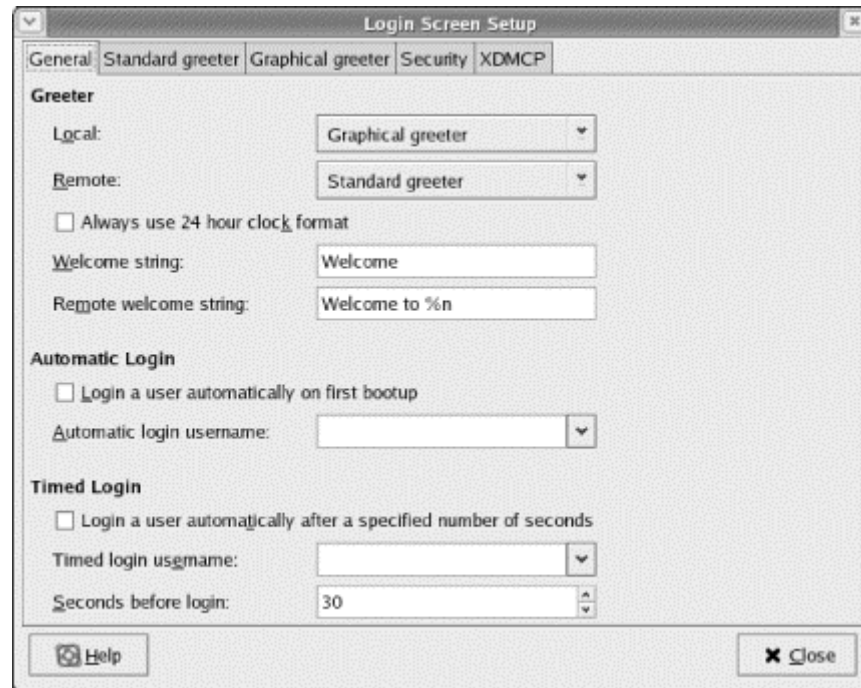
- GDM Setup tool, accessed via the Login Screen menu item
- Red Hat Printer Configuration tool
- Sound Card Configuration tool
- Red Hat User Manager
- Desktop Switching Tool

The Network Configuration tool provides many settings and functions. Explanation of the Network Configuration tool is deferred to [Chapter 10](#) and [Chapter 11](#).

9.1.1 Setting Up X Logins

The GDM Setup tool, accessible via System Settings → Login Screen from the main menu, lets you specify options related to X logins. [Figure 9-1](#) shows the tool. Most of the options are cosmetic. For example, you can choose whether times are displayed in 12- or 24-hour format. However, the Security and XDMCP tabs provide options that you should consider changing.

Figure 9-1. The GDM Setup tool



By default, the XDMCP tab disables XDMCP, the facility that provides remote X terminals and sessions the ability to login to your system. If you have X terminals or additional Linux or Unix hosts, you may find it convenient to be able to remotely log in to your system. To provide this capability, enable the Enable XDMCP checkbox. The change takes effect when you close the GDM Setup tool. Thereafter, X terminals and X sessions on your local network should be able to remotely log in to your system.



If you've enabled XDMCP but are unable to obtain an X login screen, check your firewall settings by using the Security Level Configuration tool. It's likely that your firewall is configured to block remote X access.

If you enable XDMCP, you should generally make a second configuration change. By default, the Security tab enables the setting Allow root to login remotely with GDM. Most users do not need to remotely login as root and therefore don't require this setting to be enabled. By disabling the setting, you can prevent unauthorized persons from successfully logging in as root from remote X terminals or sessions. Therefore, you should generally disable this setting.

9.1.2 Configuring a Printer

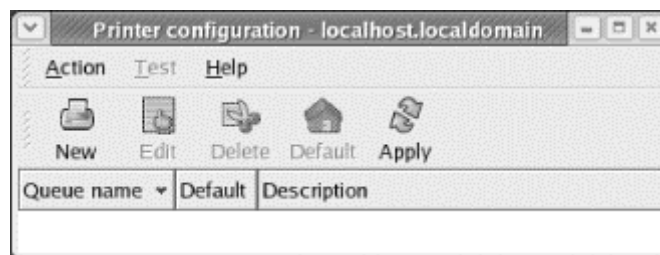
Before you can print, you must configure a printer. Red Hat Enterprise Linux and Fedora Core support local printers attached to your system's parallel port and remote printers that your system accesses via the network. Before you can configure a remote printer, you must first configure networking, as explained in [Chapter 11](#).



If your local printer attaches via the USB port, you can configure it using the same procedure used for a printer attached via the parallel port. However, please bear in mind that neither Red Hat Enterprise Linux nor Fedora Core flawlessly supports USB devices.

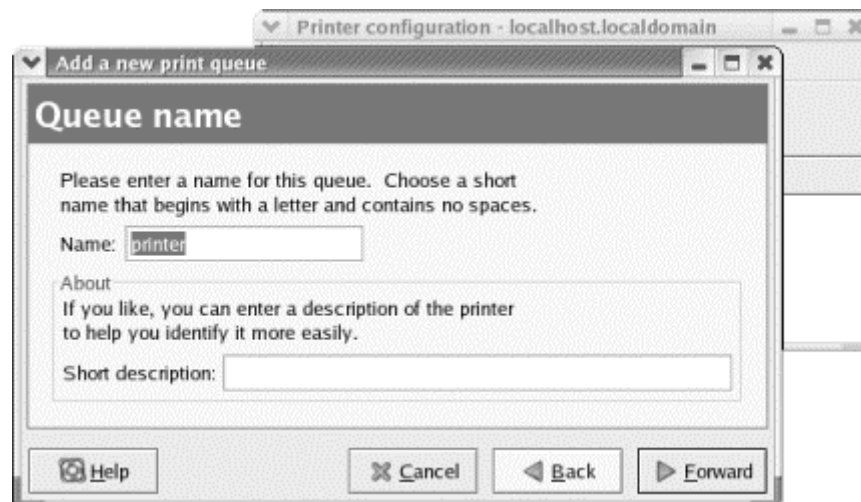
To configure a local printer, launch the printer configuration tool by selecting System Settings → Printing from the main menu. The Red Hat printer configuration tool ([Figure 9-2](#)) appears.

Figure 9-2. The printer configuration tool



First, create a new printer by clicking the New icon. The Add a New Print Queue wizard appears. Click Forward to proceed. The "Queue name" dialog box ([Figure 9-3](#)) appears.

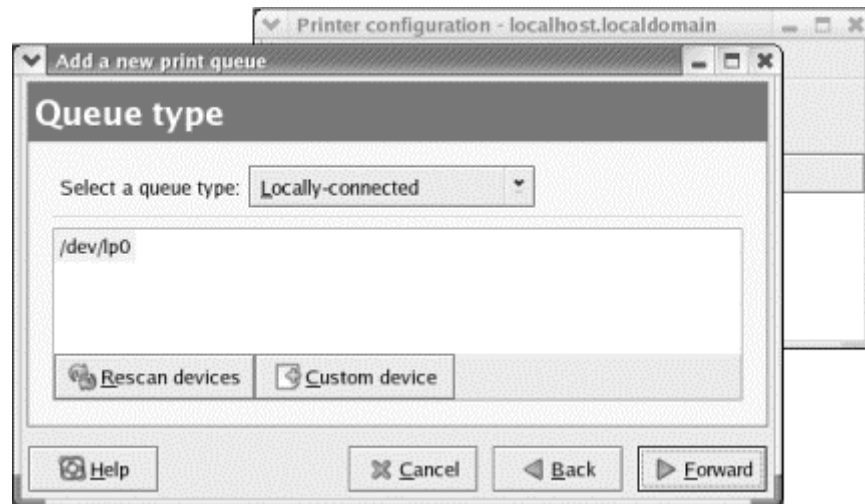
Figure 9-3. The Queue name dialog box



Every print queue has a name. The most commonly used name for the default print queue is *lp*. Type **lp** or another print queue name of your choice in the text box labeled Name and click Forward.

The "Queue type" dialog box ([Figure 9-4](#)) appears.

Figure 9-4. The Queue type dialog box

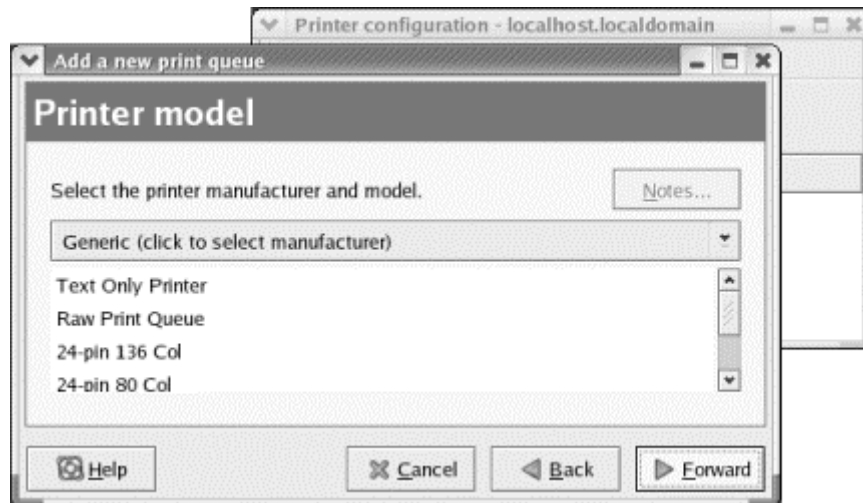


This dialog box shows the parallel ports associated with your system. Linux numbers parallel ports starting with zero, so the port designated */dev/lp0* corresponds to the device known by Microsoft Windows as LPT1.

Select the default queue type, Locally-connected. Then, select the device to which the locally connected printer is attached and click Forward. If the appropriate device does not initially appear, you may be able to use the Rescan Devices button to cause it to appear. Alternatively, you can use the Custom device button to launch a dialog box that enables you to manually specify the device. Click Forward.

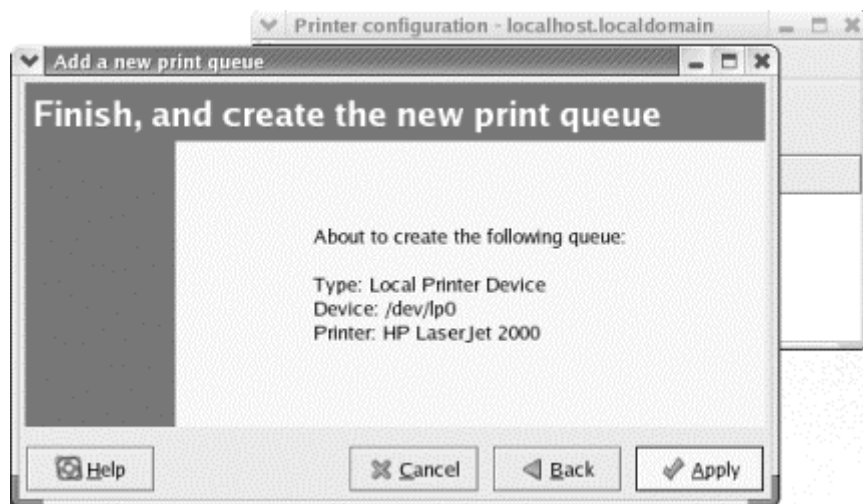
The "Printer model" dialog box ([Figure 9-5](#)) appears. Select the make of your printer by clicking the caret that appears at the right of the listbox labeled Select the printer manufacturer and model. From the sublist that appears, select the model of your printer. Then click Forward.

Figure 9-5. The Printer model dialog box



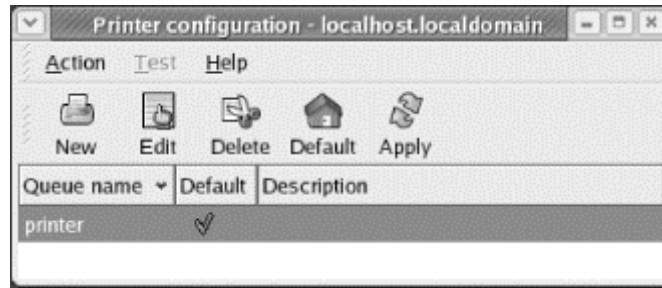
The "Finish, and create the new print queue" dialog box ([Figure 9-6](#)) appears. Check the information and use the Back button to correct any errors. When you're satisfied with your specifications, click Apply.

Figure 9-6. The Finish, and create the new print queue dialog box



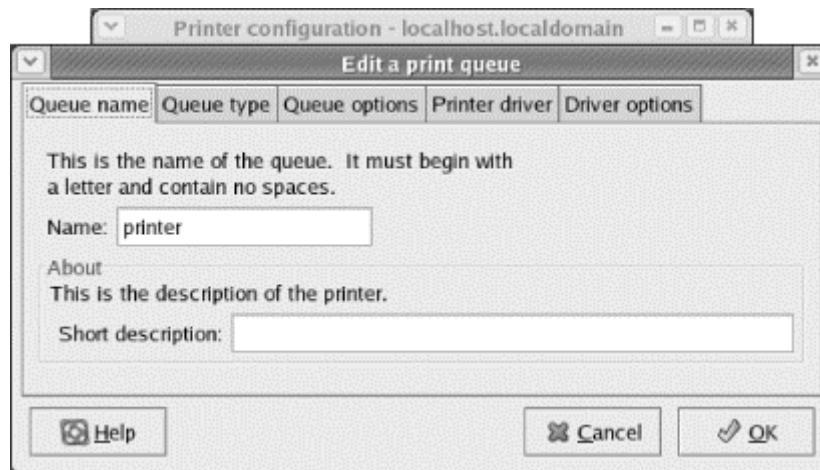
A message box appears, asking if you'd like to print a test page. In order to verify that your printer is operational, you should generally click Yes. Doing so sends a sample page to the printer. After a few seconds, another message box should inform you that the test page has been printed. When you dismiss the message box, the printer configuration tool ([Figure 9-7](#)) appears.

Figure 9-7. The printer configuration tool



If the test page was actually printed, you know that the printing facility is operational. Otherwise, you should check the printer status and cables. If the sample page still doesn't appear or appears incorrectly, select the printer and click Edit. Use the "Edit a print queue" dialog box that appears ([Figure 9-8](#)) to select a different print driver or revise options associated with the current driver.

Figure 9-8. The Edit a print queue dialog box



When you exit the printer configuration tool, you're prompted to save your configuration changes. If you choose not to do so, recent changes are discarded.

9.1.3 Changing the root User Password

You can change the password associated with the *root* user account by using the Root Password tool. To do so, select System Settings Root → Password from the main menu. A dialog box ([Figure 9-9](#)) appears. Type the desired password twice and click OK. If the two passwords match, the password is immediately changed. Otherwise, the tool gives you another opportunity to change the password.

Figure 9-9. The Root Password tool



The system evaluates passwords and can determine that a password you specify is insecure. In such a case, it may prevent you from using the insecure password or, if you are the root user, it may merely inform you that the password is insecure. Unless your PC is physically secure and never connected to a network, you should choose only secure passwords.

9.1.4 Configuring Sound

If your system includes a compatible sound adapter, you can use the Soundcard Detection tool to automatically configure your adapter. If you're unsure whether your system's sound adapter is supported, check the Red Hat Enterprise Linux hardware compatibility database at <http://hardware.redhat.com>. Several popular cards are not fully compatible with Red Hat Enterprise Linux, so it's best to check the database before wasting time trying to configure incompatible hardware.



Unfortunately, the Fedora Project does not yet provide a hardware compatibility database. However, you can use the Red Hat Enterprise Linux compatibility database as a general guide to devices compatible with Fedora Core.

To configure your system's sound adapter, choose System Settings Soundcard Detection from the main menu. The Audio Devices dialog box ([Figure 9-10](#)) appears.

Figure 9-10. The Audio Devices dialog box



The soundcard detection tool probes your system, seeking supported sound adapters. After a sound adapter has been identified, the tool displays the vendor and model of your sound card and the associated Linux kernel module name. If the tool was unable to find a sound adapter, it displays the text "No soundcards were detected."

If your system's sound adapter was successfully probed, you can click the Play test sound button to test the adapter. If you don't hear the sound, check that your speakers are plugged in and, if necessary, powered on. Otherwise, you may spend time trying to reconfigure a sound adapter that's actually working fine.

If the test sound works, but you don't hear sound at other times, check the mixer levels by choosing Sound & Video → Volume Control from the main menu. If you're using KDE, you should also check Control Center → Sound & Video → Sound System → Mixer. You may find that a volume control is set too low or a mixer setting is preventing you from hearing sounds. Be careful when adjusting volume and mixer settings. Loud noises can damage equipment, your hearing, and relationships with neighbors.

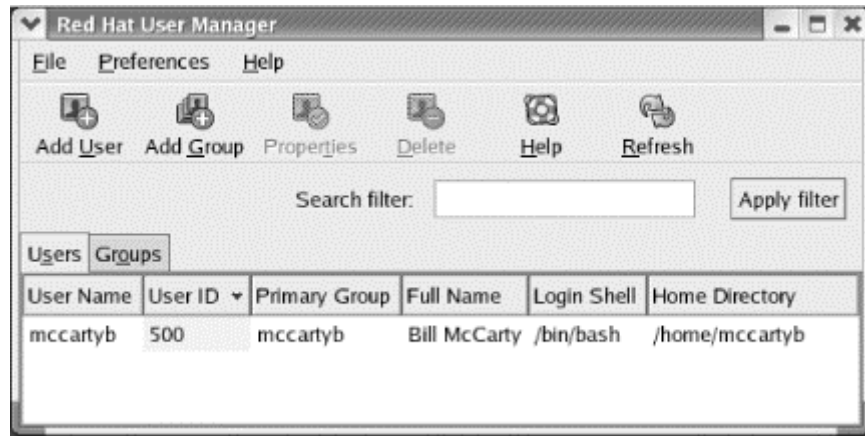


If you're using GNOME, you can enable desktop sounds by choosing Preferences → Sound to configure a sound server to start when the desktop is launched. If you're using KDE, choose Control Center → Sound & Multimedia Sound System.

9.1.5 User and Group Administration

The Red Hat User Manager tool lets you administer users and groups. To launch the tool, select System Settings → Users and Groups from the main menu. The Red Hat User Manager tool ([Figure 9-11](#)) appears.

Figure 9-11. The Red Hat User Manager



The tool presents a scrollable list of user accounts (users) and displays the following information about each user account:

User Name

The login name associated with the user account.

User ID

The numeric ID associated with the user account. This ID is automatically assigned when the user account is created and is unique to each user account.

Primary Group

The name of the primary user group associated with the user account. Accounts used by people rather than by system processes generally have an associated primary group having the same name as the user account.

Full Name

The name of the person or process that owns the user account.

Login Shell

The login shell assigned to the user account. Assigning */bin/false* or */sbin/nologin* prevents the user account from logging in.

Home Directory

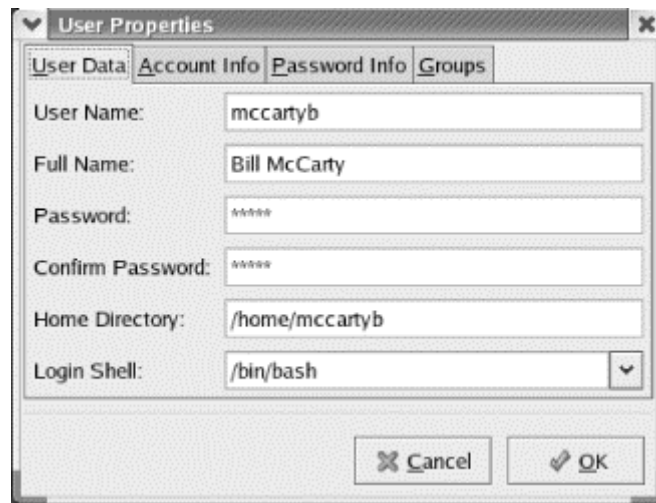
The home directory associated with the user account. When the user logs in, this directory is set as the current working directory.

When many user accounts are configured, it may be inconvenient to scroll through the list. You can use the text box labeled Search filter to display only user accounts having names matching a specified pattern. Type the pattern in the text box and click Apply filter.

9.1.5.1 Modifying a user account

To modify a user account, click the desired account and click Properties. The User Properties dialog box ([Figure 9-12](#)) appears.

Figure 9-12. The User Properties dialog box



The User Data pane of the User Properties dialog box lets you view and change a variety of properties:

User Name

The login name associated with the user account.

Full Name

The name of the person or process that owns the user account.

Password

The password to be associated with the user account.

Confirm Password

The password to be associated with the user account. The password is specified twice in order to reduce the risk of assigning an incorrect password due to a typing error.

Home Directory

The home directory associated with the user account. When the user logs in, this directory is set as the current working directory.

Login Shell

The login shell assigned to the user account. Assigning `/bin/false` or `/sbin/nologin` prevents the user account from logging in.

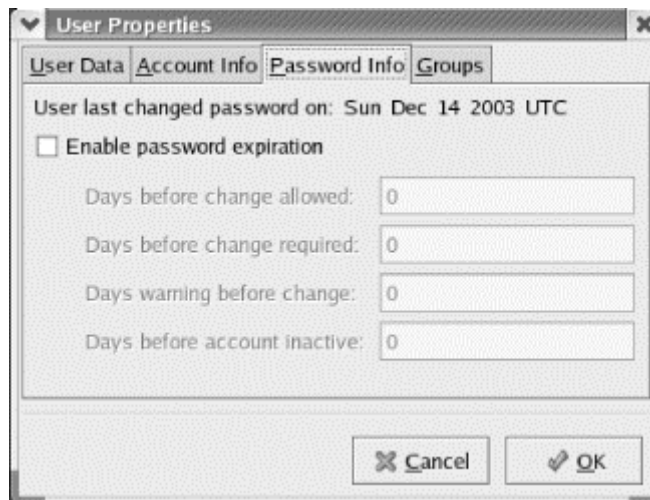
The Account Info pane ([Figure 9-13](#)) lets you specify a date on which a user account becomes unusable. Alternatively, you can use the text box labeled "Local password is locked" to immediately disable an account.

Figure 9-13. The Account Info pane of the User Properties dialog box



The Password Info pane ([Figure 9-14](#)) lets you set password expiration options for the account. To do so, enable the checkbox labeled Enable password expiration.

Figure 9-14. The Password Info pane of the User Properties dialog box



Choosing a Secure Password

A cracker who manages to obtain a copy of your system's `/etc/shadow` file may be able to discover your password by using a utility that tries to determine your password by decrypting the encrypted password stored in the file. You can make the cracker's job more difficult by using one or more of the following techniques:

- Choose a password that is at least six characters long.
- Choose a password that is not a dictionary word. Use a made-up word or a phrase.
- Choose a password that includes uppercase and lowercase letters.
- Choose a password that includes digits as well as letters. However, don't merely follow a dictionary word by the digit 1 or use a similarly guessable

scheme.

- Choose a password that includes one or more special characters, such as a dollar sign, pound sign, or underscore.

Other things being equal, the longer a password is used, the greater the likelihood that it has been compromised. One way to protect users from password crackers is to require users to change their passwords regularly.

Some overly zealous system administrators require users to change their passwords every 30 days. Unless a system contains top secret data, such a short interval is unnecessary, amounting to what's called password fascism. You may find that requiring users to change their passwords every six months or every year is sufficient to avoid hacker invasion of user accounts.

Then you can specify any of the following values:

Days before change allowed

The number of days that must elapse before the user can change the password associated with the user account.

Days before change required

After the specified number of days, the user must change the password associated with the user account.

Days warning before change

This value is used in combination with the Days Before Change Required value. The user will be given advance notice of the need to change the password associated with the user account. The notification begins the specified number of days before the change must be made.

Days before account inactive

After the specified number of days, the user account is disabled.

The Groups pane displays a series of checkboxes corresponding to existing user groups. To associate the user account with a group, check the box corresponding to the desired group.

9.1.5.2 Adding a new user

To create a new user account, click Add User in the Red Hat User Manager dialog box. This will launch the Create New User dialog box ([Figure 9-15](#)).

Figure 9-15. The Create New User dialog box

A screenshot of a 'Create New User' dialog box. It contains several input fields: 'User Name:', 'Full Name:', 'Password:', 'Confirm Password:', and 'Login Shell:' (a dropdown menu showing '/bin/bash'). Below these are three checkboxes: 'Create home directory' (checked), 'Create a private group for the user' (checked), and 'Specify user ID manually' (unchecked). A 'UID:' field with the value '500' is also present. At the bottom are 'Cancel' and 'OK' buttons.

When you create a new user account, you can specify the following values:

User Name

The name of the user account to be created.

Full Name

The name of the user who will use the account.

Password

The password to be associated with the user account.

Confirm Password

Again, the password to be associated with the user account.

Login Shell

The login shell associated with the account. You should generally select `/bin/ bash`.

Create home directory

You should generally check this box, so that a home directory is created.

Home Directory

This value is used only when Create Home Directory is enabled. The default value, which creates a home directory named after the user, is generally acceptable.

Create a private group for this user

You should generally enable this checkbox, which causes automatic creation of a primary user group having the same name as the user account.

Specify user ID manually

Leave this box unchecked unless you want to manually specify the numeric ID associated with the user account. You won't generally need to enable this option. When you've specified the desired values, click OK to create the user account.

9.1.5.3 Deleting a user account

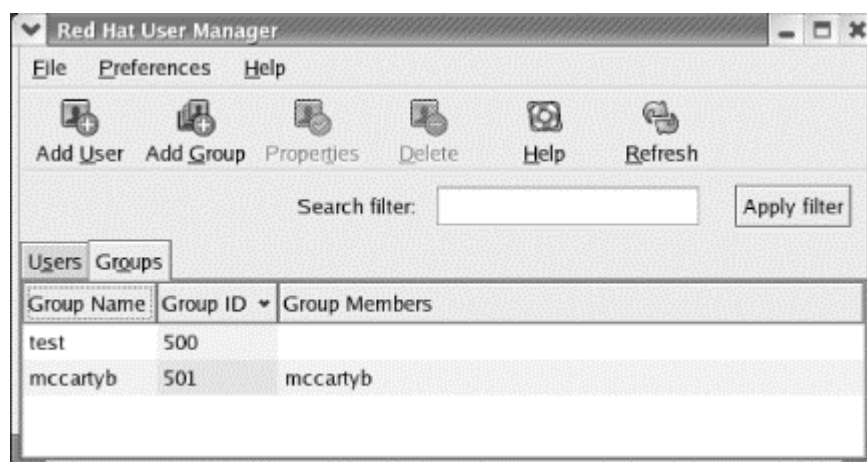
To delete a user account, click on the desired account and click Delete. A message box asks you to confirm your decision to delete the account. The message box also enables you to specify whether the home directory associated with the user account should be deleted.

9.1.5.4 Configuring groups

You may recall from [Chapter 4](#) that Linux uses groups to define a set of related user accounts that can share access to a file or directory. You probably won't find it necessary to configure group definitions very often, particularly if you use your system as a desktop system rather than a server.

To view the configured groups, launch the User Manager tool and click the Groups tab. The Groups pane ([Figure 9-16](#)) appears. Groups are shown in a scrollable list, similar to the way user accounts are displayed.

Figure 9-16. The Groups pane of the User Manager



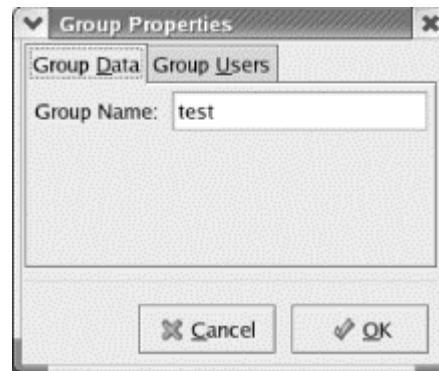
To create a new group, click the Add Group icon. The Create New Group dialog box ([Figure 9-17](#)) appears. The dialog box lets you specify the name of the new group. It also lets you manually specify the numeric ID of the group; however you should rarely need to do so. When you've specified the name of the new group, click OK to create the group.

Figure 9-17. The Create New Group dialog box



To modify a group, click on the name of the group. The Group Properties dialog box ([Figure 9-18](#)) appears. The Group Data pane of the dialog box lets you revise the name of the group. The Group Users pane contains a scrollable list of users; you can associate a user account with a group by enabling the checkbox adjacent to the username, or dissociate a user account from a group by disabling the checkbox adjacent to the username. When you've completed your changes, click OK to make them effective.

Figure 9-18. The Group Properties dialog box



To delete a group, select the group in the Groups pane and click the Delete icon. A message box asks you to confirm your decision to delete the group.

9.1.6 Switching Desktops

During the installation procedure, GNOME is automatically configured as the default desktop. If you installed KDE, you can choose KDE as the default by using the desktop switcher. To do so, select System Settings → More System Settings Desktop → Switching Tool from the main menu. The Desktop Switcher tool ([Figure 9-19](#)) appears.

Figure 9-19. The Desktop Switcher dialog box



In addition to GNOME and KDE, the Desktop Switcher lets you specify TWM (tiny window manager), a sparsely functional but highly efficient desktop. The checkbox labeled "Change only applies to current display" lets you restrict your choice of desktop to the current display (generally *localhost.localdomain:0.0*), as indicated in the top line of the dialog box. This facility is useful if your system has been configured to allow remote users to log in via X.

To specify a desktop, click the desired radio button and click OK. A dialog box appears, informing you that you must restart X for the selected desktop to appear. To restart X, simply log out and then log in.

9.2 Administering Your System by Using the System Tools Menu

The System Tools menu of the GNOME and KDE desktops provides access to a set of tools that help you administer your system. The distinction between these tools and the tools provided by the System Settings menu is rather arbitrary. That is, any given tool could as likely have been placed on one menu rather than the other. Generally, items on the System Tools menu perform an action, whereas items on the System Settings menu let you configure a facility. But, exceptions to this rule of thumb abound.

The tools provided by the System Tools menu are:

Disk Management

Provides access to the User Mount Tool, which lets users mount, unmount, and format filesystems

Floppy Formatter (GNOME only)

Provides access to the Format a Floppy tool, which formats floppies

Hardware Browser

Provides access to the Hardware Browser tool, which displays information about system hardware

Internet Configuration Wizard

Provides access to a wizard that assists you in creating new network connections

Kickstart (Red Hat Enterprise Linux only)

Provides access to the Kickstart Configurator tool, which lets you specify the parameters for automated installation of Red Hat Linux

Network Device Control

Provides access to the Network Device Control tool, which lets you configure and administer network devices and connections

Printing Notification Icon

Provides an alert icon resembling a small printer. The icon appears on the panel when one or more print jobs are pending.

Print Manager

Provides a tool for managing pending print jobs.

Red Hat Network

Provides access to the Red Hat Update Agent, which assists Red Hat Network subscribers in updating their systems

Red Hat Alert Icon

Provides a panel icon that changes shape and color when updates are available for your system.

System Logs

Provides access to the System Logs tool, which lets you view the contents of system log files

System Monitor

Provides access to the System Monitor tool, which lets you view system performance and resource consumption information

Terminal

Provides access to a shell, which lets you enter commands

Traceroute (Fedora Core only)

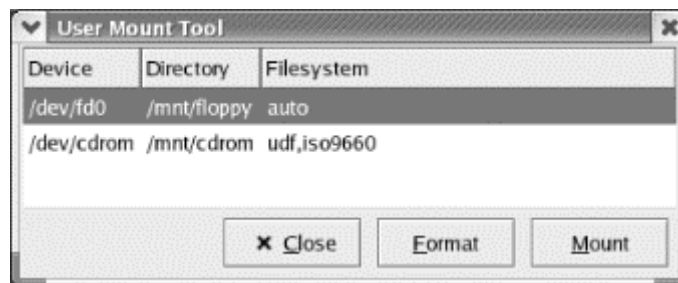
Reports intermediate hosts (hops) between a local system and remote system. The contents of the System Tools menu vary somewhat, depending on the packages you've installed and which desktop (GNOME or KDE) you're using. This section describes several of the tools that appear on the menu. However, it does not explain the Floppy Formatter, since that tool appears only on the GNOME menu and since its functions are also provided by the Disk Manager. Some of the tools are described elsewhere in this book. For instance, [Chapter 8](#) explains the Red Hat Update Agent. The Network Device Control tool concerns network configuration and administration, which are topics considered in [Chapter 10](#) and

[Chapter 11](#). The Kickstart tool is primarily used by enterprise system administrators and is not described in this book. And, I dispense with explanation of the Terminal tool, since it merely launches a familiar terminal window.

9.2.1 Managing Disks

The Disk Management tool ([Figure 9-20](#)) lets you mount and unmount filesystems. It also lets you format filesystems. To launch the tool, choose System Tools → Disk Management from the main menu.

Figure 9-20. The Disk Management tool

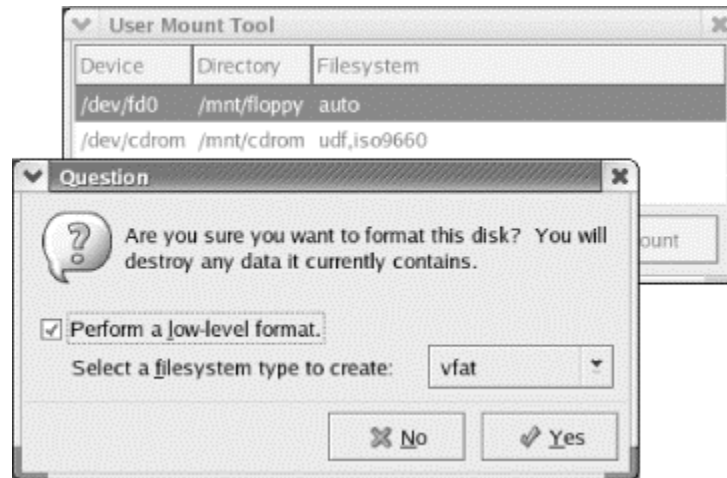


The tool shows a list of configured filesystems, taken from */etc/fstab*. To mount a filesystem, select the filesystem and click Mount. If you mount a floppy or CD-ROM under GNOME, an icon representing the mounted filesystem appears on the desktop. Click the icon to browse the filesystem.

To unmount a mounted filesystem, select the filesystem and click Unmount. The usual restrictions apply concerning the inability to unmount a file that is in use.

To format a filesystem, select the unmounted filesystem and click Format. A dialog box appears, as shown in [Figure 9-21](#). This dialog box lets you choose the type of filesystem to create. For floppies, you may find it convenient to use the **vfat** file system, which can be read by Linux and Windows 9x, ME, 2000, and XP. For other purposes, you should generally use the standard Linux filesystem type, **ext3**. The dialog box also lets you choose whether a low-level format should be performed. Generally, you should perform a low-level format only when formatting a floppy.

Figure 9-21. Formatting a filesystem



9.2.2 Browsing Hardware Devices

The Hardware Browser tool ([Figure 9-22](#)) lets you view information describing the system's hardware devices. To launch the tool, select System Tools → Hardware Browser from the main menu. When it starts up, the tool probes your system and may require a minute or so to complete its work. During that time, the system may become unresponsive.

Figure 9-22. The Hardware Browser tool

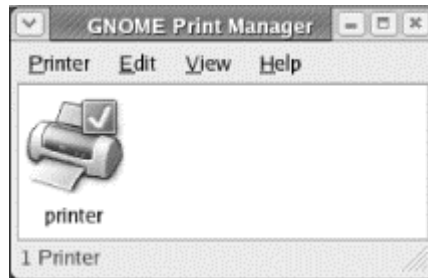


To use the tool, click a device in the list at the left side of the window. A description of the device appears in the right side of the window. The tool does not let you change or configure devices. It merely lets you view information describing them.

9.2.3 Managing Print Jobs

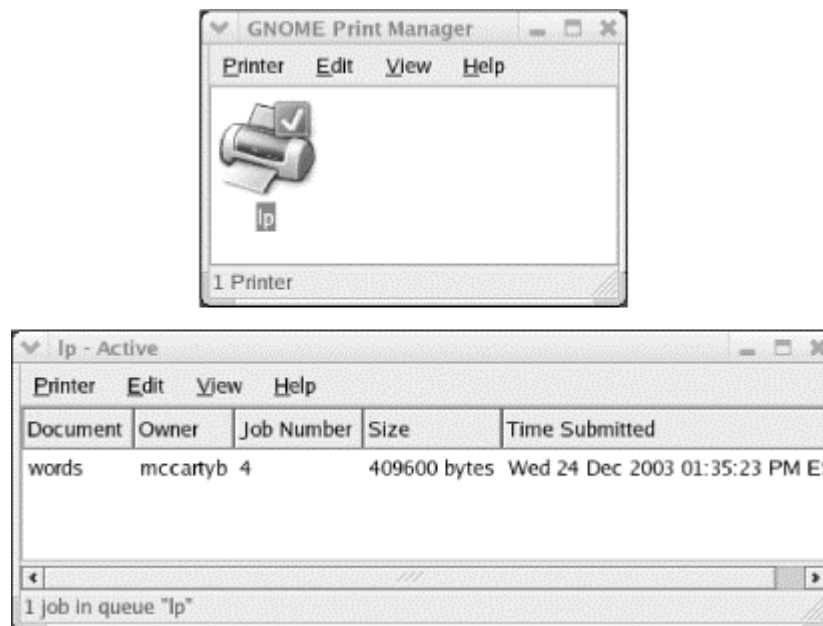
The GNOME Print Manager ([Figure 9-23](#)) lets you view and manage documents queued for printing. To launch the Print Manager, select System Tools → Print Manager from the main menu. An icon is displayed for each configured printer.

Figure 9-23. The GNOME Print Manager



To view the contents of the print queue associated with a printer, simply double-click the printer's icon. The print queue tool ([Figure 9-24](#)) appears. To cancel a print job, select the associated document and click Edit → Cancel Documents. You can cancel multiple documents in a single operation by using the **Ctrl** key to select them and then clicking Edit Cancel Documents.

Figure 9-24. The print queue tool



The Print Manager includes options that let you view only selected printers and jobs. These options are helpful in busy environments. You can learn about these options by clicking Help on the Print Manager menu.

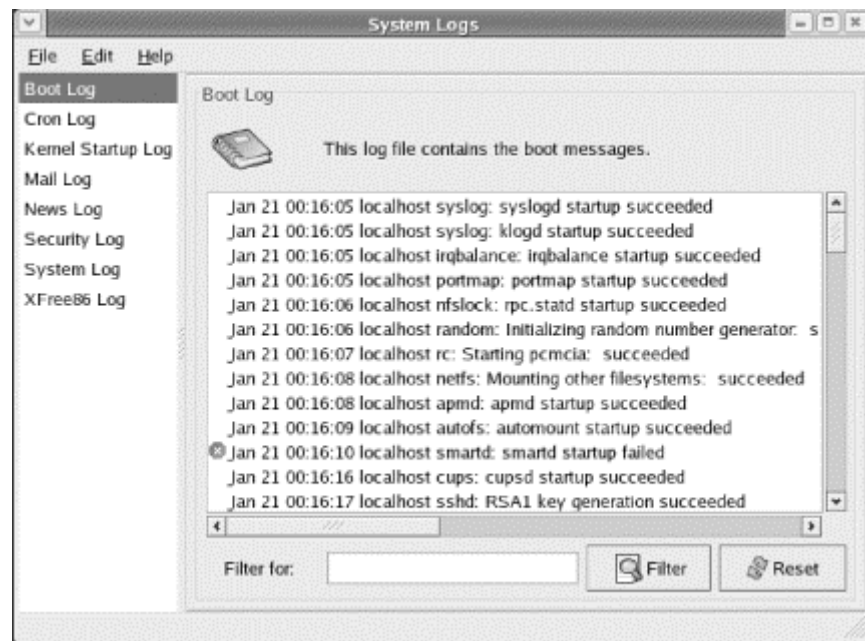
9.2.4 Viewing System Logs

The System Logs tool ([Figure 9-25](#)) lets you view information contained in system log files, which record information about important events. The tool is especially helpful in troubleshooting system problems, because log files may contain relevant entries that

indicate or suggest the cause of the problem.

To launch the tool, choose System Tools → System Logs from the main menu. To view log entries, select the desired log file in the left part of the window. The log entries appear in the right part of the window. To find a particular entry or set of entries within a log file, type text into the textbox labeled "Filter for." The tool will then display only log entries containing the specified text. To disable the filter, click Reset.

Figure 9-25. The System Logs tool



The log files available for viewing depend on which packages are installed. The standard log files are:

Boot Log

Contains information about services started and stopped during system boot up and shutdown

Cron Log

Contains information about the cron service, which runs tasks at scheduled times

Kernel Startup Log

Contains information recorded during system boot up

Mail Log

Contains information about the mail server and mail messages

News Log

Contains information about the USENET news service

Security Log

Contains information about logins and certain service start ups and shutdowns

System Log

Contains general-purpose entries not sent to another log and some important entries also sent elsewhere

Update Agent Log (Red Hat Enterprise Linux only)

Contains entries describing actions performed by the Red Hat Update Agent

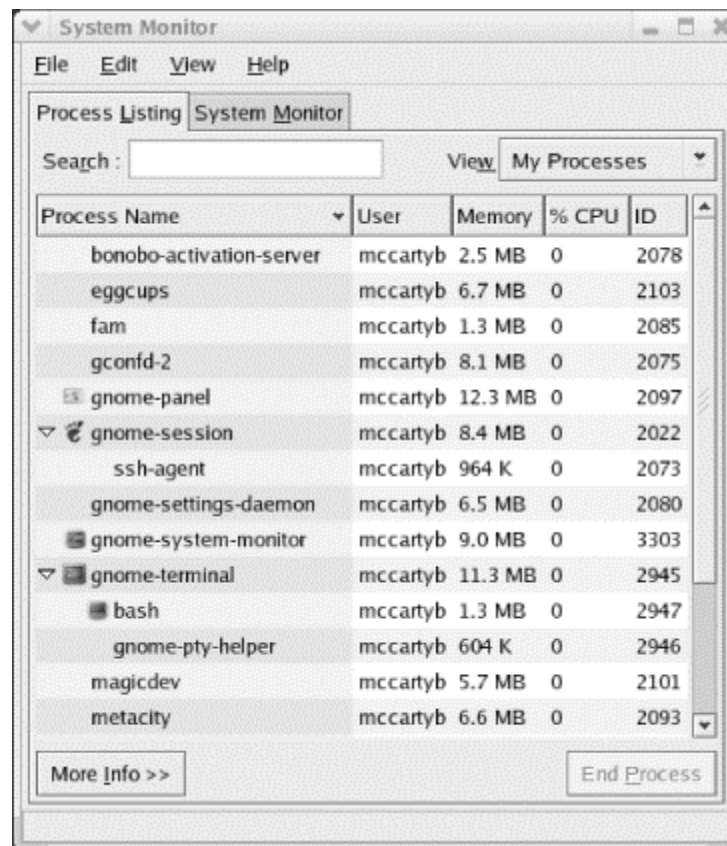
XFree86 Log

Contains entries logged by the X server

9.2.5 Monitoring the System

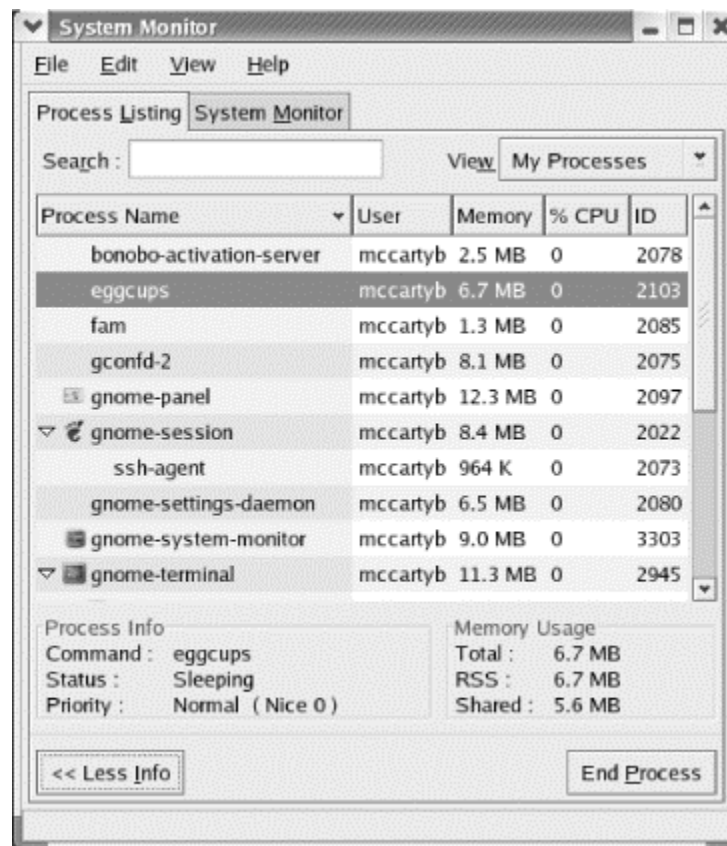
The System Monitor tool ([Figure 9-26](#)) lets you view information about system and user processes (running programs). It also lets you view CPU, memory, and disk space usage data. Essentially, the tool combines the functions performed under Windows by the Task Manager and System Information applets. To launch the tool, choose System Tools → System Monitor from the main menu.

Figure 9-26. The System Monitor tool



The display uses a tree control that displays members of a process group under their parent processes. Click the triangle next to the name of a parent process to expand or collapse the tree. To view extended information about a process, select the process and click More Info. The bottom pane of the tool is changed to show information about the selected process, as shown in [Figure 9-27](#).

Figure 9-27. Extended process information

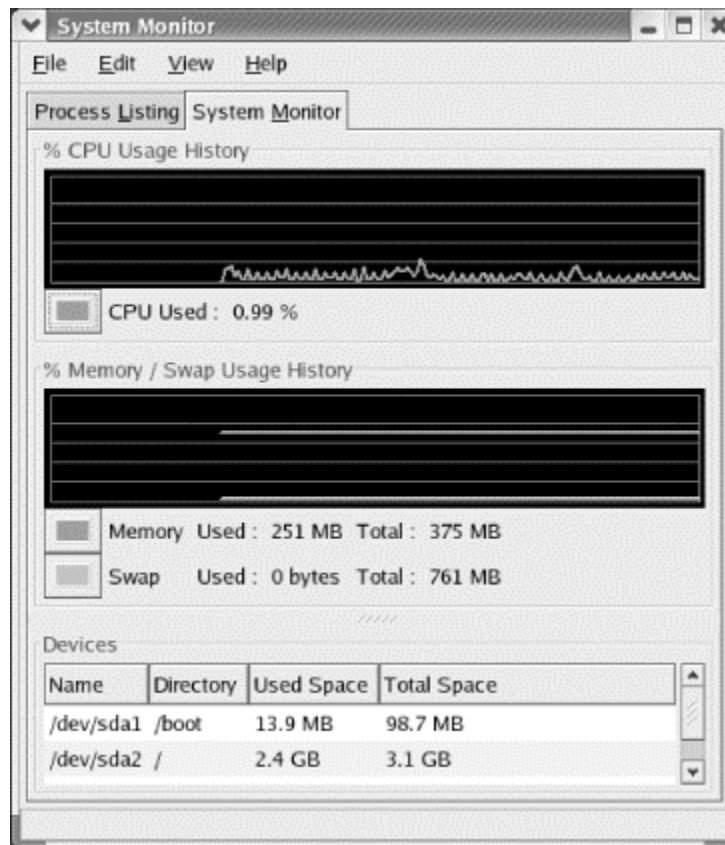


You can terminate a process by selecting it and clicking End Process. However, you should do so judiciously. Terminating a necessary process or terminating a process at an unsuitable time may destabilize your system. Generally, you should terminate only processes that appear to be hung.

If the list of processes is long, you may find it convenient to use the Search box. Or, you can use the View list to view all processes, processes owned by your user account, or active processes.

Clicking the System Monitor tab displays information such as that shown in [Figure 9-28](#). The tab includes graphs that provide a running display of CPU, memory, and swap space usage. In addition, the tab includes a list box that displays the used and total space associated with mounted disk partitions.

Figure 9-28. The System Monitor tab



9.3 Controlling Services by Using the Server Settings Menu

Services are generally processes that wait for a request to perform an operation or provide information and then do so. The System Settings → Server Settings menu provides menu items that let you administer certain services. The menu also provides the Services menu item associated with the Service Configuration tool, which lets you start and stop services and configure which services run at various runlevels. Red Hat Enterprise Linux and Fedora Core support seven runlevels:

0

Runlevel 0 is associated with a system shutdown.

1

Runlevel 1 is associated with the state known as single-user mode. It is generally used to troubleshoot or configure the system from a state that does not provide network services. It vaguely resembles Windows safe mode.

2

Runlevel 2 is associated with a state that provides access to the network but does not provide network services to remote hosts. Like runlevel 1, it's primarily used for troubleshooting and configuration.

3

Runlevel 3 is a normal system operating mode. It features a text-based login prompt.

4

Runlevel 4 is reserved; that is, it is not used.

5

Runlevel 5 is a normal system operating mode. It features a graphical login screen.

6

Runlevel 6 is associated with a system reboot.

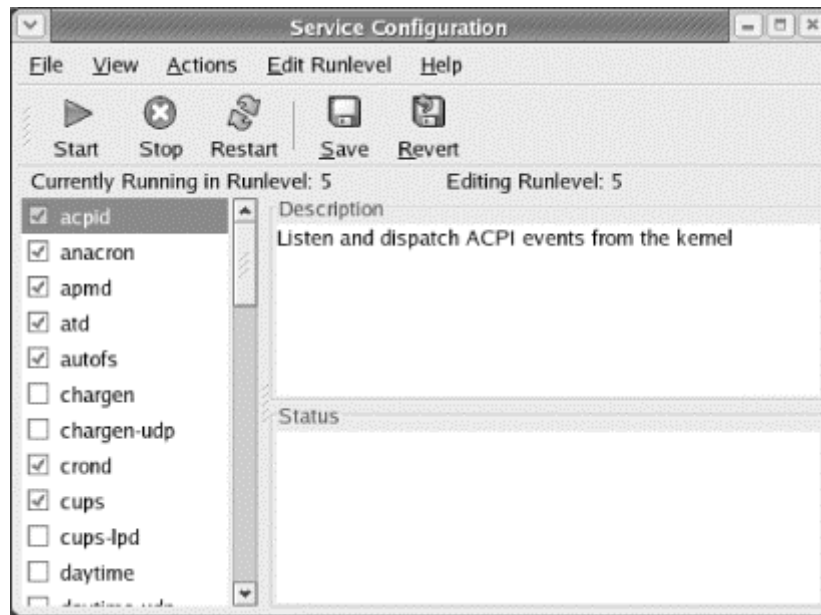
Runlevels 3 and 5 are normal system operating modes. Other modes are used transiently (levels 0, 6), for troubleshooting (levels 1, 2), or not used at all (level 4). When you associate a service with a runlevel, the service is automatically started whenever the runlevel is entered. You can set the current runlevel by issuing the command:

```
init n
```

where *n* is the number of the desired runlevel. However, it's seldom necessary to change the runlevel of a running system. You should do so judiciously, since services may be abruptly terminated during transition from one runlevel to another.

To launch the Service Configuration tool, select System Settings → Server Settings → Services from the main menu. [Figure 9-29](#) shows the appearance of the tool. Note that it reports the current runlevel. It also reports the runlevel to which changes apply, called the *editing runlevel*.

Figure 9-29. The Service Configuration tool



To learn about the function of a service, select the service's name. A description of the service may appear in the top right pane of the Service Configuration window. [Figure 9-29](#) features such a description, but some services may not.

9.3.1 Associating Services with Runlevels

To associate a service with a runlevel, select the runlevel by using the Edit Runlevel menu. You can edit only for runlevels 3, 4, and 5. Despite the fact that runlevel 4 is not used, it's common to associate with level 4 any services associated with level 3. However, you need not do so.

Once you've selected the desired runlevel, you can specify the services that should run when that runlevel is active. To specify that a service should run, enable the associated checkbox. To specify that a service should not run, disable the associated checkbox.

Your changes to a runlevel are not automatically saved. To save them, click the Save button on the toolbar of the Service Configuration tool.

9.3.2 Starting and Stopping Services

Suppose that the current runlevel is 3 and that a particular service is running. If you use the Service Configuration tool to specify that the service should no longer run in runlevel 3, the system does not immediately terminate the service. Similarly, the system does not immediately start a service newly specified to run in the current runlevel. Instead, the status of the service is changed the next time the specified runlevel is entered.

However, you can use the Service Configuration to immediately start or stop a service. To do so, select the service and click the Start or Start button on the Service Configuration toolbar. You can also restart a service, which stops and then restarts the service. Restarting a service is useful when you've manually modified the service's configuration file, since restarting the service causes it to re-read the configuration file.

9.3.3 Hardening Your System

Many attackers target services. Therefore, a useful step in hardening a system—that is, protecting it against attack—is disabling or removing unneeded services. To disable a service, simply ensure that the service is not associated with runlevels 3 or 5. However, it's better yet to remove the service from the system. That way, an attacker will find it more difficult to enable the service.

Of course, you must not disable or remove a service that's needed on the system. So, you need to know something about a system before you tamper with it. You can discover the services for which your system is configured by examining the files in */etc/init.d*. Each service is represented by a file in that directory.

To learn about a service, first determine the name of the associated RPM package. To do so, issue the command:

```
rpm -qf /etc/init.d/ service
```

where *service* is the name of a service. For example, issuing the command:

```
rpm -qf /etc/init.d/apmd
```

tells you that the apmd service is associated with the apmd package. Often, as in this case, the name of service and the name of the associated package are the same. However, this isn't always true.

Once you know the name of the associated package, you can learn about the service by issuing the command:

```
rpm -qi package
```

where *package* is the name of the package.

To disable a service, you can use the Service Configuration tool. To remove a service, you can issue the command:

```
rpm -e package
```

where *package* is the name of the package. Occasionally, it's not possible to remove a package because another package requires it. In such a case, you can remove both packages or settle for merely disabling the unremovable package.

[Table 9-1](#) identifies several services that are seldom needed that are nevertheless installed by default in some installation configurations.

Table 9-1. Commonly unused default servers	
Service	When needed
irda	Used with infrared devices and peripherals
isdn	Used for ISDN connections
kdcrotate	Used with Kerberos authentication
nfs	Used with NSF (Network File Sharing)
nfslock	Used with NSF
pcmcia	Used with PCMCIA hardware, such as that common on laptops

Table 9-1. Commonly unused default servers	
Service	When needed
portmap	Used with NSF and some other facilities
postfix	An alternative mail server that can be used in place of sendmail
snmpd	Used with SNMP (Simple Network Management Protocol)
snmptrapd	Used with SNMP
winbind	Used with Windows file sharing
xinetd	Used to host other servers, such as <code>sgi_fam</code>
ypbind	Used with NIS (Network Information Services)

Chapter 10. Connecting to the Internet

This chapter explains how to use Red Hat Enterprise Linux or Fedora Core to connect to the Internet via a telephone dialup, Integrated Services Digital Network (ISDN), Digital Subscriber Line (DSL), Ethernet, or wireless connection. First, it provides an overview of networking and explains how to use the Network Administration Tool, an X-based program that makes it easy to connect to the Internet via an Internet service provider (ISP). The chapter also explains how to use **wvdial**, a program that can establish a telephone dialup connection to the Internet but doesn't require X. Next, the chapter describes several popular network client applications available under Linux, including the Mozilla web browser, the Ximian Evolution email client, and a graphical FTP client. Finally, the chapter gives some tips on how to configure Linux to work with your cable or DSL modem if the Network Administration Tool is unable to do so.

10.1 Networking Overview

Computers handle network traffic much as the post office handles mail. Think, for example, of the steps involved in sending and receiving a letter. Your postal carrier must know where to drop off and where to pick up mail. So your home must have some kind of recognizable interface; we call this a mailbox. And whereas your postal carrier may know your neighborhood quite well, delivery in other areas requires other carriers. Mail is passed to these other carriers through a gateway; we call this the post office. Although you can think of the whole postal system as one big network, it's easier to understand if you think of it as a hierarchy of subnetworks (or subnets): the postal system is divided into states, states are

divided into counties and cities with a range of Zip Codes, Zip Codes contain a number of streets, and each street contains a unique set of addresses.

Computer networking mirrors this model. Let's trace an email message from you to a coworker. You compose the message and click Send. Your computer passes the message to a network interface. This interface may be a modem by which you dial up an Internet service provider (ISP), or it may be via an Ethernet connection on a LAN. Either way, on the other side of the interface is a gateway machine. The gateway knows how to look at the address of the recipient of the email message and interpret that message in terms of networks and subnets. Using this information, the gateway passes the message to other gateways until the message reaches the gateway for the destination machine. That gateway in turn delivers the message via a recognizable interface (such as a modem or Ethernet link) to the recipient's inbox.

If you review this analogy, you can easily see which parts of networking you'll need to configure on your Linux system. You'll need to know the address of your machine. Just as the town name Sebastopol and the Zip Code 95472 are two different names for the same location, you may have both a name, called a hostname, and a number, called an IP number or IP address, that serve as the address for your machine.

To translate between these two notations, you may need to know the address of a Domain Name Server (DNS). This is a machine that matches IP addresses with hostnames. You'll also need to know the address of a gateway machine through which network traffic will be routed. Finally, you'll need to be able to bring up a network interface on your system, and you'll need to assign a route from that interface to the gateway.

While all of this can seem complex, it really isn't any more complex than the postal system, and it functions in much the same way. Fortunately, Linux comes with tools to help you automate network configuration.

10.2 Configuring an Internet Connection

The Network Administration Tool simplifies configuration of your system to access the Internet via a telephone dialup, ISDN, DSL, Ethernet, or wireless connection. The Network Administration Tool requires you to follow a three-step process:

1. Set up the hardware device associated with the connection.
2. Specify DNS settings and hostnames.
3. Activate the device, if necessary.

The following subsections explain how to perform these steps.



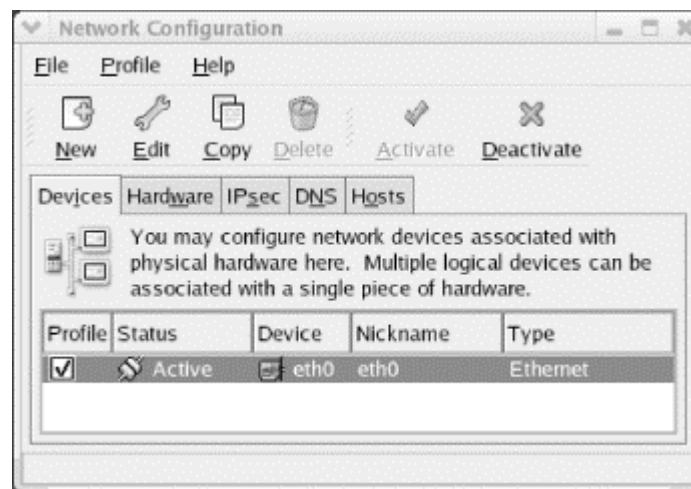
The Network Administration tool supports all these methods of connecting to the Internet. However, some hardware devices are not compatible with Red Hat Enterprise Linux and Fedora Core. And, some Internet service providers insist that their customers use only Windows. In either case, you may experience difficulties in connecting to the Internet. The final section of this chapter explains some means of last resort that might help you resolve problems.

10.2.1 Setting Up Hardware Devices

In the past, most computer users connected to the Internet via a plain old telephone service (POTS) dial-up modem. However, today, many means of connecting to the Internet are available. For example, many home computer users have high-speed connections using ISDN or DSL. Corporate computer users often connect to the Internet via a local area network, using an Ethernet adapter. And wireless network adapters are becoming quite popular.

To set up a hardware device using the Network Administration Tool, choose System Settings Network from the main menu. The Network Administration Tool appears, as shown in [Figure 10-1](#). Although the screen is titled "Network Configuration," the tool is nevertheless commonly known as the Network Administration Tool.

Figure 10-1. The Network Administration Tool



The Network Administration Tool has five tabs:

Devices

Used to associate a physical device with a network connection

Hardware

Used to set up a physical device

IPsec

Used to set up a virtual private network (VPN); to use this option you must first install the *cipe* package

DNS

Used to specify DNS servers and related options

Hosts

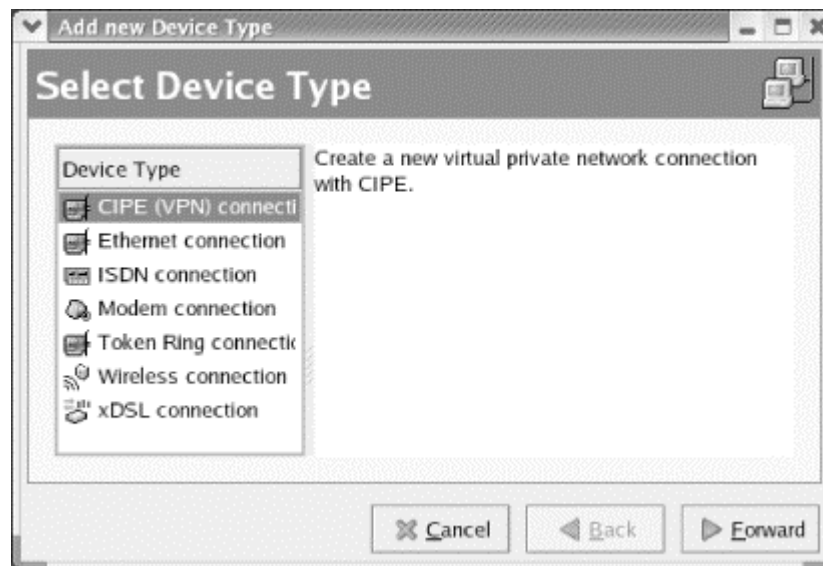
Used to specify names of hosts not known to a DNS server

To begin setting up an Internet connection, click the Devices tab and then click the New button in the toolbar. A dialog box appears, inviting you to choose the type of your device, as shown in [Figure 10-2](#). Choose the appropriate device type and click Forward. Then, follow the steps in the following subsection appropriate to the type of your device.



Red Hat Enterprise Linux and Fedora Core support two schemes used by DSL providers: Point-to-Point Protocol over Ethernet (PPPoE) and Dynamic Host Control Protocol (DHCP). If your DSL provider uses DHCP, you should configure your DSL connection as though it were an Ethernet connection. Otherwise, you should specify xDSL as the device type.

Figure 10-2. The Add new Device Type dialog box

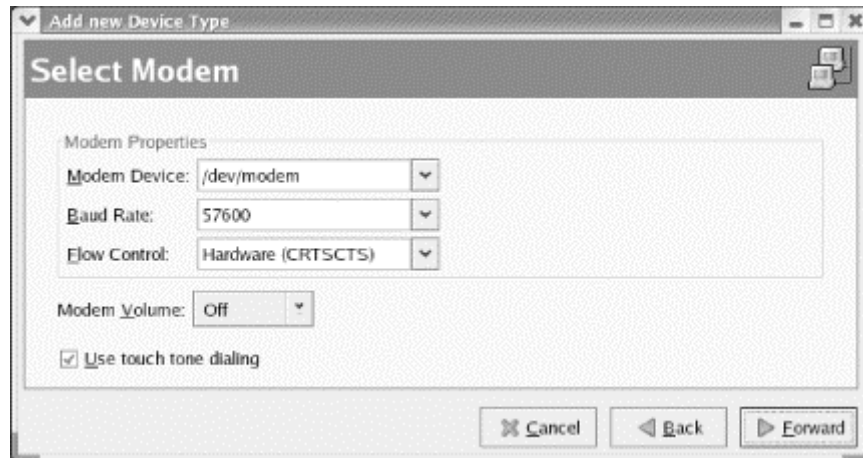


10.2.1.1 Setting up a dialup modem

When you specify a Modem as the device type, the Network Administration Tool probes for your modem. This process may take several seconds. Once the Network Administration

Tool locates the modem, it displays the Select Modem dialog box, shown in [Figure 10-3](#), which lets you specify modem characteristics. Generally, the defaults are acceptable. However, check the documentation for your modem to be sure. If your phone line does not support touch tone dialing, de-select the "Use touch tone dialing" checkbox. Then, click Forward to continue.

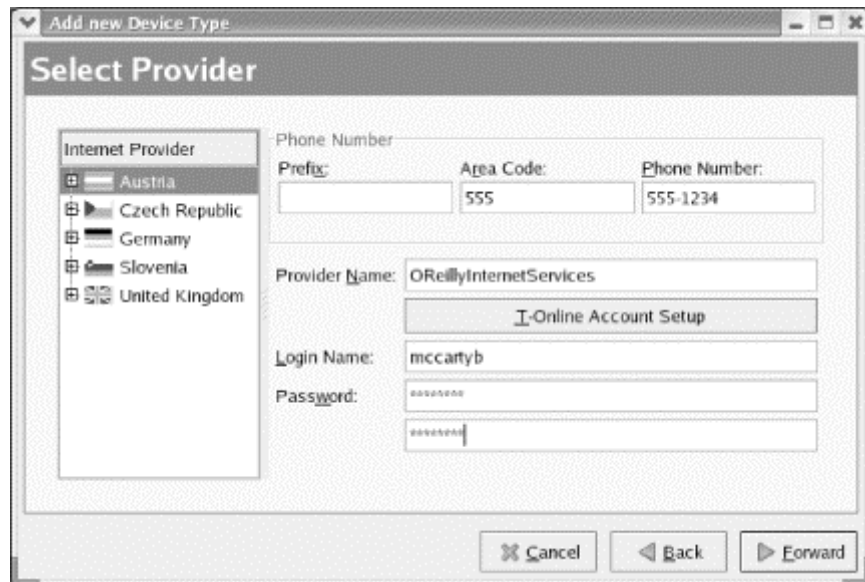
Figure 10-3. The Select Modem dialog box



If the Network Administration Tool is unable to successfully probe your modem, your modem may not be compatible with Linux. Many laptops contain incompatible modems. Some modems—so-called WinModems—are compatible only with Windows. Other modems—so-called LinModems—are supported by their manufacturer only for Windows, but have available Linux drivers. To learn more about your modem and its compatibility with Linux, see <http://www.linmodems.org> or Rob Clark's *WinModems Are Not Modems* page, <http://freewebhosting.hostdepartment.com/g/gromitkc/winmodem.html>. If your modem is not compatible with Linux, you can often work around the problem by using an external modem. Because external modems connect via the serial port, essentially every serial modem designed for PC use is compatible with Linux.

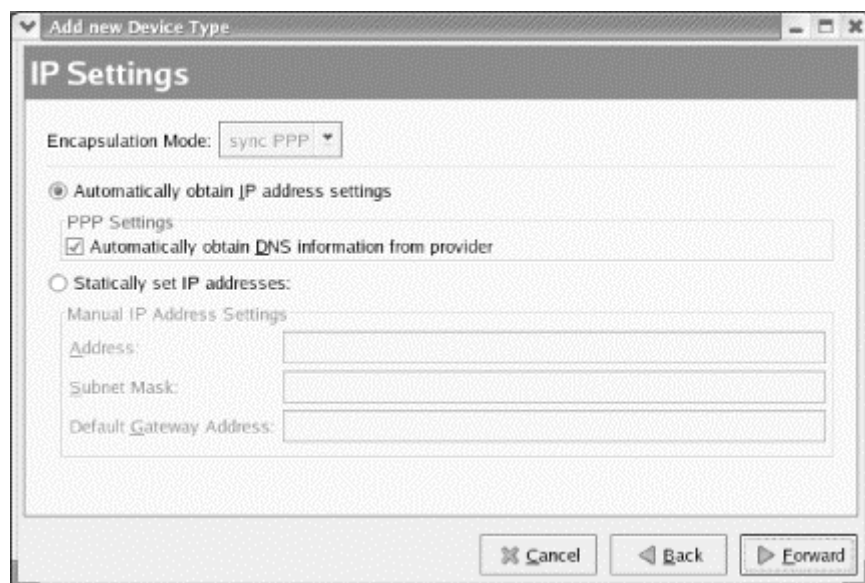
The Select Provider dialog box appears, as shown in [Figure 10-4](#). If your country and provider are listed, select them. Otherwise, specify the phone number, name, login name, and password associated with your account. As usual, you must enter the password twice, to reduce the chance of an undetected typographical error. Then click Forward to continue.

Figure 10-4. The Select Provider dialog box



The IP Settings dialog box appears, as shown in [Figure 10-5](#). If your ISP dynamically assigns your system a network configuration when you log in, accept the default setting, "Automatically obtain IP address settings." This setting is correct for most users. However, if your ISP provides your system a static network configuration, enable the radiobutton labeled "Statically set IP addresses," and specify the IP address, subnet mask, and default gateway address assigned by your ISP. Click Forward to continue.

Figure 10-5. The IP Settings dialog box



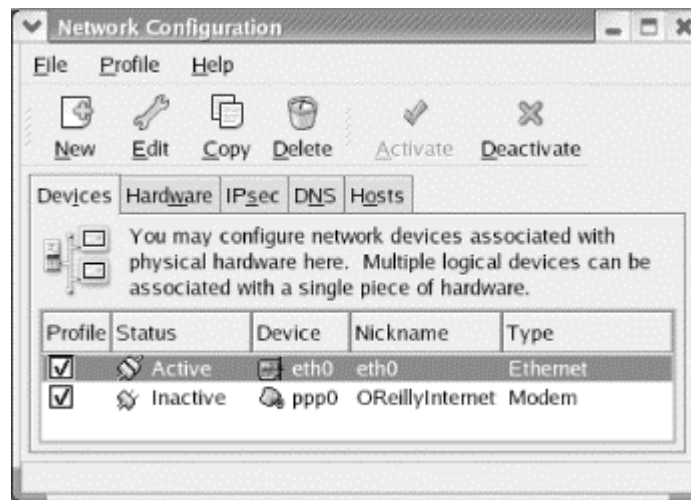
The Create Dialup Connection dialog box appears, as shown in [Figure 10-6](#). This dialog box summarizes information about your dialup connection. If the information is correct, click Apply to proceed; otherwise use the Back button to return to the appropriate dialog box and correct the mistake.

Figure 10-6. The Create Dialup Connection dialog box



After you click Apply, the main Tool screen reappears. However, this time the screen includes a line identifying your modem as a ppp (Point-to-Point Protocol) device, as shown in [Figure 10-7](#).

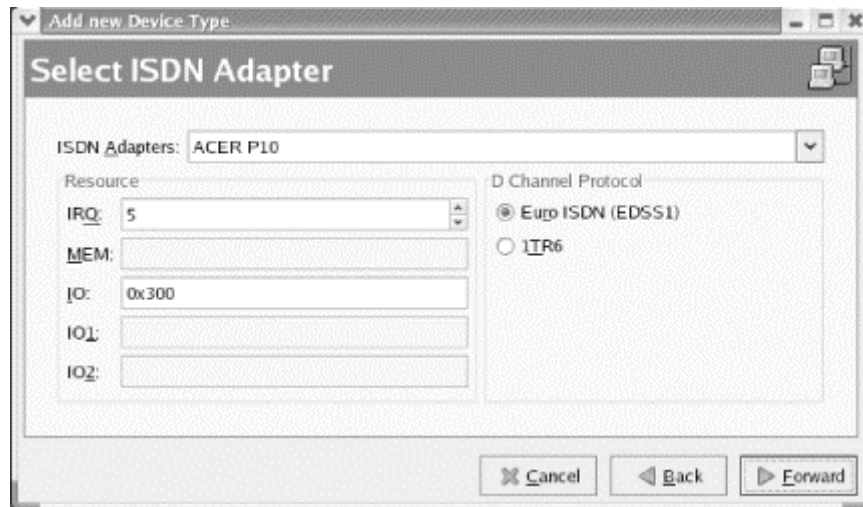
Figure 10-7. The Network Configuration tool



10.2.1.2 Setting up an ISDN modem

When you specify an ISDN modem as the device type, the Network Administration Tool presents a list of supported ISDN modems, as shown in [Figure 10-8](#). Specify the device characteristics and choose the D Channel Protocol used by your ISDN provider. Click Forward to continue.

Figure 10-8. The Select ISDN Adapter dialog box



The Select Provider dialog box appears, shown earlier in [Figure 10-4](#). If your country and provider are listed, select them. Otherwise, specify the phone number, name, login name, and password associated with your account. Then, click Forward to continue. The IP Settings dialog box appears, as shown in [Figure 10-5](#). If your ISP dynamically assigns your system a network configuration when you log in, accept the default setting, Automatically obtain IP address settings. This setting is correct for most users. However, if your ISP provides your system a static network configuration, enable the radiobutton labeled Statically set IP addresses, and specify the IP address, subnet mask, and default gateway address assigned by your ISP. Click Forward to continue.

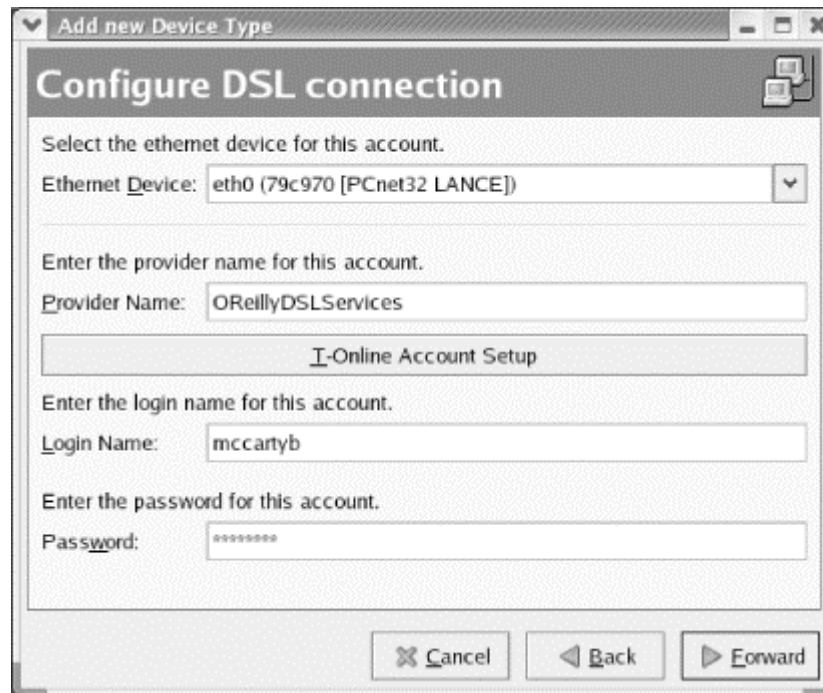
The Create Dialup Connection dialog box appears, as earlier shown in [Figure 10-6](#). This dialog box summarizes information about your dialup connection. If the information is correct, click Apply to proceed; otherwise use the Back button to return to the appropriate dialog box and correct the mistake.

After you click Apply, the main Tool screen reappears. However, this time the screen includes a line identifying your ISDN modem as an ippm (ISDN Point-to-Point protocol) device.

10.2.1.3 Setting up an xDSL modem

Several varieties of DSL are in use, including IDSL, ADSL, and SDSL. The xDSL device type supports each variety. When you specify DSL modem, the Network Administration Tool presents the Configure DSL connection dialog box, as shown in [Figure 10-9](#). Specify the Ethernet device associated with your DSL link and the login name and password associated with your DSL account. Oddly, the dialog box does not ask you to enter your password a second time; so, be careful to enter the password correctly. Click Forward to continue.

Figure 10-9. The Configure DSL connection dialog box



The Create DSL connection dialog box appears. This dialog box summarizes information about your dialup connection. If the information is correct, click Apply to proceed; otherwise use the Back button to return to the appropriate dialog box and correct the mistake. Then, click Forward.

The main Tool screen reappears. However, this time the screen includes a line identifying your DSL modem as a ppp (Point-to-Point Protocol) device.

10.2.1.4 Setting up an Ethernet adapter

When you select Ethernet adapter as the device type, the Network Administration Tool probes your system for supported Ethernet adapters and displays a list of the adapters it finds, as shown in [Figure 10-10](#). Select the adapter you want to configure and click Forward.



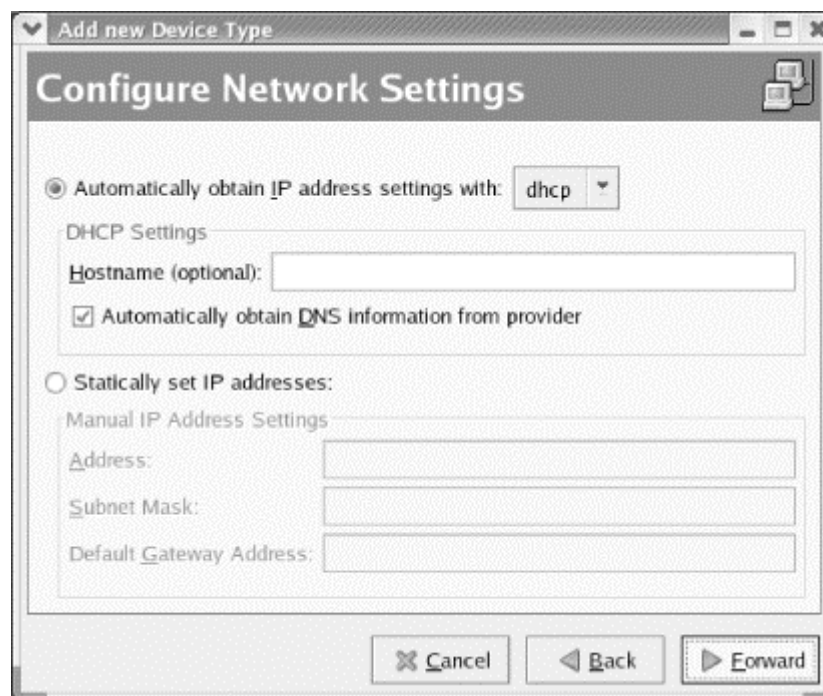
The installation program may have found and automatically configured your system's Ethernet adapter. However, the default configuration may not be appropriate. To check the configuration, highlight the Ethernet adapter and click Edit. The three-tabbed dialog box that appears contains controls similar to those appearing in the wizard sequence used to configure a new Ethernet device. Review and, as necessary, change the specified configuration. Click OK to save your changes.

Figure 10-10. The Select Ethernet Device dialog box



The Configure Network Settings dialog box appears, as shown in [Figure 10-11](#). If your ISP provides a DHCP server that supplies your system with its network configuration, enable the "Automatically obtain IP address settings with" radiobutton and choose DHCP from the drop-down list. Also, enable the "Automatically obtain DNS information from provider" checkbox. If your provider uses BOOTP, which is now rare, choose BOOTP from the drop-down list.

Figure 10-11. The Configure Network Settings dialog box



If your provider provides your system with a fixed network configuration, enable the Statically set IP address radiobutton. Then, specify the IP address, subnet mask, and default gateway address as directed by your network administrator. Click Forward to continue. The Create Ethernet Device dialog box appears. This dialog box summarizes information about your Ethernet connection. If the information is correct, click Apply to proceed; otherwise use the Back button to return to the appropriate dialog box and correct the mistake.

After you click Apply, the main Tool screen reappears. However, this time the screen includes a line identifying your Ethernet adapter as an *eth* device.

10.2.1.5 Setting up a wireless adapter

When you select Wireless adapter as the device type, the Network Administration Tool probes your system for supported wireless adapters and displays a list of the adapters it finds. Select the appropriate adapter and click Forward. If the adapter is not specifically identified, choose Other Wireless Card and click Forward. The Select Ethernet Adapter dialog box (Figure 10-12) appears. Choose the appropriate adapter and specify its characteristics. Click Forward to continue.

Figure 10-12. The Select Ethernet Adapter dialog box

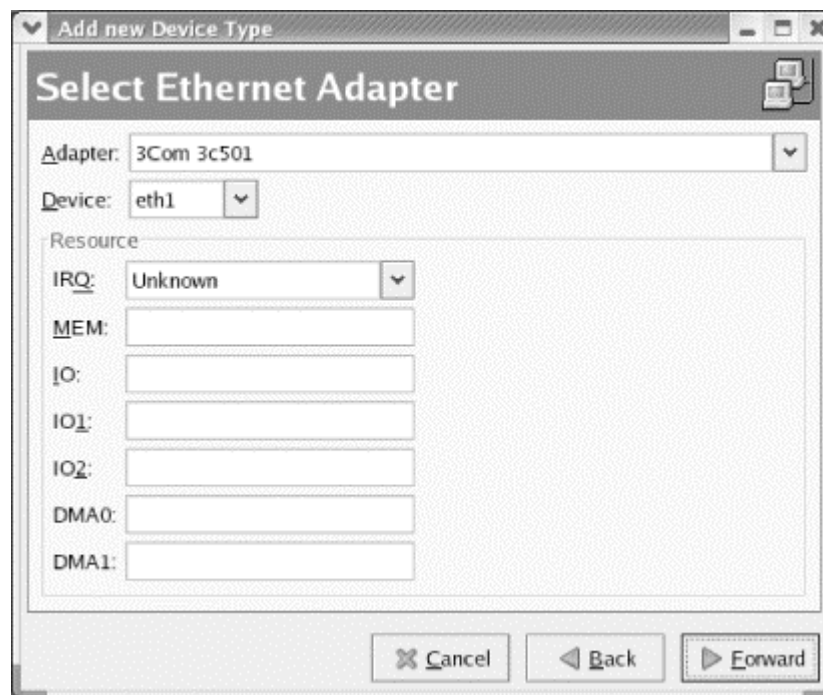
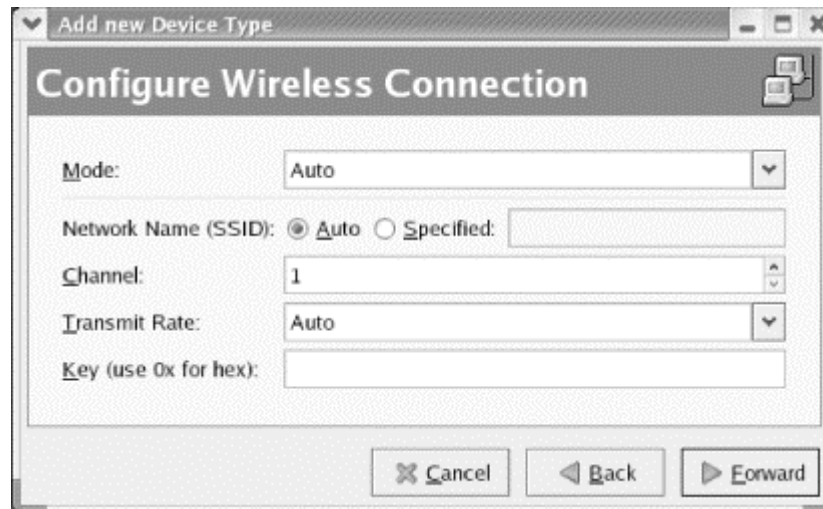


Figure 10-13. The Configure Wireless Connection dialog box



The Configure Wireless Connection dialog box ([Figure 10-13](#)) appears. Specify the mode (Managed or Ad Hoc) in which your wireless point operates or specify Auto to configure the adapter to use whatever mode the access point uses.



You should not generally specify Ad Hoc or Auto if your computer is mobile or may otherwise be in range of wireless networks you do not control or trust. The Ad Hoc or Auto setting may cause your computer to connect to a hostile access point used by an attacker who intends to compromise your system or its communications.

Specify the SSID (Service Set Identifier) of your network or specify Auto if you want to connect to any available network. Then, specify the channel, transmit rate, and key (if any) associated with your network. Click Forward to continue. The Configure Network Settings dialog box appears, as earlier shown in [Figure 10-11](#). If your wireless network provides a DHCP server that supplies your system with its network configuration, enable the "Automatically obtain IP address settings with" radiobutton and choose DHCP from the drop-down list. Also, enable the "Automatically obtain DNS information from provide" checkbox. If your wireless network uses BOOTP, which is now rare, choose BOOTP from the drop-down list. Click Forward. The Create Wireless Device dialog box appears. This dialog box summarizes information about your Ethernet connection. If the information is correct, click Apply to proceed; otherwise use the Back button to return to the appropriate dialog box and correct the mistake.

After you click Apply, the main Tool screen reappears. However, this time the screen includes a line identifying your Ethernet adapter as an eth device.

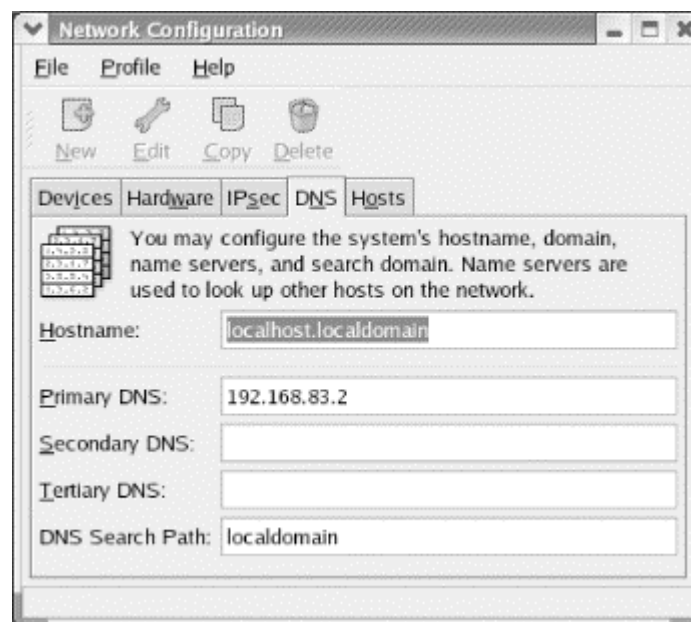


Many users install access points that operate using the default SSID set by the manufacturer and no key. Such users' networks are vulnerable to attack by passersby. Therefore, you should change the SSID associate with your wireless access point before placing it in service. You should also generally specify an encryption key, even though the security provided by standard wireless encryption protocols is imperfect. For more information, see <http://www.wardriving.com>.

10.2.2 Specifying DNS Settings and Hostnames

The DNS tab of the Network Configuration Tool ([Figure 10-14](#)) lets you configure DNS. However, if your ISP provides DNS information via DHCP, you don't need to do so.

Figure 10-14. The DNS tab of the Network Configuration Tool



Internet hosts are generally known by both an IP address and hostname. DNS translates hostnames to IP addresses and IP addresses to hostnames. Translating a hostname to an IP address is called *hostname lookup* or *address resolution*. Translating an IP address to a hostname is called *reverse lookup*. DNS is important, but not essential. For instance, without DNS, you'd have to type IP addresses rather than hostnames when browsing the Web. Doing so would be inconvenient, but workable. However, you wouldn't be able to simply click hyperlinks specified using hostnames. Instead, you'd have to somehow figure out and type the proper IP address.

Using the DNS tab, you can specify the hostname of your system, and the IP addresses of primary, secondary, and tertiary DNS servers. You can also specify one or more domain names that are automatically added to hostnames when performing hostname lookups. For

example, if you frequently access hosts in the *example.com* domain, you can include *example.com* on the search path. Then, you can refer to the host *www.example.com* as simply *www*.

10.2.3 Activating Devices

The Network Administration Tool activates some network devices, such as Ethernet adapters, automatically. Other network devices, such as dial-up modems, must be manually activated and deactivated. To activate a device, select its name in the Devices tab of the Network Administration Tool and click Activate. If the device is a dial-up modem, it will attempt to connect to your ISP.

Once a connection is established, you should be able to access the Internet. Try to ping an Internet host by issuing a command such as **ping www.redhat.com** in a terminal window. You should see replies from the host:

```
PING www.redhat.com (66.187.232.50) from 10.0.0.2 : 56(84) bytes of data.  
64 bytes from redhat.com (66.187.232.50): icmp_seq=0 ttl=239 time=94.383  
msec  
64 bytes from redhat.com (66.187.232.50): icmp_seq=1 ttl=239 time=92.060  
msec  
64 bytes from redhat.com (66.187.232.50): icmp_seq=2 ttl=239 time=91.961  
msec  
--- www.redhat.com ping statistics ---  
3 packets transmitted, 3 packets received, 0% packet loss  
round-trip min/avg/max/mdev = 91.961/92.801/94.383/1.146 ms
```

Terminate the **ping** command by typing **Ctrl-C**.

If the command doesn't work, perhaps your connection isn't working. Or perhaps your ISP's DHCP server failed to properly provide DNS information. Try pinging the IP address of a host you know to be available. For example, issue a command such as:

```
$ ping -n 66.187.232.50
```

If pinging the IP address works, simply use the DNS tab to revise your DNS configuration and you're set. Otherwise, you may have some difficulty getting the connection to work.

Use the **ifconfig** and **route** commands to view your network configuration. If you can figure out the problem, again you're set. If not, you may be able to obtain help from your ISP or from participants in an Internet newsgroup, such as *comp.os.linux.redhat* or *comp.os.linux.misc*.

To terminate a connection, deactivate the associated device by clicking Deactivate on the Devices tab.



After setting up your hardware device, you can edit its characteristics by highlighting the line identifying it and clicking Edit.

If you use your computer in multiple locations, you can use profiles to establish multiple network connections from which you can conveniently choose. For information on doing so, see section 19.12 of *The Red Hat Enterprise Linux 3 System Administration Guide*, at <http://www.redhat.com/docs/manuals/enterprise/RHEL-3-Manual/sysadmin-guide/>.

10.3 The Mozilla Web Browser

Once you've established a connection to the Internet, you can surf the Web using Mozilla, the default Red Hat Enterprise Linux and Fedora Core web browser. To launch Mozilla, choose Internet → Web Browser from the main menu. Mozilla resembles its closed source ancestor, Netscape Navigator. So, if you've used Navigator, you'll feel at home in Mozilla.

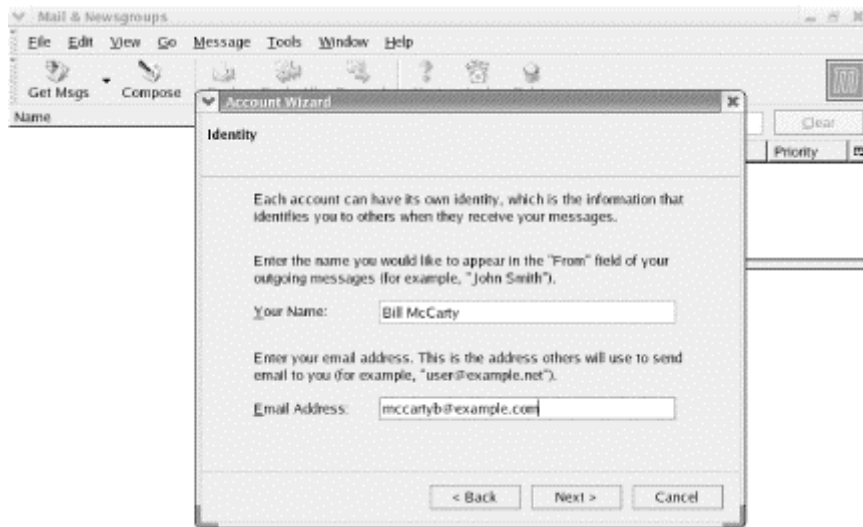


The Web contains quite a bit of malicious software. You should log in as an ordinary user—that is, a user other than the root user—to surf the Web. That way, malicious software is less likely to be able to compromise your system.

When you launch Mozilla for the first time, you may see a dialog box explaining that a Mozilla profile is being created from existing Netscape 4 files. Click Convert Profile to allow Mozilla to create the profile.

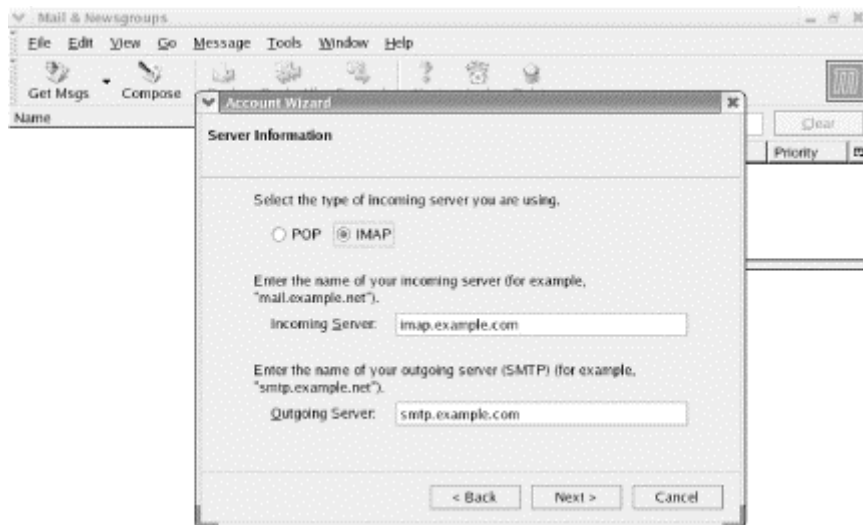
Mozilla includes email and news clients that are easily configured. It also includes a web page composer and address book. To configure Mozilla email, choose Window → Mail & Newsgroups from the Mozilla menu. An Account wizard appears, asking whether you want to configure an email or newsgroup account. Select Email Account and click Next. The Identity dialog box ([Figure 10-15](#)) appears. Specify your name and email address and click Next.

Figure 10-15. The Identity dialog box



The Server Information dialog box ([Figure 10-16](#)) appears. Specify which protocol your mail server uses, POP (Post Office Protocol) or IMAP (Interim Mail Access Protocol). POP servers require you to download email. IMAP servers let you read email that resides on the server. Most up-to-date ISPs support IMAP, the newer protocol. Also specify the hostnames of your incoming and outgoing mail services. A single host may fill both roles. If you don't know this information, you can obtain it from your ISP. Click Next to continue.

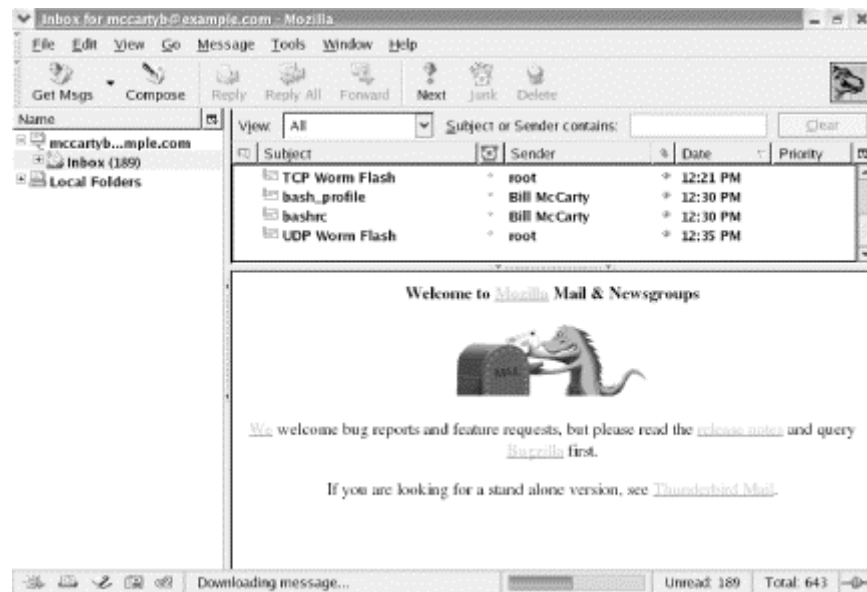
Figure 10-16. The Server Information dialog box



The User Name dialog box appears. Specify the username by which you login to your ISP's mail server. Click Next to continue. The Account Name dialog box appears. Specify a name by which to refer to your new email account so that you can distinguish it from other accounts you may create. Click Next to continue. The Congratulations! dialog box appears. Verify that the information it presents is accurate. If necessary, click Back to return to an earlier dialog box and correct the information. When you're satisfied that the information is correct, click Finish to create your profile.

Mozilla asks your password so that it can login to your email account. Specify the password. If you want Mozilla to remember your password, enable the checkbox Use Password Manager to remember this password. Click OK to login. Mozilla reads and displays the contents of your Inbox, as shown in [Figure 10-17](#). The left pane of the window displays your mail folders. The top pane displays messages within the current folder. And, the bottom pane displays the contents of the currently selected message.

Figure 10-17. The Mozilla email screen

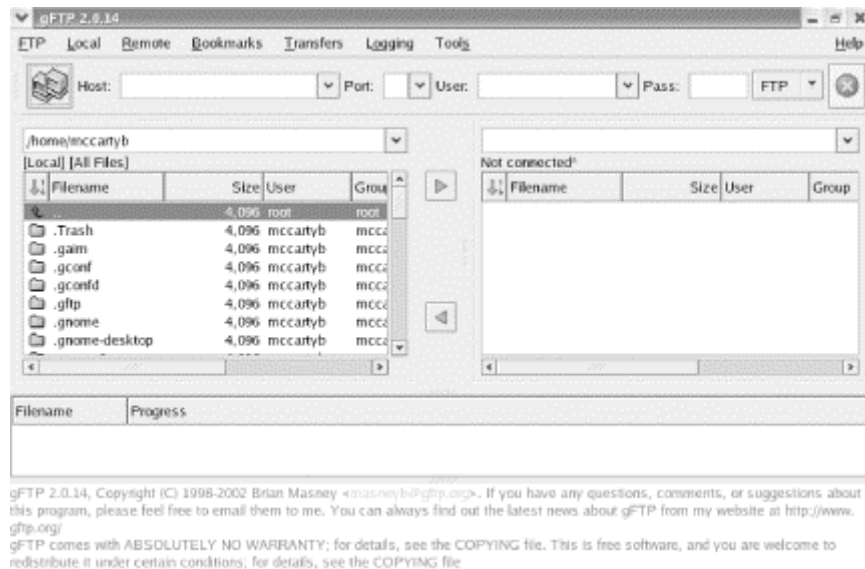


Mozilla's user interface is straightforward. For example, to send a message, click the Compose button in the toolbar. The best way to learn your way around Mozilla is to explore its menu and toolbar, clicking and observing the results. You'll soon be at home.

10.4 gFTP FTP Client

You can use your web browser to download files from an FTP server, but to upload files you need an FTP client. The gFTP client, included with Red Hat Enterprise Linux and Fedora Core, is an excellent choice, because its user interface resembles that of popular Windows FTP clients, such as WS-FTP. [Figure 10-18](#) shows the gFTP client, which can be launched by choosing Internet → More Internet Applications → gFTP from the main menu.

Figure 10-18. The gFTP FTP client



To connect to a remote system, specify the hostname, username, and password in the textboxes appearing on the toolbar. If the server permits anonymous logins, you can omit the username and password. To connect, click the Connect icon resembling a pair of computers at the left of gFTP's toolbar. To upload a file, click on the name of the file in the local list box at the left of the window and then click on the right-pointing arrow. To download a file, click on the name of the file in the list box at the right of the window and then click on the left-pointing arrow. When you've transferred all your files, choose Remote → Disconnect or click again on the Connect icon.



You can access an FTP server in command-line mode, if you prefer. [Chapter 12](#) explains how to do so.

10.5 Using wvdial

If you have your dial-up network connection working perfectly, you may have little interest in exploring **wvdial**. However, there are two reasons you should consider learning more about **wvdial**: you can use **wvdial** even if X isn't working or isn't installed, and you can use **wvdial** in shell scripts of your own design. [Chapter 13](#) includes an example script.

10.5.1 The /etc/wvdial.conf File

To configure **wvdial**, become the root user and issue the following command:
wvdialconf /etc/wvdial.conf

This command analyzes your system and creates a template configuration file, */etc/wvdial.conf*. You must edit this file to specify the username and password your ISP expects.

The contents of the template file look something like this:

```
[Dialer Defaults]
Modem = /dev/modem
Baud = 115200
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 S11=55 +FCLASS=0
ISDN = 0
; Phone = <Target Phone Number>
; Username = <Your Login Name>
; Password = <Your Password>
```

Edit the last three lines of the file, deleting the leading semicolon and space and substituting the proper phone number, username, and password required to connect to your ISP. When you're done, your file should look something like this:

```
[Dialer Defaults]
Modem = /dev/modem
Baud = 115200
Init1 = ATZ
Init2 = ATQ0 V1 E1 &C1 &D2 +FCLASS=0
ISDN = 0
Phone = 15625551100
Username = bill100
Password = donttell
```



Be sure that only the root user can read the *wvdial.conf* file. Use a file manager or the shell to change the file's permissions, if necessary. Otherwise, someone who uses your system may discover your password.

Now, you're ready to make a connection by issuing the following command:

```
# wvdial &
```

The command generates quite a bit of output, which makes further use of this virtual terminal distracting. The simplest solution is to switch to another terminal window, or to another virtual terminal by pressing **Alt-*n***, where *n* stands for the virtual terminal (1-7). Alternatively, you can direct the output of the command to a file, by typing this command in place of the one given earlier:

```
# wvdial 2>/tmp/wvdial.messages &
```

Of course, you'll need to consult the file if something goes wrong with **wvdial**. Do so by using the **less** command:

```
# less /tmp/wvdial.messages
```

Once your connection is up, you can browse the Web and access other Internet services. For now, simply verify that your connection is working by issuing the command:

```
# ping www.redhat.com
```

The **ping** command should report that echo packets were successfully received from the server. If not, check your name server configuration and other details, as described earlier in the chapter.

When you want to terminate the Internet connection, issue the command:

```
# killall wvdial
```

10.6 Configuring Linux to Use a Cable or DSL Modem

At one time, setting up a cable or DSL modem for use with Linux was difficult. The new Network Administration Tool included in Red Hat Enterprise Linux and Fedora Core often makes it a snap. But sometimes, the Tool fails to successfully configure the cable or DSL modem.

In such a case I recommend using a cable/DSL gateway router. Netgear and Linksys, among others, manufacture popular models. These inexpensive devices—often costing less than \$50 retail—sit between your cable or DSL modem and your home network.

Cable/DSL gateway routers generally provide a masquerading firewall and DNS proxy services. Better models have multiple ports so that you can connect several PCs without buying additional hubs or switches. Some recent models even provide a wireless LAN. I recently purchased a very satisfactory gateway router having four Ethernet ports and a built-in wireless adapter for \$30, net of rebates.

Because such devices are designed to work with as many cable and DSL configurations as possible, they work right out of the box most of the time. It's true that they generally provide no function that couldn't be provided—at least in principle—by a Linux PC. But they consume less power, occupy less space, make less noise, and require less configuration and administration than a Linux PC. I retired a quite venerable Pentium 166 Linux PC from cable modem gateway duty several years ago and have never regretted the decision.

Some useful resources when setting up Linux to access a cable or DSL modem include:

Hal Burgiss' DSL HOWTO for Linux, <http://www.tldp.org/HOWTO/DSL-HOWTO>

This somewhat dated HOWTO examines the technology behind DSL and offers advice on subscription, installation, configuration, and troubleshooting for Linux users.

Paul Ramey's Red Hat Linux 6.X as an Internet Gateway for a Home Network, <http://www.tldp.org/HOWTO/mini/Home-Network-mini-HOWTO.html>

This HOWTO relates to an obsolete version of Red Hat Linux, but nevertheless contains much useful information.

These resources are both a bit out of date, but are the most current available. That they haven't been recently updated seems to demonstrate that most Linux users are using gateway routers rather than Linux hosts as their means of connecting to their cable or DSL provider.

Chapter 11. Setting Up Network Services

Linux's greatest strength is its powerful and robust networking capabilities. The good news is that everything about Linux's networking setup is open to inspection and completely configurable. Nothing is hidden from the user, and no parameters are forced on you. The challenge is to get the most out of this setup.

Basic networking principles don't differ much between Windows and Linux, and indeed the principles aren't unfamiliar. This chapter begins with an overview of networking and then looks in more detail at Linux networking on a local area network (LAN) and the Internet. This chapter explains how to set up a LAN that includes a Linux Samba server, which lets Microsoft Windows and Unix systems access shared files and printers across the network. Samba not only lets you share files and printers, it can also be used to back up and restore files via the network.

This chapter also explains how to set up and configure the Apache web server and an SSH server, which you can use to remotely administer your system. These applications let you and others access data on your Linux system via the Internet. These applications will be most useful if your system is connected to the Internet 24/7. But, even if your connection is intermittent, you and others can access the services these applications provide whenever the connection is active.

Finally, the chapter also explains how to implement a basic host firewall to help protect your systems from unauthorized access via the Internet. Security is best when it consists of multiple layers. So, you're strongly urged not to connect any host to the Internet except through a network firewall. To help you ensure that your host and network firewalls are working, the chapter explains how to use *Nmap* to test your security.

If your host is protected—as it should be—by a host or network firewall, hosts on the other side of the firewall will not be able to connect to services offered by your host. To permit such hosts to access services, you must disable or customize the network and host firewalls. Ask your network firewall administrator to enable access to your host's services. Also, you must disable your host firewall, as described in [Section 11.5.2](#). However, you should disable the firewall only if your host resides on a trusted LAN that is protected from access by Internet hosts and other untrusted hosts.



Previous editions of this book provided instructions on setting up a DHCP server that enables you to manage network configurations centrally. However, Red Hat Enterprise Linux WS does not provide a DHCP server. If you need a DHCP server for your LAN, you may find it convenient to purchase a gateway router of the sort described in [Chapter 10](#). Many gateway routers provide a simple DHCP server adequate for use on home and small business LANs.

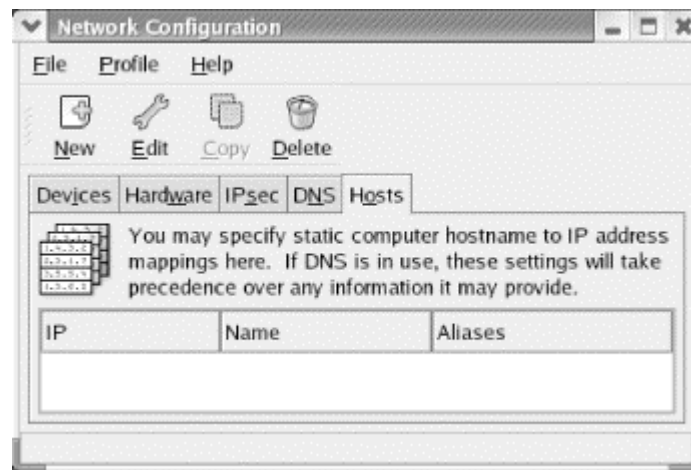
Most Internet services are configurable only by the root user. So, most of the operations in this chapter require that you be logged in as root, or possess temporary root privileges as indicated by the keys icon.

11.1 Configuring Hosts

Most systems attached to a network use DNS services to determine the IP address associated with a hostname. However, private hosts on your local network won't be known to your ISP's DNS server and therefore can be accessed only by IP address, not by hostname. Moreover, Samba and other local area network services won't operate correctly unless they can resolve hostnames.

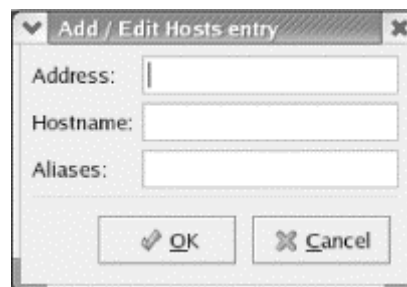
Fortunately, using the Network Administration Tool, you can configure your system to determine the IP address associated with a hostname even when DNS services are not available. To do so, use the Hosts tab of the Network Administration Tool, shown in [Figure 11-1](#).

Figure 11-1. The Hosts tab of the Network Administration Tool



To specify host information, launch the Network Administration Tool by choosing System Settings → Network from the main menu. Select the Hosts tab of the Tool, click New. The Add/Edit Hosts entry dialog box appears ([Figure 11-2](#)). Specify the IP address and name of the host. If desired, you can specify one or more aliases or abbreviated names for the host. By default, the host information includes an entry for the IP address 127.0.0.1, which is associated with the hostname *localhost*. The Add/Edit Hosts entry dialog box does not display this entry, which protects it from accidental modification.

Figure 11-2. The Add/Edit Hosts entry dialog box





Linux stores configured hostnames in the file */etc/hosts*, which you can edit by using a text editor, if you prefer. Windows has a similar file that you must revise in order for Windows hosts to interoperate with Samba and other Linux-based services. The file is named *hosts* and resides in an *etc* subdirectory of your Windows directory. Under Windows 2000, the path of the file is often *c:\windows\system32\drivers\etc\hosts*; the location varies slightly depending on the installation options.

Each line of the file describes a single host. The left column contains the host's IP address and the right column contains the associated hostname or names. If multiple names (aliases) appear, the principal name is given first. Simply use a text editor, such as Notepad, to add the appropriate lines describing your local hosts.

11.2 Samba

Once you've configured your hostnames, you're ready to provide services to hosts on the network. To provide printer and file sharing, Windows uses a facility known as the Server Message Block (SMB). This same facility is sometimes known as the Common Internet File System (CIFS), NetBIOS, or LanManager. Thanks to Andrew Tridgell and others, Linux systems provide support for SMB via a package known as Samba. Like SMB, Samba lets you:

- Authorize users to access Samba resources
- Share printers and files among Windows, OS/2, Netware, and Unix systems
- Establish a simple name server for identifying systems on your LAN
- Back up PC files to a Linux system and restore them

Samba has proven its reliability and high performance in many organizations. According to the survey at <http://www.samba.org/pub/samba/survey/ssstats.html>, Bank of America is using Samba in a configuration that includes about 15,000 clients, and Hewlett-Packard is using Samba in a configuration that includes about 7,000 clients.

11.2.1 Installing the Samba Server

If you've never installed and configured a network server, Samba is a good place to begin; its installation and configuration are generally straightforward.

The Samba server includes the **nmbd** and **smbd** programs (which run as daemons), several utility programs, manpages and other documentation, and three configuration files: */etc/samba/smbusers*, */etc/samba/smb.conf*, and */etc/samba/lmhosts*. The *smbusers* file associates several user accounts that are special to Samba with Linux user accounts; for example, it associates the Samba user IDs, *administrator* and *admin*, with *root*. Generally,

you don't need to change *smbusers*. Likewise, you don't generally need to revise *lmhosts*. You'll learn how to configure the *smb.conf* file shortly.

The simplest way to install Samba is to select the Windows File Server package group during system installation. However, if you failed to do so, you can install Samba by using the Package Management Tool.

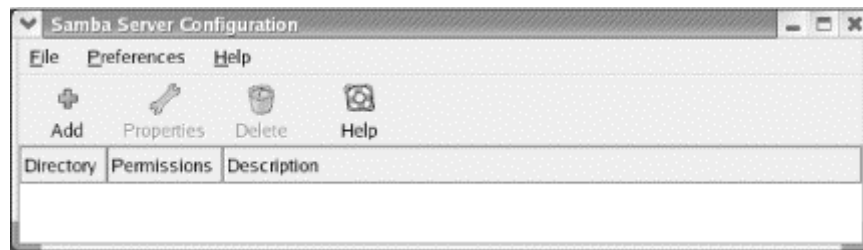
11.2.2 Configuring Samba

The `/etc/samba/smb.conf` file lets you specify a variety of options that control Samba's operation. You can edit the file by using your favorite text editor; however, the Samba Server Configuration Tool enables you to view and change options using your web browser, which is generally much easier than using a text editor. The Samba Server Configuration tool verifies the values of parameters you enter and provides online help. To install the Samba Server Configuration Tool, select System Settings → Add/Remove Applications from the main menu. The Package Management Tool appears. Select Server Configuration Tools → Details → `redhat-config-samba`. Select Close → Update, and insert the appropriate installation CDs when prompted. To launch the tool, select System Settings → Server Settings → Samba from the main menu. [Figure 11-3](#) shows the Samba Server Configuration screen.



The Samba menu item may not appear immediately following the installation of the Windows File Server package. If your system's menu lacks the Samba menu item, log out and log in again. Doing so should cause your desktop manager to regenerate the menus, causing the Samba menu item to appear.

Figure 11-3. The Samba Server Configuration screen



Configuring your Samba server involves configuring:

- Security options
- Users
- Shares

The following three sections explain how to configure your server.

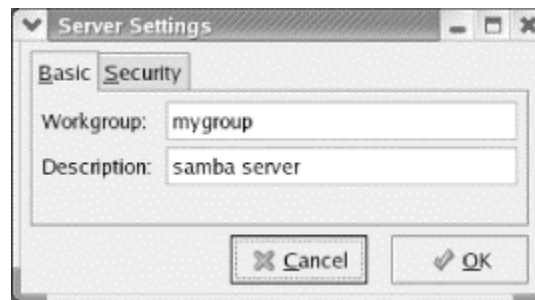


Although Samba supports sharing printers, the Samba Server Configuration Tool does not provide access to the related Samba configuration items. Instead, the Samba Server Configuration Tool automatically configures shared access to all available printers. If you want to manually specify options related to a shared printer, see the manual page on Samba configuration, *smb.conf*.

11.2.2.1 Configuring Samba security options

To configure your Samba server, select Preferences → Server Settings from the Samba Server Configuration Tool menu. [Figure 11-4](#) shows the Server Settings dialog box that appears. The dialog box has two tabs. The Basic tab lets you specify the name of the workgroup that your Samba server will serve, and a description of your Samba server. If your Samba server is the only Samba server on your LAN, you can accept the default workgroup name, **mygroup**.

Figure 11-4. The Basic tab of the Server Settings dialog box



The Security tab ([Figure 11-5](#)) enables you to specify options that control access to your server and the file and printer shares it provides.

Figure 11-5. The Security tab of the Server Settings dialog box



The most important security option is the authentication mode, which determines how the server decides who can use which shared files and printers. The dialog box provides five authentication modes from which you can choose:

ADS

In ADS mode, the Samba server acts as a member server within an Active Directory Services (ADS) realm. To use this mode, you must install and configure Kerberos authentication on your workstation. The realm must also be configured to accept your workstation's Samba server. Setting up a Samba server in ADS mode requires special networking knowledge and skill and is not considered further in this book.

Domain

In Domain mode, the Samba server relies on a Windows domain controller to perform authentication. Specify the NetBIOS name of the domain controller in the textbox labeled Authentication Server.

Server

In Server mode, the Samba server relies on another Samba server to perform authentication. Specify the NetBIOS name of the domain controller in the textbox labeled Authentication Server. If the Samba server cannot contact the designated authentication server, it attempts to authenticate using User mode.

Share

In Share mode, a user is prompted for a username and password each time he attempts to access a shared file or printer.

User

In User mode, a user is prompted for a username and password when he initially contacts the Samba server.

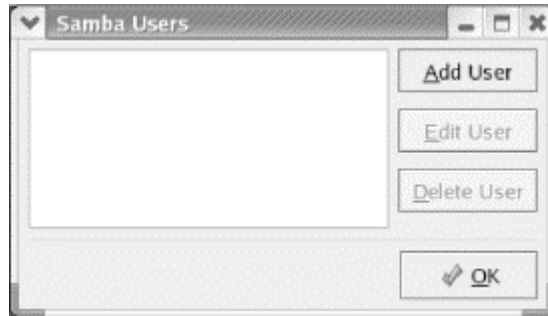
The default authentication mode, User, is appropriate for most common situations. The Encrypt Password option is initially enabled. You should not disable this option if your network includes clients using Windows 98 or any more recent version of Windows. PCs running recent versions of Windows send passwords in encrypted form and, unless the appropriate option is enabled, your Samba server will be unable to communicate properly with them.

The Guest Account option enables you to specify a default account, known as the guest account. Users logging into your Samba server will automatically receive the privileges assigned to the guest account in addition to any privileges assigned to their personal account. To maximize LAN security, you should not specify a guest account. After specifying the security options, click OK. The Server Settings dialog box saves your changes, which take effect immediately.

11.2.2.2 Configuring Samba users

To configure Samba users, select Preferences → Samba Users from the Samba Server Configuration Tool menu. The Samba Users dialog box ([Figure 11-6](#)) appears.

Figure 11-6. The Samba Users dialog box



To create a new Samba user, click the Add User button. The Create New Samba User dialog box ([Figure 11-7](#)) appears. Choose an existing Linux username by using the listbox titled Unix Username. The user will receive the privileges associated with the specified username. The user may log in to the Samba server by using a different username. Specify the login username by using the textbox labeled Windows Username. Finally, specify the password by entering in the the textboxes labeled Samba Password and Confirm Samba Password. Click OK to create the Samba user..

Figure 11-7. The Create New Samba User dialog box



Sometimes, the Create New Samba User dialog box gets stuck, displaying only a single Linux username rather than the Unix Username listbox. Deselecting any selected username before clicking Add User may not solve this problem. If you find yourself in this situation, exit the Samba Server Configuration Tool and then re-launch it. This will generally clear the problem

You can use the Samba Users dialog box to edit or delete a Samba user. To do so, highlight the user and click Edit User or Delete User, as appropriate.



Changes made using Edit User take effect when you dismiss the Edit User dialog box. Samba users are immediately deleted when you click Delete User. Therefore, be sure you click the proper buttons and enter

the proper data when using the Samba Users dialog boxes.

Samba passwords are not automatically synchronized with Linux passwords. To change the Samba password associated with a Samba user, user the Edit User button of the Samba Users dialog box.

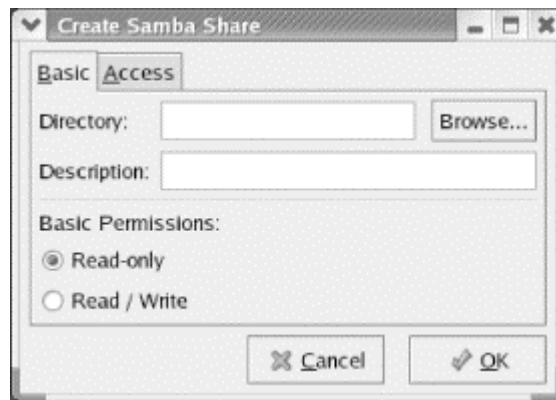
11.2.2.3 Configuring Samba shares

Once you've configured one or more Samba users, you're ready to configure Samba shares. To do so, click Add on the Samba Server Configuration Tool toolbar. The Create Samba Share dialog box, shown in [Figure 11-8](#), appears.



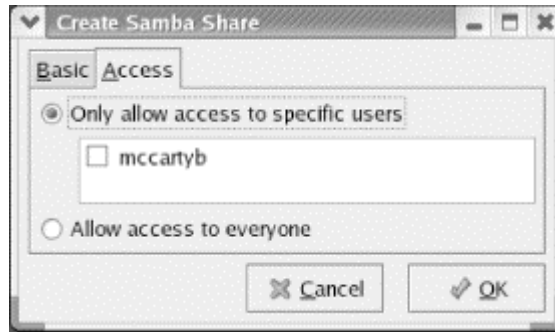
Any user having valid access to a Samba server (which may require the user to login when trying to access a share) has automatic access to her home directory.

Figure 11-8. The Create Samba Share dialog box



To specify the Linux directory to be shared, enter the directory name in the textbox labeled Directory or use the Browse button to choose a directory. You can specify a description of the share in the textbox labeled Description. By default, users are permitted only read access to shares. If you want users to be able to write to the share, enable the Read/Write radiobutton. Click OK to save your changes and immediately activate the share. If you want to limit access to the share to certain users, you can use the Access tab of the Create Samba Share dialog box ([Figure 11-9](#)). Enable the radiobutton labeled "Only allow access to specific users," and enable the checkbox associated with users permitted to access the share.

Figure 11-9. The Access tab of the Create Samba Share dialog box



11.2.3 Starting and Stopping Samba

If you want Samba to start automatically when you boot your system, use the Service Configuration Tool, described in [Chapter 9](#), to associate the *smb* service with the current runlevel.

To stop Samba, highlight the Service Configuration Tool's entry for the *smb* service and click Stop.

11.2.4 Troubleshooting Samba

To verify that Samba is working, use the Create Samba Share dialog box to temporarily create a read-only share accessible to an arbitrary Samba user. On the Windows host, launch the Explorer and choose Tools → Map Network Drive. The Map Network Drive dialog box appears. Click Connect using a different username and specify a username and password that you configured Samba to accept. Click OK to return to the Map Network Drive dialog box. Specify the hostname and share name in the Folder textbox by using the Windows convention, `\\server\share`, where *system* is the hostname of your Samba system and *share* is the name of a share you created. You'll find more information on using Samba shares in the next section.

If you can't access the share, consider the following likely reasons:

- Your host firewall is blocking access.
- You specified the wrong username or password.
- The username and password are not the same on the Samba and Windows hosts.
- You haven't authorized the user to access Samba.
- Networking isn't properly configured on the Samba or Windows host.

If your host firewall is enabled, Samba will not function. You can easily disable the host firewall. However, you should not do so if your system resides on a LAN that is connected to the Internet or another un-trusted network unless the LAN is protected by a network firewall, such as a gateway router. To disable the firewall, select System Settings → Security Level from the main menu. In the Security Level listbox, select the value Disable Firewall and click OK.

If you're unable to find the problem, consult the documentation that accompanies Samba. In particular, use a web browser to peruse the file *diagnosis.html*, which resides in the

`/usr/share/doc/samba-*/docs/html/docs` directory. This file includes a step-by-step procedure for verifying the operation of your Samba server. When a step fails, you can consult the file to determine the likely causes and how to go about fixing the problem. Chances are, you'll be able to administer Samba without outside help, but if not, you'll find the participants in the *comp.protocols.smb* newsgroup to be helpful. Another resource is O'Reilly's *Using Samba*, by Robert Eckstein, David Collier-Brown, and Peter Kelly. Since the book was published under the Open Publication License (OPL), *Using Samba* is also available online in electronic form at <http://www.oreilly.com/catalog/samba>.

Like any network server, Samba provides a wealth of options and facilities. If you thoroughly explore these facilities, you're likely to break your server. To avoid problems, you should keep a backup copy of your `/etc/samba/smb.conf` file. Doing so can be as easy as issuing the following command after Samba is up and running:

```
# cp /etc/samba/smb.conf /etc/samba/smb.conf.bak
```

Then, if your server ceases to work, you can restore your old configuration by issuing the command:

```
# cp /etc/samba/smb.conf.bak /etc/samba/smb.conf
```

You'll also need to restart the *smb* service.

11.2.5 Samba Client Configuration and Use

Once you've got your Samba server up and running, you can access it via Windows and Linux. This section shows you how to access the Samba server and also how to use your Samba server to create backups of important datafiles on client systems.

SMB clients are also available for most popular operating systems, including OS/2 and Mac OS (including Mac OS X). You shouldn't expect to have trouble getting them to work with Samba. If your client seems not to work, simply follow the procedure given earlier in [Section 11.2.4](#).

11.2.5.1 Windows client

Windows 3.11, 9x, Me, and NT—including Windows 2000 and XP, which are updated releases of Windows NT—have built-in support for the SMB protocol, so systems running these operating systems can easily access your Samba server's resources. Under Windows 9x/NT, you can access Samba resources by using the Windows Explorer. Log on with a user account that's authorized to access Samba resources, then click Network Neighborhood, and you should see a subtree that corresponds to your workgroup. Click that subtree, and you should see a subtree that corresponds to your Samba server. By expanding the subtree, you can see the browseable file and printer shares that are available. You can easily drag-and-drop files to and from a shared directory, assuming your user account is permitted the necessary access.

To use a shared printer, click Start → Settings → Printers and Faxes, or the equivalent on your system, and then double-click Add Printer. The wizard will guide you through the setup procedure. Simply choose the Network Printer option and browse to select the desired printer. If you have difficulty locating the printer share, you can manually specify its name. To do so, type two backward slashes, followed by the name of your Samba server, followed

by a single backslash, followed by the name of the printer. For example, if you want to access a printer named *lp* on the Samba server known as *SERVER*, you'd type `\\SERVER\\lp`.

You can map a file share to a drive letter by using the Tools → Map Network Drive menu item of the Windows Explorer. Simply select an available drive letter and type the name of the file share, which consists of two backward slashes, followed by the name of your Samba server, followed by a single backslash, followed by the name of the file share. For example, if you want to access a file share named *db* on the Samba server known as *SERVER*, you'd type `\\SERVER\\db`.

If you have difficulty connecting to your Samba server, follow the procedure given earlier in [Section 11.2.4](#).

The Samba package includes a simple SMB client that can access your Samba server and other SMB servers accessible to your system. To demonstrate that your client and server are working, log on using a user account that has Samba authorization and issue the following command:

```
$ smbclient -L localhost
```

If you're prompted for a password, be sure to enter the Samba password for the user account rather than the Linux password.

You should see a list of the browseable shares available on your server. To query a different SMB server, issue the following command:

```
$ smbclient -L server
```

where *server* is the name of the SMB server you want to contact. Rather than logging on using an authorized user account, you can explicitly specify a user account by using this command form:

```
$ smbclient -L server -U userid
```

To actually access resources via SMB, use the following command form:

```
$ smbclient 'service' -U userid
```

where *service* specifies the name of the SMB host and share and *userid* specifies the user account to be used. The name of the SMB host should be preceded by two slashes and followed by one slash; for example:

```
$ smbclient //server/myshare -U billmccarty
```



Although Windows uses backward slashes to specify server names of SMB shares, you can use forward slashes when specifying arguments to the Linux **smbclient** command.

If the SMB server accepts your request, the client displays a special prompt:

```
smb: dir>
```

where *dir* indicates the current working directory on the SMB server. To download a file from the server, issue the command:

```
get file
```

where *file* specifies the name of the file to be downloaded. To upload a file to the server, issue the command:

```
put file
```

where *file* specifies the name of the file. To list the contents of the current directory, issue the command:

```
dir
```

To enter a subdirectory, issue the following command, where *dir* specifies the name of the subdirectory:

```
cd dir
```

You can return to the parent directory by issuing the command:

```
cd ..
```

You can obtain a list of commands by issuing the command **help** or, to obtain help on a particular command, by issuing the command:

```
help command
```

where *command* specifies the command that you need help with. To exit the SMB client, issue the command **exit**.

You can use the *smbprint* script included in the Samba package to print Linux files by using a printer share. However, you'll probably have to do some tweaking of configuration files and adjusting of shell scripts to get *smbprint* to work.

11.2.5.2 Using the Linux Samba client for file backup and recovery

One of the most practical uses of the Linux SMB client is creating backup copies of files stored on a Windows system. To do so, simply share the drive or directory containing the files you want to back up. Using the Windows Explorer, right-click the drive or directory, click Properties, click the Sharing tab, and select the desired share options. Then, access the share from Linux using *smbclient*. Once you have the SMB prompt, move to the directory you want to back up and issue the SMB **tar** command:

```
tar c backup.tar
```

The syntax of the SMB **tar** command resembles that of the **tar** command, though it supports only a handful of options. When you issue the SMB **tar** command with the **c** option, the files of the current directory and all its subdirectories will be backed up and stored in the file *backup.tar* on your Linux system. Of course, you can specify a filename other than *backup.tar* if you wish (although the *.tar* extension is required). Once you've created the backup file, you can write it to a tape, a writable CD-ROM, or other media. If your backup requirements are meager, it may be sufficient merely to have a copy of the file on both your Windows and Linux systems.

To restore a backup, move to the directory where you want the files restored and issue the SMB **tar** command:

```
tar x backup.tar
```

The SMB client restores each file from the *backup.tar* file. Of course, you must have write access to the shared directory in order to be able to restore files.

11.3 Running Apache

Installing and configuring the Apache web server is not difficult. Once your web server is up and running, other Internet users can view and download documents within the web-enabled directories on your Linux system. This section explains the installation and configuration of Apache, the most popular web server on the Internet.

11.3.1 Installing Apache

Use the Package Management Tool to install the Web Server package group, which contains the Apache web server. This package group contains over two dozen optional packages. Unless you specifically require the support they provide, you should deselect them before initiating the installation. However, you should not generally deselect the optional *redhat-config-httpd* package, which is helpful in configuring Apache.



When you install Apache, the procedure may fail owing to missing packages. It appears that Red Hat's developers erroneously specified some required packages in the Apache web server package group as optional. Select the following packages for installation and try again:

- *commons-logging*
- *xerces-j*
- *jakarta-regexp*
- *bcel*
- *redhat-java-rpm-scripts*
- *xalan-j*

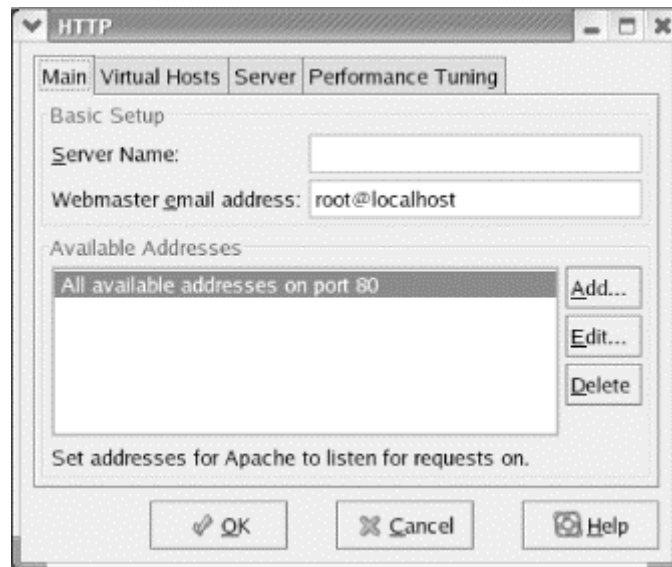
11.3.2 Configuring Apache

Configuring a web server can be as easy or as difficult as you choose. Like other web servers, Apache provides seemingly countless options. As distributed with Red Hat Enterprise Linux and Fedora Core, Apache has a default configuration that generally requires only a little tweaking before use. Apache's configuration files reside in the directory */etc/httpd/conf*. For historical reasons that no longer apply, Apache has three configuration files:

- *access.conf*
- *httpd.conf*
- *srm.conf*

However, the only configuration file that's currently used is *httpd.conf*. The easiest way to perform a basic configuration of Apache is with the Apache Configuration Tool. To configure Apache, choose System Settings → Server Settings → HTTP from the main menu. The main configuration screen ([Figure 11-10](#)) appears.

Figure 11-10. The Apache Configuration Tool



The Apache Configuration Tool contains four tabs: Main, Virtual Hosts, Server, and Performance Tuning. The Main tab lets you specify the following:

Server Name

This is the hostname of your system. Often, this will be *www.domain.com*, where *domain.com* is the name of your domain.

Webmaster email address

Any messages concerning the web server will be sent to this address.

Available Addresses

This is the IP address (or addresses) on which the web server listens.



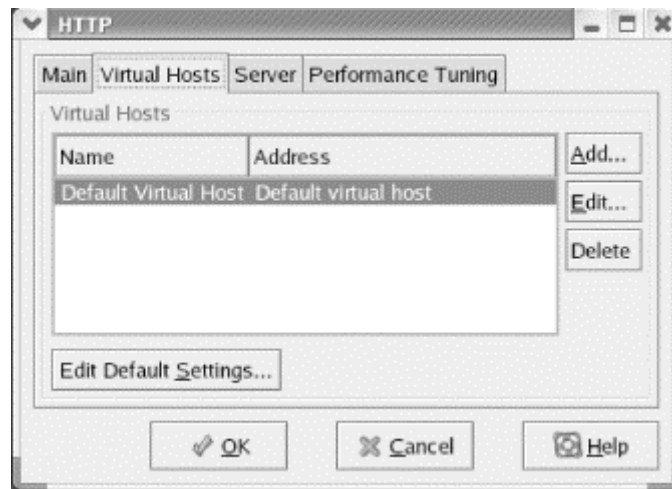
When offering a web server (or any other server) you should make sure that a DNS server is available to resolve your host's name to its IP address. If your host is reachable on a network but the domain administrator won't add your host's name to the DNS database, you can specify the web server's IP address as its server name. Users can then contact the web server using its IP address rather than its host name.

You should specify the Server Name and Webmaster email address. Unless your system has multiple network adapters or you want to run the web server on a nonstandard port (that

is, a port other than 80), you don't need to modify the Available Addresses configuration item.

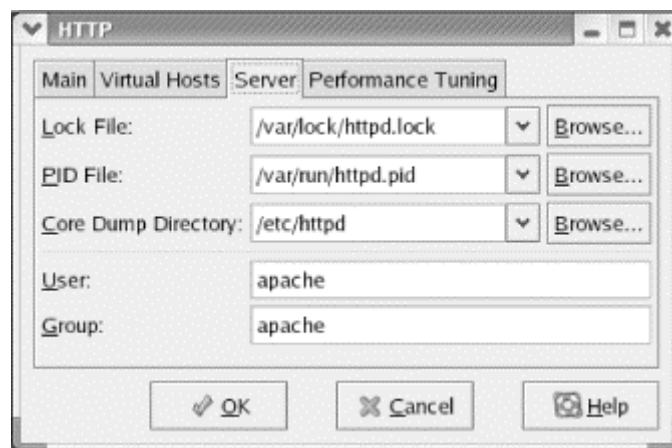
The Virtual Hosts tab, shown in [Figure 11-11](#), lets you specify virtual hosts. Virtual hosting is a feature that lets you host multiple web sites with a single IP address. For example, both *http://www.myfirstsite.com* and *http://www.myothersite.com* could be hosted on the same system using a single IP address. Virtual hosting is not compatible with HTTP 1.0 browsers. However, only a small minority of web users currently use HTTP 1.0 browsers, which are quite out of date.

Figure 11-11. The Virtual Hosts tab



The Server tab ([Figure 11-12](#)) lets you specify the location of important files and directories and the user account and group used by Apache. You generally should not alter these configuration items.

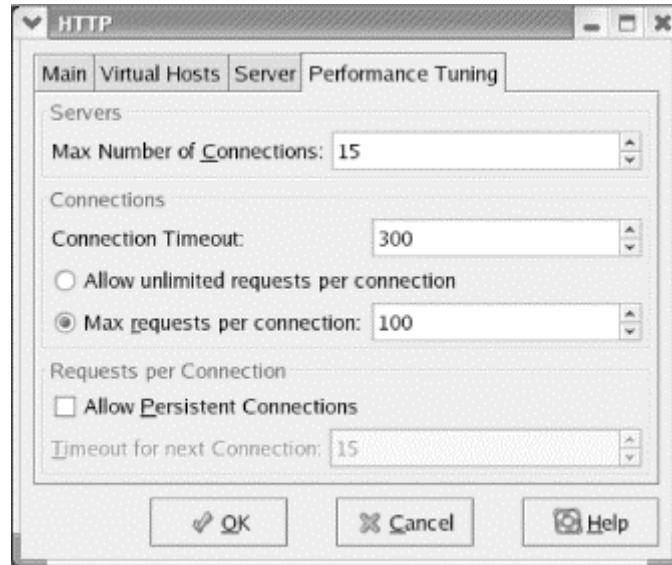
Figure 11-12. The Server tab



The Performance Tuning tab ([Figure 11-13](#)) provides access to configuration items that let you optimize Apache's performance. Unless your web server will see very heavy service,

you should reduce the maximum number of connections. A value of 15 is more than appropriate for a personal web server.

Figure 11-13. The Performance Tuning tab



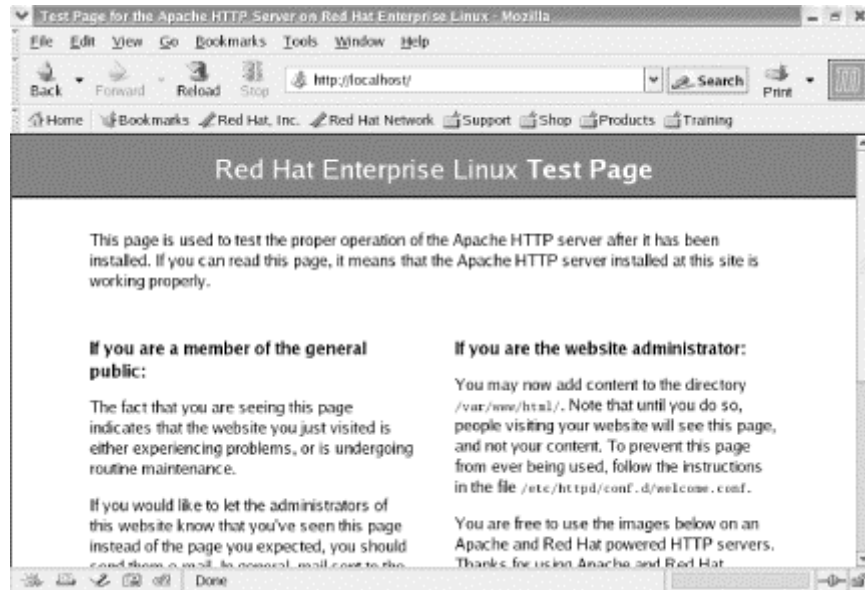
If you're interested in exploring the many other configuration options provided by Apache, see the Apache documentation, available at <http://httpd.apache.org/docs-2.0>.

11.3.3 Starting and Stopping Apache

Once you've configured your web server, you can start, stop, and restart it by using the Service Configuration Tool. You can also associate the *httpd* service with one or more runlevels, so that it starts automatically when your system boots. You can use Mozilla to test your web server by pointing Mozilla to the URL *http://localhost/*.

If you're using Red Hat Enterprise Linux, you should see a screen that resembles [Figure 11-14](#). If you're using Fedora Core, you should see a similar page with a different title and color scheme.

Figure 11-14. The Apache start page as viewed with Mozilla



Once you can access your web server locally, try accessing it from a remote computer. This should be as simple as forming a URL that includes the fully qualified hostname or IP address of your system (that is, the host and domain names), for example, *http://mysystem.mydomain*. However, bear in mind that you can reference your system by hostname only if its IP address is known to the DNS server or contained in the client's */etc/hosts* file.

If you change Apache's configuration, you must restart the server so that the server reads the revised configuration file. You can do so by using the Service Configuration Tool.

11.3.4 Creating Web Pages

You can create HTML pages in */var/www/html*, owned by the *root* user account. These pages are accessible via the URL <http://www.domain.com>, where *domain.com* is the name of the host.

Alternatively, users can create a *public_html* subdirectory within their home directory, for example, */home/joepublic/public_html*. There, they can publish files that are web-accessible. To access such files, use a special URL that consists of a tilde (~) followed by the name of the user account. For example, *http://www.domain.com/~joepublic* refers to the user *joepublic*'s web directory. However, support for user publishing of web pages is not enabled by default. To enable it, edit the file */etc/httpd/conf/httpd.conf*, changing the line:

```
UserDir disable
```

to:

```
UserDir enable all
```



Any configuration changes made manually are overridden when you launch the HTTP configuration tool. Therefore, you must remake them each time you run the tool.

Then, restart the *httpd* service. If you want to permit only specified users to publish web pages, replace the keyword *all* with a list of usernames, separating each from the next with one or more spaces:

```
UserDir enable billmccarty andyoram
```

You must also change the permissions of files and directories you want to be accessible via the Web. For the user *joepublic*'s web pages to be web-accessible, the *apache* user account or group must have execute access to the directories */home*, */home/joepublic*, and */home/joepublic/public_html*. Moreover, if automatic directory indexes are desired, the *apache* user account or group must have read access to the */home/joepublic/public_html* directory. The files themselves must be readable by the *apache* user.



The rule of thumb is to give the */home/joepublic* directory permissions 711 and the */home* and */home/joepublic/public_html* directory permissions 755. Within */home/joepublic/public_html*, directories should have permissions 755 and files should have permissions 644. However, your particular needs may dictate that other permissions should be used.

If you have trouble accessing web pages on your server, check Apache's log files, which you can view via the System Logs Tool. The log files may provide valuable clues to help you understand what's going wrong.

11.4 The Secure Shell

The Secure Shell (SSH) lets you connect to a system from another system via TCP/IP and obtain a shell prompt, from which you can issue commands and view output in a secure fashion. SSH works similarly to the older and possibly more familiar Telnet service, but differs in that conversations between SSH and its clients are sent in encrypted form so hackers cannot easily discover private information, including user account names and passwords.

11.4.1 Installing SSH

The installation procedure automatically installs an SSH client and server and associates the *sshd* service with runlevels 3-5. You can start, stop, and restart the *sshd* service and changes its associations with runlevels by using the Service Configuration Tool. The service must be running in order to respond to clients.



The SSH service has several configuration files, residing in */etc/ssh*. You don't have to modify them to get SSH running. If you're curious about them, view the *sshd* manpage.

11.4.2 Using SSH

To verify that the SSH server is properly running, you can access it via a client on the local system by issuing the following command:

```
$ ssh localhost
```

The client will attempt to log you on to the local system using your current user account and will prompt you for your password. If you supply the correct password, you should see a shell prompt, indicating that the client and server are functioning correctly. Type **exit** and press **Enter** to exit SSH.

To log on to a remote system, simply specify the hostname or IP address of the remote system in place of *localhost*. If you want to log in to a user account other than one named identically to the account you're using on the local system, issue the command:

```
$ ssh userid @ host
```

where *host* is the hostname or IP address of the remote host and *userid* is the name of the user account you want to use. For example:

```
$ ssh billmccarty@example.com
```

You can use the SSH client's **scp** command to transfer files to or from a remote system running an SSH server. To transfer a file to a remote system, issue a command such as this one:

```
$ scp file userid @ host : destination
```

where *file* is the path of the file to be transferred, *host* is the hostname or IP address of the remote host, *destination* is the directory to which the file should be transferred, and *userid* is your user account on the remote system. If given as a relative path, the destination path is understood as being relative to the home directory of the specified user. For example:

```
$ scp rhbook_rev.txt billmccarty@example.com:files
```

To transfer files to your home directory on the remote system, omit the **path** argument; however, retain the colon or the command will be misinterpreted.

You can specify multiple files to be transferred if you like. You can use shell metacharacters to specify a set of files to be transferred. You can also specify the **-r** flag, which specifies that **scp** should recursively copy a directory. For example, the following command copies an entire directory to the remote system:

```
$ scp -r Desktop billmccarty@example.com:files
```

To transfer files from a remote system, issue a command based on this pattern:

```
$ scp userid @ host : file path
```

where *host* is the hostname or IP address of the remote system, *file* is the path of the file to be transferred, *path* is the destination path of the file, and *userid* is your user account on the remote system. For example:

```
$ scp billmccarty@author.example.com:/out/ch12.doc files
```

This command would log in the user *billmccarty* to *author.example.com/out*, retrieve the *ch12.doc* file, and place it in his *files* directory.

SSH also provides the **sftp** command, which lets you transfer files in much the same way the **ftp** command does. The command has the following form:

```
$ sftp user @ host
```

The command will prompt for the password associated with the specified user account. For example, to transfer files to and from the host *author.example.com*, you could issue the following command:

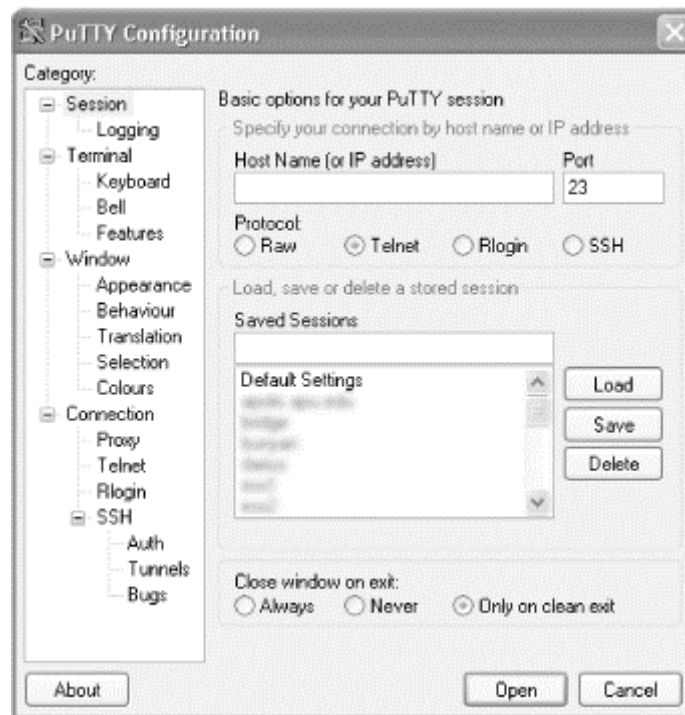
```
$ sftp billmccarty@author.example.com
```

After establishing a connection to the specified host, the **sftp** command presents a prompt that lets you enter commands similar to those supported by the **ftp** command. Use the **help** command to learn more about the supported commands.

11.4.3 Using a Windows SSH Client

To log on to your Linux system from a remote system via SSH, you must install an SSH client on the remote system. A suitable client for Windows is Simon Tatham's *PuTTY*, available at <http://www.chiark.greenend.org.uk/~sgtatham/putty>. Simply download *PuTTY* to any convenient directory (the *windows* directory is a good choice). The program doesn't have a setup script; you can run it by selecting Start → Run and typing **putty**; if the directory in which *PuTTY* resides is not on the execution path, you must type the drive, path, and filename. Alternatively, you can create a shortcut that spares you the trouble. [Figure 11-15](#) shows *PuTTY*'s main screen.

Figure 11-15. PuTTY's main screen



SSH is designed to be secure. Nevertheless, various implementations of SSH have suffered from vulnerabilities that have enabled attackers to compromise systems running SSH. Therefore, unless you need SSH, you should disassociate the service from all run levels, by using the Services Configuration Tool.

If you do need SSH, you can make the service more resistant to attack by restricting the hosts from which SSH will accept connections. To do so, edit the */etc/hosts.allow* and */etc/hosts.deny* files as explained in the remaining paragraphs of this section.

TCP Wrappers is a networking facility that limits access to certain TCP facilities, SSH among them. Together, the */etc/hosts.allow* and */etc/hosts.deny* files specify the IP addresses of hosts that are authorized to access TCP services that support TCP Wrappers, including SSH. More precisely, the */etc/hosts.allow* file specifies hosts that are authorized; the */etc/hosts.deny* file specifies hosts that are not authorized.

By default, the */etc/hosts.allow* file has the following contents:

```
#
# hosts.allow      This file describes the names of the hosts which are
#                  allowed to use the local INET services, as decided
#                  by the '/usr/sbin/tcpd' server.
#
```

At the bottom of the file, add a line of the following form specifying the IP addresses of hosts allowed to use the SSH service:

```
sshd: 127.0.0.1 1.2.3.4 1.2.3.5 1.2.4.
```

The line consists of the literal **sshd**: followed by a list of IP addresses, each separated by the next by one or more spaces. An IP address can be specified in shortened form, such as **1.2.4.** (notice the trailing dot). This specification permits all IP addresses in the range 1.2.4.0 to 1.2.4.255 to use the SSH service. Your list of IP addresses should generally include 127.0.0.1, so that you can test the SSH server by using that IP address, which is the standard IP address of the local host.

The changes to the */etc/hosts.allow* file have no effect until you make the proper change to the */etc/hosts.deny* file. By default, the */etc/hosts.deny* file has the following contents:

```
#
# hosts.deny      This file describes the names of the hosts which are
#                  *not* allowed to use the local INET services, as decided
#                  by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow.  In particular
# you should know that NFS uses portmap!
```

The comment in the file references a "portmap line" that does not appear in the file. Ignore this minor error and add the following line at the end of the file:

```
sshd: ALL
```



To configure TCP wrappers to protect another TCP service or services, specify each service on a separate line of the */etc/hosts.allow* file. To determine the name by which you should refer to a service,

search the file `/etc/services` and use the name given there.

Be sure to test your configuration to ensure that it enables the authorized hosts and blocks others. Also, bear in mind that TCP wrappers can block access to TCP services but cannot generally block access to UDP services.

11.5 Implementing a Basic Firewall

Sometimes you may want a host to provide certain services to only local clients or clients on other hosts of a network that you control. If your network is connected to the Internet, you can use a *firewall* to prevent undesired access to services. A Linux firewall depends on certain kernel facilities to examine incoming and outgoing packets. Packets that fail to pass specified rules can be rejected, preventing undesired access to private services. Unlike TCP Wrappers, a firewall does not require special support from applications or services it protects. And, a firewall can work with protocols other than TCP, such as UDP and ICMP.

11.5.1 Configuring the Firewall

To configure a firewall, launch the Security Level Tool by choosing System Settings → Security Level from the main menu.

The Security Level Tool ([Figure 11-17](#)) appears.

Figure 11-17. The Security Level Configuration dialog box



The Security Level Configuration dialog box lets you enable or disable the firewall. In addition, you can specify that requests for any of several predefined services are allowed to freely transit the firewall. To do so, simply enable the checkbox associated with the name of the service.



By allowing service requests to transit the firewall, you may expose your system to network-based attacks. Therefore, don't specify trusted services casually or unnecessarily.

You can use the Trusted devices checkbox to specify that packets originating from the specified device will not be blocked by the firewall. This facility is useful when a host has two network adapters: one associated with a public network, such as the Internet, and another associated with a private network. By specifying the network adapter associated with the private network as a trusted device, you permit clients on the private network free access to services, while blocking clients on the public network from access other than that permitted by the firewall configuration.

11.5.2 Controlling the Firewall

For the firewall to be effective, it must be enabled and the associated *iptables* service must be running. To start, stop, or restart the *iptables* service, you can use the Service Configuration Tool. Generally, you should use the Tool to associate the *iptables* service with runlevels 2-5, so that your system is protected when networking is active.

11.6 Checking Your Security by Using Nmap

Many users have misconfigured firewalls such that they are all but useless in defending against attacks. You can determine the state of your firewall by using the same tool hackers use to find its weaknesses, Nmap. To install Nmap, use the Package Management Tool to install the System Tools package group, and the extra package *nmap*.

Unfortunately, Nmap doesn't have a place on the GNOME or KDE menus. Instead, you use Nmap by issuing shell commands from a terminal window.

Nmap has enough features to be the topic of a book. You can learn more about Nmap from its author's web site, <http://www.insecure.org/>. A vanilla use of Nmap involves configuring it to send TCP or UDP packets to every important port of a specified system. The target system's responses reveal whether it has services listening on scanned ports.

To scan your system, open a terminal window, become the root user and issue a command having the form:

```
# nmap -sT -sU -p 1-1023 xxx.xxx.xxx.xxx
```

where *xxx.xxx.xxx.xxx* is the IP address of your system. After a few seconds, Nmap reports the ports on which services are listening:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on (192.168.83.131):
(The 2042 ports scanned but not shown below are in state: closed)
Port      State      Service
```

```
22/tcp      open       ssh
68/udp      open       dhcpclient
80/tcp      open       http
123/udp     open       ntp
```

```
Nmap run completed -- 1 IP address (1 host up) scanned in 5 seconds
```

Some ports listed in Nmap's report may not be remotely accessible. Therefore, to accurately determine the state of a system, you must scan it remotely. If a remote scan of your system discloses unexpected open ports, you should check your firewall and service configurations to determine whether anything is amiss.

Open ports above port 1023 that come and go with successive scans are not generally cause for concern. Usually, these are associated with established connections and are not really open. However, they can also be an early indication of the presence of a Trojan horse installed by an attacker. It's good practice to scan your hosts after you configure them so that you can later distinguish normal from abnormal behavior.



Don't scan someone else's system without permission. In some jurisdictions, it's a crime to do so. Even when scanning is legal, it may violate your ISP's terms of use. And, even if your ISP doesn't object, the scanned host's administrator may believe that the host is under attack, diverting his or her attention from other business. So, it's best to scan only systems on your local network, under your direct administrative control.

11.7 Network Security Tips

Anyone who administers a system connected to the Internet needs to know something about network security. It's not uncommon for systems connected to the Internet to be probed by would-be hackers several times daily. If a would-be hacker manages to detect a vulnerability, the hacker can often exploit it in a matter of seconds. Therefore, it's almost certain that a system administrator ignorant of network security will eventually suffer a system break-in.

Network security is a large and sophisticated topic that can be only cursorily surveyed in a book such as this. Concerned readers should consult books such as the following:

- *Building Internet Firewalls*, Second Edition, by Elizabeth D. Zwicky, Simon Cooper, and D. Brent Chapman (O'Reilly)
- *Building Secure Servers with Linux*, by Michael D. Bauer (O'Reilly).
- *Computer Security Basics*, by Deborah Russell and G.T. Gangemi, Sr. (O'Reilly)
- *Linux Security Cookbook*, by Daniel J. Barrett, Richard Silverman, Robert G. Byrnes (O'Reilly).
- *Linux Server Hacks*, by Rob Flickenger (O'Reilly).
- *Practical Unix & Internet Security*, 3rd. ed., by Simson Garfinkel, Gene Spafford, and Alan Schwartz (O'Reilly).
- *Red Hat Linux Firewalls*, by Bill McCarty (Red Hat Press).

If a sufficiently skilled hacker is intent on compromising a system you administer, the hacker will probably succeed. However, here are some tips that can help you avoid falling victim to amateur hackers:

- Use a network or host firewall to prevent outsiders from accessing services you don't need to make publicly available.
- Monitor security web sites and mailing lists so that you're aware of recent threats and the associated countermeasures. The CERT Coordination Center, <http://www.cert.org>, provides many useful resources.
- Apply bug fixes promptly after Red Hat Network advises you that they are available.

Chapter 12. Advanced Shell Usage and Shell Scripts

Like an MS-DOS prompt window, the Unix shell is a command interpreter that lets you issue and execute commands. By means of the shell, you use and control your system. If you're accustomed to the point-and-click world of graphical user interfaces, you may question the value of learning to use the Linux^[1] shell. Many users initially find the shell cumbersome, and some retreat to the familiar comfort of the graphical user interface (GUI), avoiding the shell whenever possible. However, as this chapter explains, the shell unlocks the true power of Linux.

^[1] Most of the statements in this chapter are true of Linux generally, rather than merely true of Red Hat Enterprise Linux WS and Fedora Core. However, in this chapter, I use the term *Linux* as a concise way of referring to Red Hat Enterprise Linux WS and Fedora Core.

12.1 The Power of the Unix Shell

While it's true that the shell is an older style of interacting with a computer than the GUI, the graphical user interface is actually the more primitive interface. The GUI is easy to learn and widely used, but the shell is vastly more sophisticated. Using a GUI is somewhat like communicating in American Indian sign language. If your message is a simple one, like "We come in peace," you can communicate it by using a few gestures. However, if you attempted to give Lincoln's Gettysburg address—a notably brief public discourse—you'd find your task quite formidable.^[2]

^[2] American Sign Language, used to communicate with those who have a hearing impairment, is a much richer language than American Indian sign language.

The designer of a program that provides a GUI must anticipate all the possible ways in which the user will interact with the program and provide ways to trigger the appropriate program responses by means of pointing and clicking. Consequently, the user is constrained to working only in predicted ways. The user is therefore unable to adapt the GUI program to accommodate unforeseen tasks and circumstances. In a nutshell, that's why

many system administration tasks are performed using the shell: system administrators, in fulfilling their responsibility to keep a system up and running, must continually deal with and overcome the unforeseen.

The shell reflects the underlying philosophy of Unix, which provides a wide variety of small, simple tools (that is, programs), each performing a single task. When a complex operation is needed, the tools work together to accomplish the complex operation as a series of simple operations, one step at a time. Many Unix tools manipulate text, and since Unix stores its configuration data in text form rather than in binary form, the tools are ideally suited for manipulating Unix itself. The shell's ability to freely combine tools in novel ways is what makes Unix powerful and sophisticated. Moreover, as you'll learn, the shell is extensible: you can create *shell scripts* that let you store a series of commands for later execution, saving you the future tedium of typing or pointing and clicking to recall them.

The contrary philosophy is seen in operating systems such as Microsoft Windows, which employ elaborate, monolithic programs that provide menus, submenus, and dialog boxes. Such programs have no way to cooperate with one another to accomplish complex operations that weren't anticipated when the programs were designed. They're easy to use so long as you remain on the beaten path, but once you step off the trail, you might find yourself in a confusing wilderness.

Of course, not everyone shares this perspective. The Usenet newsgroups, for example, are filled with postings debating the relative merits of GUIs. Some see the Unix shell as an arcane and intimidating monstrosity. But, even if they're correct, it's inarguable that when you learn to use the shell, you begin to see Unix as it was intended (whether that's for better or for worse).

When you are performing common, routine operations, a GUI that minimizes typing can be a relief, but when faced with a complex, unstructured problem that requires a creative solution, the shell is more often the tool of choice. Creating solutions in the form of shell scripts allows solutions to be stored for subsequent reuse. Perhaps even more important, shell scripts can be studied to quickly bone up on forgotten details, expediting the solution of related problems.

12.2 Filename Globbing

Before the shell passes arguments to an external command or interprets a built-in command, it scans the command line for certain special characters and performs an operation known as *filename globbing*. Filename globbing resembles the processing of wildcards used in MS-DOS commands, but it's much more sophisticated. [Table 12-1](#) describes the special characters, known as filename metacharacters, used in filename globbing.

Table 12-1. Filename metacharacters

Metacharacter	Function
*	Matches a string of zero or more characters
?	Matches exactly one character

Table 12-1. Filename metacharacters	
Metacharacter	Function
[abc...]	Matches any of the characters specified
[a-z]	Matches any character in the specified range
[!abc...]	Matches any character other than those specified
[!a-z]	Matches any character not in the specified range
~	The home directory of the current user
~userid	The home directory of the specified user
~+	The current working directory
~-	The previous working directory

In filename globbing, just as in MS-DOS wildcarding, the shell attempts to replace metacharacters appearing in arguments in such a way that arguments specify filenames. Filename globbing makes it easier to specify names of files and sets of files. For example, suppose the current working directory contains the files *file1*, *file2*, *file3*, and *file04*. Suppose you want to know the size of each file. The following command reports that information:

```
ls -l file1 file2 file3 file04
```

However, the following command reports the same information and is much easier to type:

```
ls -l file*
```

As [Table 12-1](#) shows, the `*` filename metacharacter can match any string of characters.

Suppose you issued the following command:

```
ls -l file?
```

The `?` filename metacharacter can match only a single character. Therefore, *file04* would not appear in the output of the command.

Similarly, the command:

```
ls -l file[2-3]
```

would report only *file2* and *file3*, because only these files have names that match the specified pattern, which requires that the last character of the filename be in the range 2-3. You can use more than one metacharacter in a single argument. For example, consider the following command:

```
ls -l file??
```

This command will list *file04*, because each metacharacter matches exactly one filename character.

Most commands let you specify multiple arguments. If no files match a given argument, the command ignores the argument. Here's another command that reports all four files:

```
ls -l file0* file[1-3]
```




Suppose that a command has one or more arguments that include one or more metacharacters. If none of the arguments matches any filenames, the shell passes the arguments to the program with the metacharacters intact. When the program expects a valid filename, an unexpected error may result.

The tilde (~) metacharacter lets you easily refer to your home directory. For example, the following command:

```
ls ~
```

would list the files in your home directory.

Filename metacharacters don't merely save you typing. They let you write scripts that selectively process files by name. You'll see how that works later in this chapter.

12.3 Shell Aliases

Shell aliases make it easier to use commands by letting you establish abbreviated command names and by letting you pre-specify common options and arguments for a command. To establish a command alias, issue a command of the form:

```
alias name='command'
```

where *command* specifies the command for which you want to create an alias and *name* specifies the name of the alias. For example, suppose you frequently type the MS-DOS command **dir** when you intend to type the Linux command **ls -l**. You can establish an alias for the **ls -l** command by issuing this command:

```
alias dir='ls -l'
```

Once the alias is established, if you mistakenly type **dir**, you'll get the directory listing you wanted instead of the default output of the **dir** command, which resembles **ls** rather than **ls -l**. If you like, you can establish similar aliases for other commands.

Your default Linux configuration probably defines several aliases on your behalf. To see what they are, issue the command:

```
alias
```

If you're logged in as *root*, you may see the following aliases:

```
alias cp='cp -i'
alias l.='ls -d .* --color=tty'
alias ll='ls -l --color=tty'
alias ls='ls -color=tty'
alias mv='mv -i'
alias rm='rm -i'
alias vi='vim'
alias which='alias | /usr/bin/which -tty-only -read-alias -show-dot -show-tilde'
```

Notice how several commands are self-aliased. For example, the command **rm -i** is aliased as **rm**. The effect is that the **-i** option appears whenever you issue the **rm** command, whether or not you type the option. The **-i** option specifies that the shell will prompt for confirmation before deleting files. This helps avoid accidental deletion of files, which can be particularly hazardous when you're logged in as *root*. The alias ensures that you're prompted for confirmation even if you don't ask to be prompted. If you don't want to be prompted, you can issue a command like:

```
rm -f files
```

where *files* specifies the files to be deleted. The **-f** option has an effect opposite that of the **-i** option; it forces deletion of files without prompting for confirmation. Because the command is aliased, the command actually executed is:

```
rm -i -f files
```

The **-f** option takes precedence over the **-i** option, because it occurs later in the command line.

If you want to remove a command alias, you can issue the **unalias** command:

```
unalias alias
```

where *alias* specifies the alias you want to remove. Aliases last only for the duration of a login session, so you needn't bother to remove them before logging off. If you want an alias to be effective each time you log in, you can use a shell script, which we'll discuss later in the chapter.

12.4 Using Virtual Consoles

You can use a terminal window to issue shell commands. However, you can issue shell commands even when X is not running or available. To do so, you use the Linux virtual console feature.

Linux provides six virtual consoles for interactive use; a seventh virtual console is associated with the graphical user interface. You can use special keystrokes to switch between virtual consoles. The keystroke **Alt-F n** , where n is the number of a virtual console (1-6), causes Linux to display virtual console n . For example, you can display virtual console 2 by typing **Alt-F2**. You can view only a single console at a time, but you can switch rapidly between consoles by using the appropriate keystroke. The keystroke **Alt-F7** causes Linux to enter graphical mode using virtual console 7.

Virtual consoles also have a screensaver feature like that found on Windows. If a virtual console is inactive for an extended period of time, Linux blanks the monitor. To restore the screen without disturbing its contents, simply press the **Shift** key.

12.4.1 Logging In

To log in using a virtual console, respond to the login prompt by typing your user ID and pressing **Enter**. The system prompts you for the password associated with your account.

Type the proper password and press **Enter**. To prevent anyone nearby from learning your password, Linux does not display your password as you enter it. If you suspect you've typed it incorrectly, you can either hit the **Backspace** key a number of times sufficient to delete the characters you've entered and type the password again or simply press **Enter** and start over. If you type the user ID or password incorrectly, Linux displays the message "login incorrect" and prompts you to try again.

When you've successfully logged in, you'll see a command prompt that looks something like this:

```
[bill@home bill]$
```

If you logged in as the root user, you'll see a prompt that contains a hash mark (**#**); if you've logged in as an ordinary user, you'll see a dollar sign (**\$**). The prompt tells you that the Linux *bash* shell is ready to accept your commands.

12.4.2 Logging Out

When you're done using a virtual console, you should log out by typing the command **exit** and pressing **Enter**. When you log out, the system frees memory and other resources that were allocated when you logged in, making those resources available to other users and processes.

When the system logs you out, it immediately displays a login prompt. If you change your mind and want to access the system, you can log in simply by supplying your username and password.

12.5 X and the Shell

You can configure your system to boot into nongraphical mode, if you prefer. If your video adapter is not compatible with X, you have no alternative but to do so. However, some Linux users having X-compatible video adapters prefer to configure their system to boot into nongraphical mode. A simple command lets such users launch an X session whenever they wish.

12.5.1 Configuring a Nongraphical Login

Linux provides several runlevels. Each runlevel has an associated set of services. For instance, runlevel 3 is associated with a text-based login and run level 5 is associated with an X-based, graphical login. Changing runlevels automatically starts and stops services associated with the old and new run levels.

You can determine the current runlevel by issuing the following command:

```
# runlevel
```

The output of the command shows the previous and current runlevels. For example, the output:

```
3 5
```

indicates that the current run level is 5 and that the previous run level was 3.

To change the current runlevel, issue the **init** command. For example, to enter runlevel 3, issue the following command while logged in as *root*:

```
# init 3
```

In response to this command, the system will start and stop services as required to enter runlevel 3.

The */etc/inittab* file specifies the default runlevel, which the system enters when booted. By changing the default run level to 3, you can configure your system to provide a nongraphical login when it boots. To do so, log in as *root* and load the */etc/inittab* file into the *nano* editor by issuing the command:

```
# nano /etc/inittab
```

Find the line that reads:

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

Change the 5 to a 3:

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

Save the file and exit *nano*. The next time you boot your system, it will automatically enter runlevel 3 and provide a nongraphical login screen.

12.5.2 Starting and Stopping X from a Text-Based Login

To start X from a text-based login, type the command:

```
$ startx
```

Your system's screen should briefly go blank and then you should see X's graphical desktop.



If the screen is garbled or remains blank for more than about 30 seconds, your X configuration may be faulty. Immediately turn off your monitor or terminate X by pressing **Ctrl-Alt-Backspace**.

When running X from a text-based login, you can gracefully quit X by merely exiting the desktop. To do so, select Logout from the main menu. To quit X abruptly, press **Ctrl-Alt-Backspace**.

12.6 Shell Scripts

A shell script is simply a file that contains a set of commands to be run by the shell when invoked. By storing commands as a shell script, you make it easy to execute them again and again. As an example, consider a file named *deleter*, which contains the following lines:

```
echo -n Deleting the temporary files...
rm -f *.tmp
echo Done.
```

The **echo** commands simply print text on the console. The **-n** option of the first **echo** command causes omission of the trailing newline character normally written by the **echo** command, so both **echo** commands write their text on a single line. The **rm** command removes all files having names ending in *.tmp* from the current working directory.

You can execute this script by issuing the **sh** command, as follows:

```
$ sh deleter
```



If you invoke the **sh** command without an argument specifying a script file, a new interactive shell is launched. To exit the new shell and return to your previous session, issue the **exit** command.

If the *deleter* file were in a directory other than the current working directory, you'd have to type an absolute path, for example:

```
$ sh /home/bill/deleter
```

You can make it a bit easier to execute the script by changing its access mode to include execute access. To do so, issue the following command:

```
$ chmod 555 deleter
```

This gives you, members of your group, and everyone else the ability to execute the file. To do so, simply type the absolute path of the file, for example:

```
$ /home/bill/deleter
```

If the file is in the current directory, you can issue the following command:

```
$ ./deleter
```

You may wonder why you can't simply issue the command:

```
$ deleter
```

In fact, this still simpler form of the command will work, so long as *deleter* resides in a directory on your search path. You'll learn about the search path later.

Linux includes several standard scripts that are run at various times. [Table 12-2](#) identifies these and gives the time when each is run. You can modify these scripts to operate differently. For example, if you want to establish command aliases that are available whenever you log in, you can use a text editor to add the appropriate lines to the *.profile* file that resides in your */home* directory. Since the name of this file begins with a dot (*.*), the **ls** command won't normally show the file. You must specify the **-a** option in order to see this and other hidden files.

Table 12-2. Special shell scripts

Script	Function
/etc/profile	Executes when the user logs in
~/.bash_profile	Executes when the user logs in
~/.bashrc	Executes when <i>bash</i> is launched
~/.bash_logout	Executes when the user logs out



If you want to modify one of the standard scripts that should reside in your home directory but find that your */home* directory does not contain the indicated file, simply create the file. The next time you log in, log out, or launch *bash* (as appropriate) the shell will execute your script.

12.6.1 Input/Output Redirection and Piping

The shell provides three standard data streams:

stdin

The standard input stream

stdout

The standard output stream

stderr

The standard error stream

By default, most programs read their input from *stdin* and write their output to *stdout*. Because both streams are normally associated with a console, programs behave as you generally want, reading input data from the console keyboard and writing output to the console screen. When a well-behaved program writes an error message, it writes the message to the *stderr* stream, which is also associated with the console by default. Having separate streams for output and error messages presents an important opportunity, as you'll see in a moment.

Although the shell associates the three standard input/output streams with the console by default, you can specify input/output redirectors that, for example, associate an input or output stream with a file. [Table 12-3](#) summarizes the most important input/output redirectors.

Table 12-3. Input/output redirectors

Redirector	Function
>file	Redirects standard output stream to specified file

Table 12-3. Input/output redirectors

Redirector	Function
2>file	Redirects standard error stream to specified file
>>file	Redirects standard output stream to specified file, appending output to the file if the file already exists
2>>file	Redirects standard error stream to specified file, appending output to the file if the file already exists
&>file	Redirects standard output and error streams to the specified file
2>&1	Combines the standard error stream with the standard output stream
<file	Redirects standard input stream from the specified file
<<text	Reads standard input until a line matching text is found, at which point end-of-file is posted
cmd1 cmd2	Takes the standard input of cmd2 from the standard output of cmd1 (also known as the pipe redirector)

To see how redirection works, consider the **wc** command. This command takes a series of filenames as arguments and prints the total number of lines, words, and characters present in the specified files. For example, the command:

```
$ wc /etc/passwd
```

might produce the output:

```
22      26      790 /etc/passwd
```

which indicates that the file */etc/passwd* contains 22 lines, 26 words, and 790 characters. Generally, the output of the command appears on your console. But consider the following command, which includes an output redirector:

```
$ wc /etc/passwd > total
```

If you issue this command, you won't see any console output, because the output is redirected to the file *total*, which the command creates (or overwrites, if the file already exists). If you execute the following commands:

```
$ wc /etc/passwd > total
$ cat total
```

you will see the output of the **wc** command on the console.

Perhaps you can now see the reason for having the separate output streams *stdout* and *stderr*. If the shell provided a single output stream, error messages and output would be mingled. Therefore, if you redirected the output of a program to a file, any error messages would also be redirected to the file. This might make it difficult to notice an error that occurred during program execution. Instead, because the streams are separate, you can choose to redirect only *stdout* to a file. When you do so, error messages sent to *stderr* appear on the console in the usual way. Of course, if you prefer, you can redirect both

stdout and *stderr* to the same file or redirect them to different files. As usual in the Unix world, you can have it your own way.

A simple way of avoiding annoying output is to redirect it to the null device file, */dev/null*. If you redirect the *stderr* stream of a command to */dev/null*, you won't see any error messages the command produces. For example, the **grep** command prints an error message if you invoke it on a directory. So, if you invoke the **grep** command on files in the current directory (*) and the current directory contains subdirectories, you'll see unhelpful error messages. To avoid them, use a command like this one, which searches for files containing the text "localhost":

```
$ grep localhost * 2>/dev/null
```

Just as you can direct the standard output or error stream of a command to a file, you can also redirect a command's standard input stream to a file so the command reads from the file instead of the console. For example, if you issue the **wc** command without arguments, the command reads its input from *stdin*. Type some words and then type the end-of-file character (**Ctrl-D**), and **wc** will report the number of lines, words, and characters you entered. You can tell **wc** to read from a file, rather than the console, by issuing a command like:

```
$ wc </etc/passwd
```

Of course, this isn't the usual way of invoking **wc**. The author of **wc** helpfully provided a command-line argument that lets you specify the file from which **wc** reads. However, by using a redirector, you could read from any desired file even if the author had been less helpful.



Some programs are written to ignore redirectors. For example, when invoked without special options, the **passwd** command expects to read the new password only from the console, not from a file. You can compel such programs to read from a file, but doing so requires techniques more advanced than redirectors.

When you specify no command-line arguments, many Unix programs read their input from *stdin* and write their output to *stdout*. Such programs are called *filters*. Filters can be easily fitted together to perform a series of related operations. The tool for combining filters is the *pipe*, which connects the output of one program to the input of another. For example, consider this command:

```
$ ls ~ | wc -l
```

The command consists of two commands, joined by the pipe redirector (**|**). The first command lists the names of the nonhidden files in the user's home directory, one file per line. The second command invokes **wc** by using the **-l** option, which causes **wc** to print only the total number of lines, rather than printing the total number of lines, words, and characters. The pipe redirector sends the output of the **ls** command to the **wc** command, which counts and prints the number of lines in its input, which happens to be the number of files in the user's home directory.

This is a simple example of the power and sophistication of the Unix shell. Unix doesn't include a command that counts the files in the user's home directory and doesn't need to do

so. Should the need to count the files arise, a knowledgeable Unix user can prepare a simple script that computes the desired result by using general-purpose Unix commands.

12.6.2 Shell Variables

If you've studied programming, you know that programming languages resemble algebra. Both programming languages and algebra let you refer to a value by a name. And both programming languages and algebra include elaborate mechanisms for manipulating named values.

The shell is a programming language in its own right, letting you refer to variables known as *shell* or *environment variables*. To assign a value to a shell variable, you use a command that has the following form:

```
$ variable = value
```

For example, the command:

```
$ DifficultyLevel=1
```

assigns the value 1 to the shell variable named DifficultyLevel. Unlike algebraic variable, shell variables can have nonnumeric values. For example, the command:

```
$ Difficulty=medium
```

assigns the value medium to the shell variable named Difficulty.

Shell variables are widely used within Unix, because they provide a convenient way of transferring values from one command to another. Programs can obtain the value of a shell variable and use the value to modify their operation, in much the same way they use the value of command-line arguments.

You can see a list of shell variables by issuing the **set** command. Usually, the command produces more than a single screen of output. So, you can use a pipe redirector and the **less** command to view the output one screen at a time:

```
$ set | less
```

Press the spacebar to see each successive page of output. You'll probably see several of the shell variables described in [Table 12-4](#) among those printed by the **set** command. The values of these shell variables are generally set by one or another of the startup scripts described earlier in this chapter.

Table 12-4. Important environment variables	
Variable	Function
DISPLAY	The X display to be used; for example, localhost:0
HOME	The absolute path of the user's home directory
HOSTNAME	The Internet name of the host
LOGNAME	The user's login name

Table 12-4. Important environment variables

Variable	Function
MAIL	The absolute path of the user's mail file
PATH	The search path (see Section 12.6.3)
SHELL	The absolute path of the current shell
TERM	The terminal type
USER	The user's current username; may differ from the login name if the user executes the su command

You can use the value of a shell variable in a command by preceding the name of the shell variable by a dollar sign (\$). To avoid confusion with surrounding text, you can enclose the name of the shell variable within curly braces ({ }); it's good practice (though not necessary) to do so consistently. For example, you can change the current working directory to your */home* directory by issuing the command:

```
$ cd ${HOME}
```

Of course, issuing the **cd** command with no argument causes the same result. However, suppose you want to change to the */work* subdirectory of your home directory. The following command accomplishes exactly that:

```
$ cd ${HOME}/work
```

An easy way to see the value of a shell variable is to specify the variable as the argument of the **echo** command. For example, to see the value of the HOME shell variable, issue the command:

```
$ echo ${HOME}
```

To make the value of a shell variable available not just to the shell, but to programs invoked by using the shell, you must export the shell variable. To do so, use the **export** command, which has the form:

```
$ export variable
```

where *variable* specifies the name of the variable to be exported. A shorthand form of the command lets you assign a value to a shell variable and export the variable in a single command:

```
$ export variable = value
```

You can remove the value associated with a shell variable by giving the variable an empty value:

```
$ variable =
```

However, a shell variable with an empty value remains a shell variable and appears in the output of the **set** command. To dispense with a shell variable, you can issue the **unset** command:

```
$ unset variable
```

Once you unset the value of a variable, the variable no longer appears in the output of the **set** command.

12.6.3 The Search Path

The special shell variable **PATH** holds a series of paths known collectively as the *search path*. Whenever you issue an external command, the shell searches the paths that comprise the search path, seeking the program file that corresponds to the command. The startup scripts establish the initial value of the **PATH** shell variable, but you can modify its value to include any desired series of paths. You must use a colon (:) to separate each path of the search path. For example, suppose that **PATH** has the following value:

```
/usr/bin:/bin:/usr/local/bin:/usr/bin/X11:/usr/X11R6/bin
```

You can add a new search directory, say */home/bill*, by issuing the following command:

```
$ PATH=${PATH}:/home/bill
```

Now, the shell will look for external programs in */home/bill* as well as in the default directories. However, the problem is that the shell will look there last. If you prefer to check */home/bill* first, issue the following command instead:

```
$ PATH=/home/bill:${PATH}
```

The **which** command helps you work with the **PATH** shell variable. It checks the search path for the file specified as its argument and prints the name of the matching path, if any. For example, suppose you want to know where the program file for the **wc** command resides. Issuing the command:

```
$ which wc
```

will tell you that the program file is */usr/bin/wc* (or whatever other path is correct for your system).

12.6.4 Quoted Strings

Sometimes the shell may misinterpret a command you've written, globbing a filename or expanding a reference to a shell variable that you hadn't intended. Of course, it's actually your interpretation that's mistaken, not the shell's. Therefore, it's up to you to rewrite your command so the shell's interpretation is congruent with what you intended.

Quote characters, described in [Table 12-5](#), can help you by controlling the operation of the shell. For example, by enclosing a command argument within single quotes, you can prevent the shell from globbing the argument or substituting the argument with the value of a shell variable.

Table 12-5. Quote characters

Character	Function
' (single quote)	Characters within a pair of single quotes are interpreted literally; that is, their metacharacter meanings (if any) are ignored. Similarly, the shell does not replace references to shell or environment variables with the value of the referenced variable.
" (double quote)	Characters within a pair of double quotes are interpreted literally; that is, their metacharacter meanings (if any) are ignored. However, the shell does replace references to shell or environment variables with the value of the referenced variable.
` (backquote)	Text within a pair of backquotes is interpreted as a command, which the shell executes before executing the rest of the command line. The output of the command replaces the original backquoted text.
\ (backslash)	The following character is interpreted literally; that is, its metacharacter meaning (if any) is ignored. The backslash character has a special use as a line continuation character. When a line ends with a backslash, the line and the following line are considered part of a single line.

To see quoted characters in action, consider how you might cause the **echo** command to produce the output `$PATH`. If you simply issue the command:

```
$ echo $PATH
```

the **echo** command prints the value of the `PATH` shell variable. However, by enclosing the argument within single quotes, you obtain the desired result:

```
$ echo '$PATH'
```

Double quotes have a similar effect. They prevent the shell from globbing a filename but permit the expansion of shell variables.

Backquotes operate differently; they let you execute a command and use its output as an argument of another command. For example, the command:

```
$ echo My home directory contains `ls ~ | wc -l` files.
```

prints a message that gives the number of files in the user's `/home` directory. The command works by first executing the command contained within backquotes:

```
$ ls ~ | wc -l
```

This command, as explained earlier, computes and prints the number of files in the user's directory. Because the command is enclosed in backquotes, its output is not printed; instead the output replaces the original backquoted text.

The resulting command becomes:

```
echo My home directory contains 22 files.
```

When executed, this command prints the output:

```
My home directory contains 22 files.
```

You may now begin to appreciate the power of the Linux shell: by including command aliases in your *bashrc* script, you can extend the command repertoire of the shell. And, by using filename completion and the history list, you can reduce the amount of typing it takes to enter frequently used commands. Once you grasp how to use it properly, the Linux shell is a powerful, fast, and easy-to-use interface that avoids the limitations and monotony of the more familiar point-and-click graphical interface.

But the shell has additional features that extend its capabilities even further. As you'll see in the next section, the Linux shell includes a powerful programming language that provides argument processing, conditional logic, and loops.

12.7 Understanding Shell Scripts

This section explains how more advanced shell scripts work. The information is also adequate to equip you to write many of your own useful shell scripts. The section begins by showing how to process a script's arguments. Then it shows how to perform conditional and iterative operations.

12.7.1 Processing Arguments

You can easily write scripts that process arguments, because a set of special shell variables holds the values of arguments specified when your script is invoked. [Table 12-6](#) describes the most popular such shell variables.

Table 12-6. Special shell variables used in scripts	
Variable	Meaning
<code>\$#</code>	The number of arguments
<code>\$0</code>	The command name
<code>\$1, \$2, ... , \$9</code>	The individual arguments of the command
<code>\$*</code>	The entire list of arguments, treated as a single word
<code>@</code>	The entire list of arguments, treated as a series of words
<code>?</code>	The exit status of the previous command; a value of 0 denotes a successful completion
<code>\$\$</code>	The ID of the current process

For example, here's a simple one-line script that prints the value of its second argument:

```
echo My second argument has the value $2.
```

Suppose you store this script in the file *second*, change its access mode to permit execution, and invoke it as follows:

```
$ ./second a b c
```

The script will print the output:

```
My second argument has the value b.
```

Notice that the shell provides variables for accessing only nine arguments. Nevertheless, you can access more than nine arguments. The key to doing so is the **shift** command, which discards the value of the first argument and shifts the remaining values down one position. Thus, after executing the **shift** command, the shell variable `$9` contains the value of the 10th argument. To access the 11th and subsequent arguments, you simply execute the **shift** command the appropriate number of times.

12.7.2 Exit Codes

The shell variable `?` holds the numeric exit status of the most recently completed command. By convention, an exit status of zero denotes successful completion; other values denote error conditions of various sorts. You can set the error code in a script by issuing the **exit** command, which terminates the script and posts the specified exit status.

The format of the command is:

```
$ exit status
```

where *status* is a nonnegative integer that specifies the exit status.

12.7.3 Conditional Logic

A shell script can employ conditional logic, which lets the script take different action based on the values of arguments, shell variables, or other conditions. The **test** command lets you specify a condition, which can be either true or false. Conditional commands (including the **if**, **case**, **while**, and **until** commands) use the **test** command to evaluate conditions.

12.7.3.1 The test command

[Table 12-7](#) describes some argument forms commonly used with the **test** command. The **test** command evaluates its arguments and sets the exit status to zero, which indicates that the specified condition was true, or a nonzero value, which indicates that the specified condition was false.

Table 12-7. Commonly used argument forms of the test command	
Form	Function
-d <i>file</i>	The specified file exists and is a directory.
-e <i>file</i>	The specified file exists.

Table 12-7. Commonly used argument forms of the test command

Form	Function
<code>-r file</code>	The specified file exists and is readable.
<code>-s file</code>	The specified file exists and has nonzero size.
<code>-w file</code>	The specified file exists and is writable.
<code>-x file</code>	The specified file exists and is executable.
<code>-L file</code>	The specified file exists and is a symbolic link.
<code>f1 -nt f2</code>	File <i>f1</i> is newer than file <i>f2</i> .
<code>f1 -ot f2</code>	File <i>f1</i> is older than file <i>f2</i> .
<code>-n s1</code>	String <i>s1</i> has nonzero length.
<code>-z s1</code>	String <i>s1</i> has zero length.
<code>s1 = s2</code>	String <i>s1</i> is the same as string <i>s2</i> .
<code>s1 != s2</code>	String <i>s1</i> is not the same as string <i>s2</i> .
<code>n1 -eq n2</code>	Integer <i>n1</i> is equal to integer <i>n2</i> .
<code>n1 -ge n2</code>	Integer <i>n1</i> is greater than or equal to integer <i>n2</i> .
<code>n1 -gt n2</code>	Integer <i>n1</i> is greater than integer <i>n2</i> .
<code>n1 -le n2</code>	Integer <i>n1</i> is less than or equal to integer <i>n2</i> .
<code>n1 -lt n2</code>	Integer <i>n1</i> is less than integer <i>n2</i> .
<code>n1 -ne n2</code>	Integer <i>n1</i> is not equal to integer <i>n2</i> .
<code>!</code>	The NOT operator, which reverses the value of the following condition.
<code>-a</code>	The AND operator, which joins two conditions. Both conditions must be true for the overall result to be true.
<code>-o</code>	The OR operator, which joins two conditions. If either condition is true, the overall result is true.
<code>\(... \)</code>	You can group expressions within the test command by enclosing them within <code>\(</code> and <code>\)</code> .

To see the **test** command in action, consider the following script:

```
test -d $1
echo $?
```

This script tests whether its first argument specifies a directory and displays the resulting exit status, a zero or a nonzero value that reflects the result of the test.

If the script was stored in the file *tester*, which permitted execute access, executing the script might yield results similar to the following:

```
$ ./tester /
0
$ ./tester /missing
1
```

These results indicate that the root directory (/) exists and that the */missing* directory does not.

12.7.3.2 The if command

The **test** command is not of much use by itself, but combined with commands such as the **if** command, it is useful indeed. The **if** command has the following form:

```
if command
then
    commands
else
    commands
fi
```

The command that usually follows **if** is a **test** command. However, this need not be so. The **if** command merely executes the specified command and tests its exit status. If the exit status is zero, the first set of commands is executed; otherwise, the second set of commands is executed. An abbreviated form of the **if** command does nothing if the specified condition is false:

```
if command
then
    commands
fi
```

When you type an **if** command, it occupies several lines; nevertheless, it's considered a single command. To underscore this, the shell provides a special prompt, called the *secondary prompt*, after you enter each line. You won't see the secondary prompt when entering a script using a text editor, or any other shell prompt for that matter.

As an example, suppose you want to delete a file, *file1*, if it's older than another file, *file2*.

The following command would accomplish the desired result:

```
if test file1 -ot file2
then
    rm file1
fi
```

You could incorporate this command in a script that accepts arguments specifying the filenames:

```
if test $1 -ot $2
then
    rm $1
    echo Deleted the old file.
fi
```

If you name the script *riddance* and invoke it as follows:

```
$ riddance thursday wednesday
```

the script will delete the *thursday* file if that file is older than the *wednesday* file.

12.7.3.3 The case command

The **case** command provides a more sophisticated form of conditional processing:

```
case value in
  pattern1 ) commands ;;
  pattern2 ) commands ;;
  ...
esac
```

The **case** command attempts to match the specified value against a series of patterns. The commands associated with the first matching pattern, if any, are executed. Patterns are built using characters and metacharacters, such as those used to specify command arguments. As an example, here's a **case** command that interprets the value of the first argument of its script:

```
case $1 in
  -r) echo Force deletion without confirmation ;;
  -i) echo Confirm before deleting ;;
  *)  echo Unknown argument ;;
esac
```

The command echoes a different line of text, depending on the value of the script's first argument. As done here, it's good practice to include a final pattern that matches any value.

12.7.3.4 The while command

The **while** command lets you execute a series of commands iteratively (that is, repeatedly) so long as a condition tests true:

```
while command
do
  commands
done
```

Here's a script that uses a **while** command to print its arguments on successive lines:

```
echo $1
while shift 2> /dev/null
do
  echo $1
done
```

Notice how the **2>** operator is used to direct error messages to the device */dev/null*, which prevents them from being seen. You can omit this operator if you prefer.

The commands that comprise the **do** part of a **while** (or any other loop command) can include **if**, **case**, and even other **while** commands. However, scripts rapidly become difficult to understand when this occurs often. You should include conditional commands within other conditional commands only with due consideration for the clarity of the result. Don't

forget to include comments in your scripts (with each commented line beginning with a #) to clarify difficult constructs.

12.7.3.5 The until command

The **until** command lets you execute a series of commands iteratively (that is, repeatedly) so long as a condition tests false:

```
until  command
do
    commands
done
```

Here's a script that uses an **until** command to print its arguments on successive lines, until it encounters an argument that has the value *red*:

```
until test $1 = red
do
    echo $1
    shift
done
```

For example, if the script were named *stopandgo* and stored in the current working directory, the command:

```
$ ./stopandgo green yellow red blue
```

would print the lines:

```
green
yellow
```

12.7.3.6 The for command

The **for** command iterates over the elements of a specified list:

```
for  variable in list
do
    commands
done
```

Within the commands, you can reference the current element of the list by means of the shell variable *\$variable*, where *variable* is the name specified following the **for**. The list typically takes the form of a series of arguments, which can incorporate metacharacters. For example, the following **for** command:

```
for i in 2 4 6 8
do
    echo $i
done
```

prints the numbers 2, 4, 6, and 8 on successive lines.

A special form of the **for** command iterates over the arguments of a script:

```
for  variable
do
    commands
```

done

For example, the following script prints its arguments on successive lines:

```
for i
do
    echo $i
done
```

12.7.3.7 The break and continue commands

The **break** and **continue** commands are simple commands that take no arguments. When the shell encounters a **break** command, it immediately exits the body of the enclosing loop (**while**, **until**, or **for**) command. When the shell encounters a **continue** command, it immediately discontinues the current iteration of the loop. If the loop condition permits, other iterations may occur; otherwise the loop is exited.

12.7.4 Periscope: A Useful Networking Script

Suppose you have a free email account such as that provided by Yahoo! You're traveling and find yourself in a remote location with web access. However, you're unable to access files on your home machine or check email that has arrived there, because your home machine connects to the Internet via a dial-up connection rather than a cable modem/DSL connection that is continually active. This is a common circumstance, especially if your business requires that you travel.

If your home computer runs Windows, you're pretty much out of luck. You'll find it difficult to access your home computer from afar. However, if your home computer runs Linux, gaining access is practically a piece of cake.

In order to show the power of shell scripts, this subsection explains a more complex shell script, *periscope*. At an appointed time each day, *periscope* causes your computer (which you must leave powered on) to establish a PPP connection to your ISP, which is maintained for about one hour. This provides you enough time to connect to an ISP from your hotel room or other remote location and then connect via the Internet with your home Linux system, avoiding long-distance charges. Once connected, you have about an hour to view or download mail and perform other work. Then, *periscope* breaks its PPP connection, which it will reestablish at the appointed time the next day.

The following code shows the *periscope* script file, which is considerably larger than any script you've so far encountered in this chapter. Therefore, we'll disassemble the script, explaining it line by line. As you'll see, each line is fairly simple in itself, and the lines work together in a straightforward fashion.

```
1 route del default
2 wvdial &
3 sleep 1m
4 ifconfig | mail username@mail.com
5 sleep 1h
6 killall wvdial
7 sleep 2s
8 killall -9 wvdial
9 killall pppd
```

```
10 sleep 2s
11 killall -9 pppd
12 echo "/root/periscope" | at 10:00
```

Here's the line-by-line analysis of the *periscope* script:

1. `route del default`

This line is perhaps the most complex line of the entire script. The **route** command is normally issued by the system administrator. You've probably never issued the command yourself, because **neat** or another network configuration program has issued it on your behalf. The effect of the command is to delete the default network route, if any. The default route is the one along which TCP/IP sends packets when it knows no specific route to their specified destination. It's necessary to delete the default route because the **wvdial** program, which the script uses to establish its PPP connection, will not override an existing default route.

2. `wvdial &`

This line launches the **wvdial** program. As specified by the ampersand (&), the program will run in the background, so the script continues executing while **wvdial** starts up and runs.

3. `sleep 1m`

This line pauses the script for one minute, giving **wvdial** time to establish the PPP connection.

4. `ifconfig | mail username@mail.com`

This line runs the **ifconfig** command and mails its output to the specified user (you must replace *username@mail.com* with your own email address, which you can access remotely).

The **ifconfig** command produces output that looks something like this:

```
ppp0    Link encap:Point-Point Protocol
        inet addr:10.144.153.105  P-t-P:10.144.153.52
Mask:255.255.255.0
        UP POINTOPOINT RUNNING  MTU:552  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0
        TX packets:0 errors:0 dropped:0 overruns:0
```

You'll probably see other sections in the output that describe your Ethernet interface (eth0) and a loopback device (lo). The `inet addr` given in the command output (10.144.153.105) is the current IP address of your computer. By mailing the

output to yourself, you provide a simple way to discover your computer's IP address, which is likely to be different each time it connects to your ISP.

5. `sleep 1h`

This line causes the script to pause for an interval of one hour. You can easily change this interval to something more appropriate to your own needs.

6. `killall wvdial`

Now that the connection interval has elapsed, the last line terminates all executing instances of the **wvdial** program.

[Appendix D](#) briefly describes the **killall** command and other possibly unfamiliar commands employed in this script.

7. `sleep 2s`

The script then pauses for two seconds, to ensure that **wvdial** has completely terminated.

8. `killall -9 wvdial`

Under some circumstances, a program will ignore a termination request. This line deals with this possibility by sending a special code that compels a reluctant program to terminate without further delay.

9. `killall pppd`

Behind the scenes, **wvdial** launches a program known as **pppd**, which actually establishes and manages the PPP connection. Another **killall** command is designed to terminate **pppd** if **wvdial** has failed to do so.

10. `sleep 2s`

Again, the script pauses for a few seconds. This time, it does so to ensure that **pppd** has completely terminated.

11. `killall -9 pppd`

And, again, the script uses the **-9** option to specify that any remaining instances of **pppd** should terminate immediately.

12. `echo "/root/periscope" | at 10:00`

Finally, the script uses the **at** command to schedule itself for execution at 10:00 tomorrow. The **at** command reads one or more commands from its standard input and executes them at the time specified as an argument.

To try the script for yourself, you must have installed the **wvdial** program, as explained in [Chapter 10](#). Place the script in the file `/root/periscope`. Of course, you'll probably want to customize the script to specify an appointment time and duration of your own choosing. To start *periscope*, log in as *root* and issue the command:

```
# (echo "/root/periscope" | at 10:00) &
```

The parentheses cause the **&** operator to apply to the entire command, not just the **at**. When 10:00 a.m. (or any other time you specify) comes around, your Linux system should obediently dial your ISP and maintain the connection for the specified interval of time.

12.7.5 Using Periscope

At the appointed time, fire up your computer and access your email account. You should find a mail message that contains the **ifconfig** output giving your computer's current IP address. Now you can use *telnet* or an *ssh* client—your choice corresponds to the server you're running on your Linux system—to contact your computer and work for the remainder of the specified connection time. At the end of the connection time, your Linux system will sever its PPP connection and begin counting down until it's again time to connect.

Appendix A. Linux Directory Tree

[Table A-1](#) describes the directories in the Linux directory tree.

Table A-1. The Linux directory tree	
Directory	Description
<code>/bin</code>	User programs and scripts essential to system startup
<code>/boot</code>	Boot information, including the kernel
<code>/dev</code>	Device files
<code>/etc</code>	Host-specific configuration files
<code>/etc/sysconfig</code>	Stores configuration files specific to Red Hat Linux
<code>/home</code>	Users' home directories
<code>/initrd</code>	Used during boot process as a mount point for a directory containing special device drivers
<code>/lib</code>	Libraries, modules, and other object files
<code>/lib/modules</code>	Loadable kernel modules

Table A-1. The Linux directory tree

Directory	Description
<i>/lost+found</i>	Recovered data from bad clusters
<i>/mnt</i>	Temporarily mounted filesystems
<i>/opt</i>	Used to store large applications
<i>/proc</i>	Kernel pseudo-directory that provides access to kernel information and configuration items
<i>/root</i>	System administrator's home directory
<i>/sbin</i>	System administration programs and scripts essential to system startup
<i>/tmp</i>	Temporary files, which are automatically deleted by Red Hat Linux
<i>/usr</i>	Files needed for system operation but not needed to boot system (can be mounted read-only, except when being changed by root user)
<i>/usr/bin</i>	Programs and scripts not essential to system startup
<i>/usr/dict</i>	System dictionaries for spell checking
<i>/usr/etc</i>	Configuration files
<i>/usr/games</i>	Game files
<i>/usr/include</i>	C/C++ header files
<i>/usr/kerberos</i>	Kerberos files
<i>/usr/lib</i>	Libraries and kernel modules
<i>/usr/libexec</i>	Libraries stored in alternate formats
<i>/usr/local</i>	Locally defined directory tree (structure is similar to that of <i>/usr</i>)
<i>/usr/sbin</i>	System administration programs and scripts not essential to system start up
<i>/usr/share</i>	Shared files
<i>/usr/share/doc</i>	Documentation (formerly residing in <i>/usr/doc</i>)
<i>/usr/share/man</i>	Manpages
<i>/usr/src</i>	Source files
<i>/usr/src/linux</i>	Linux kernel source
<i>/usr/X11R6</i>	X-related files
<i>/var</i>	Dynamic files, such as log files and spool files
<i>/var/cache</i>	Application cache data

Table A-1. The Linux directory tree	
Directory	Description
<i>/var/ftp</i>	Data used by the FTP server
<i>/var/gdm</i>	Data used by the GNOME display manager
<i>/var/lib</i>	Variable state information
<i>/var/lib/rpm</i>	Files related to RPM
<i>/var/local</i>	Locally defined data
<i>/var/lock</i>	Lock files
<i>/var/log</i>	Log files and directories
<i>/var/named</i>	Data used by BIND (named)
<i>/var/opt</i>	Data used by applications in <i>/opt</i>
<i>/var/run</i>	Process IDs of running processes
<i>/var/spool</i>	Application spool data
<i>/var/spool/at</i>	At's spool data
<i>/var/spool/cron</i>	Cron's spool data
<i>/var/spool/cups</i>	CUPS printing service data.
<i>/var/spool/lpd</i>	Printer queues
<i>/var/spool/mail</i>	Mail boxes
<i>/var/spool/mqueue</i>	Mail queue
<i>/var/spool/samba</i>	Samba spool data
<i>/var/spool/up2date</i>	Files used by Red Hat Update Agent
<i>/var/tmp</i>	Temporary files preserved between system reboots
<i>/var/www</i>	Files used by Apache
<i>/var/yp</i>	Files used by NIS

For more information on the Linux directory tree, see the current version of the Linux Filesystem Hierarchy Standard (<http://www.pathname.com/fhs>). Red Hat Linux generally complies with that standard.

Appendix B. Principal Linux Files

[Table B-1](#) describes the principal Linux files. You can use it, for example, to help you locate configuration files quickly.

Table B-1. Principal Linux files	
File(s)	Description
<i>/boot/grub/grub.conf</i>	GRUB configuration file
<i>/boot/module-info-*</i>	Module information for the Linux kernel
<i>/boot/System.map-*</i>	Map of the Linux kernel
<i>/boot/vmlinuz-*</i>	Linux kernel
<i>/etc/aliases</i>	Mail aliases
<i>/etc/at.deny</i>	User IDs of users forbidden to use the at command
<i>/etc/auto.master</i>	Configuration file for the <i>autofs</i> daemon, which automatically mounts filesystems
<i>/etc/auto.misc</i>	Automounter map file
<i>/etc/bashrc</i>	Systemwide functions and aliases for the <i>bash</i> shell
<i>/etc/cron.daily/*</i>	Daily <i>cron</i> jobs
<i>/etc/cron.hourly/*</i>	Hourly <i>cron</i> jobs
<i>/etc/cron.monthly/*</i>	Monthly <i>cron</i> jobs
<i>/etc/cron.weekly/*</i>	Weekly <i>cron</i> jobs
<i>/etc/crontab</i>	System <i>cron</i> file
<i>/etc/cups/*</i>	Printer configuration files
<i>/etc/default/useradd</i>	Defaults for the useradd command
<i>/etc/DIR_COLORS</i>	Directory listing colors
<i>/etc/exports</i>	NFS exported directories
<i>/etc/filesystems</i>	Supported filesystem types
<i>/etc/fstab</i>	Filesystems mounted or available for mounting
<i>/etc/group</i>	System group definitions
<i>/etc/host.conf</i>	Resolver configuration file
<i>/etc/hosts</i>	Map of IP numbers to hostnames
<i>/etc/hosts.allow</i>	Hosts allowed to access Internet services

Table B-1. Principal Linux files

File(s)	Description
<i>/etc/hosts.deny</i>	Hosts forbidden to access Internet services
<i>/etc/httpd/conf/*</i>	Apache configuration files
<i>/etc/httpd/httpd.conf</i>	Main Apache configuration file
<i>/etc/init.d/*</i>	SysV initialization scripts
<i>/etc/initlog.conf</i>	Logging configuration file
<i>/etc/inittab</i>	Configuration for the <i>init</i> daemon, which controls executing processes
<i>/etc/issue</i>	Linux kernel and distribution version (local users)
<i>/etc/issue/net</i>	Linux kernel and distribution version (remote users)
<i>/etc/ld.so.conf</i>	Shared library configuration file
<i>/etc/login.defs</i>	Options for useradd and related commands
<i>/etc/logrotate.conf</i>	Log rotation configuration file
<i>/etc/logrotate.d/*</i>	Scripts to rotate logs
<i>/etc/mail/*</i>	Mail server configuration files
<i>/etc/mailcap</i>	<i>metamail</i> MIME information
<i>/etc/man.config</i>	<i>man</i> configuration file
<i>/etc/mime.types</i>	MIME types
<i>/etc/mime-magic*</i>	Magic numbers for MIME data
<i>/etc/minicom.users</i>	User IDs allowed to use <i>minicom</i>
<i>/etc/modules.conf</i>	Aliases and options for loadable kernel modules
<i>/etc/motd</i>	Message of the day
<i>/etc/mtab</i>	Mounted filesystems
<i>/etc/nsswitch.conf</i>	Resolver configuration file
<i>/etc/openldap/*</i>	Open LDAP configuration files
<i>/etc/pam.d/*</i>	PAM configuration files
<i>/etc/paper.config</i>	Paper sizes
<i>/etc/passwd</i>	User account information
<i>/etc/ppp/*</i>	PPP configuration

Table B-1. Principal Linux files

File(s)	Description
<i>/etc/printcap</i>	Printer options and capabilities
<i>/etc/profile</i>	Default environment for users of the <i>bash</i> shell
<i>/etc/profile.d/*</i>	Shell initialization
<i>/etc/protocols</i>	Protocol names and numbers
<i>/etc/pwdb.conf</i>	<i>pwdb</i> library configuration
<i>/etc/rc</i>	Scripts for system and process startup and shutdown
<i>/etc/rc.local</i>	Local startup script
<i>/etc/rc.sysinit</i>	System initialization file
<i>/etc/rc?.d/*</i>	Service start/stop scripts
<i>/etc/rpc</i>	RPC program number database
<i>/etc/rpm/*</i>	RPM database and configuration files
<i>/etc/samba/*</i>	Samba configuration files
<i>/etc/securetty</i>	Secure tty configuration
<i>/etc/security/*</i>	PAM configuration files
<i>/etc/sensors.conf</i>	<i>libsensors</i> configuration file
<i>/etc/services</i>	Standard service names and numbers
<i>/etc/shadow</i>	Secure user account information
<i>/etc/skel</i>	Skeleton files used to establish new user accounts
<i>/etc/ssh/*</i>	SSH configuration files
<i>/etc/sysconfig/*</i>	System configuration files
<i>/etc/sysconfig/network-scripts/*</i>	Network adapter configuration files
<i>/etc/sysctl.conf</i>	<i>sysctl</i> configuration file
<i>/etc/syslog.conf</i>	System logging process configuration
<i>/etc/termcap</i>	Terminal capabilities and options
<i>/etc/updatedb.conf</i>	<i>updatedb/locate</i> configuration file
<i>/etc/wvdial.conf</i>	GNOME dialer configuration file
<i>/etc/X11/applnk/*</i>	X application shortcuts
<i>/etc/X11/fs/config</i>	X font server configuration

Table B-1. Principal Linux files

File(s)	Description
<i>/etc/X11/gdm/*</i>	GNOME display manager configuration
<i>/etc/X11/prefdm</i>	Display manager configuration file
<i>/etc/X11/xdm/*</i>	X display manager configuration file
<i>/etc/X11/XF86Config</i>	X configuration file
<i>/etc/X11/xinit/Xclients</i>	Default script for <i>xinit</i>
<i>/etc/X11/xinit/xinitrc</i>	X session initialization file
<i>/etc/X11/Xmodmap</i>	Key mappings used by <i>xdm</i> and <i>xinit</i>
<i>/etc/xinetd.conf</i>	General <i>xinetd</i> configuration file
<i>/etc/xinetd.d/*</i>	<i>xinetd</i> configuration files for specific servers
<i>/home/*/public_html</i>	User web pages
<i>/root/.bash_history</i>	<i>bash</i> command history for system administrator
<i>/root/.bash_logout</i>	<i>bash</i> logout script for system administrator
<i>/root/.bash_profile</i>	<i>bash</i> initialization script for system administrator
<i>/root/.bashrc</i>	<i>bash</i> options for system administrator
<i>/root/.Xresources</i>	X resources for system administrator
<i>/usr/share/config/*</i>	Miscellaneous configuration files
<i>/usr/share/fonts/*</i>	Fonts
<i>/usr/share/ssl/openssl.cnf</i>	SSL certificate configuration
<i>/usr/X11R6/lib/X11/app-defaults/*</i>	X application defaults
<i>/usr/X11R6/lib/X11/fonts/*</i>	X fonts
<i>/var/log/cron</i>	Log of <i>cron</i> activity
<i>/var/log/httpd/access_log</i>	Log of web server access
<i>/var/log/httpd/error_log</i>	Log of web server errors
<i>/var/log/boot.log</i>	Boot messages
<i>/var/log/cron</i>	<i>Cron</i> log
<i>/var/log/dmesg</i>	Kernel message log
<i>/var/log/lastlog</i>	Last login log

Table B-1. Principal Linux files

File(s)	Description
<code>/var/log/maillog</code>	Mail transfer log
<code>/var/log/messages</code>	System log
<code>/var/log/samba/*</code>	Samba logs
<code>/var/log/secure</code>	System security log
<code>/var/log/up2date</code>	Up2date log
<code>/var/www/cgi-bin</code>	CGI scripts
<code>/var/www/html/*</code>	Web pages

 [Print](#)  [Download](#)  [E-Mail](#)  [Add Note](#)  [Add Bookmark](#)

[< Previous](#) [Next >](#)



[Learning Red Hat Enterprise Linux & Fedora, 4th Edition](#)

By Bill McCarty

Slots : 1.0

[Table of Contents](#)

Appendix C. Managing the Boot Process

In this appendix, you'll learn more about how to boot a Linux system; in particular, you'll learn more about configuring your computer system to boot any of several operating systems. The appendix focuses on GNU GRUB, the Grand Unified Bootloader, the most popular utility for booting Linux systems. Most users won't need to use the information in this appendix. But, this information can help you troubleshoot and resolve boot-related problems.



bootloaders other than GRUB, such as LILO. Occasionally, LILO is able to boot a system that can't be booted using GRUB. To learn more about LILO, see the related section of the *Red Hat Linux 9 Reference Guide*, available at

[http://www.redhat.com/docs/manuals/enterprise/Red_Hat_Linux_9_Reference_Guide/](#)

<http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/ref-guide/s1-grub-lilo.html>.

C.1 Booting Linux

When you boot a PC, you cause it to execute a small program known as a *boot loader*. The purpose of the boot loader is to locate and read into memory the first stage of an operating system and transfer control to it. The operating system then locates and reads its remaining components as needed.

The simplest way to boot Linux is by using a floppy diskette. By doing so, you're able to leave the boot information on your hard drive untouched, ensuring that you can still boot Microsoft Windows or another operating system stored on the same hard drive. Moreover, some operating systems and virus protection programs prevent modification of the boot information on your hard drive. By booting from a floppy diskette, you avoid several potential problems.

However, many users find booting from a floppy disk slow or inconvenient. You don't have to boot Linux from a floppy diskette; you can boot it in any of several other ways. The most popular method is by using GRUB, which replaces the boot loader stored on your hard drive.

This appendix cannot describe the entire range of issues involved in booting Linux. Much of the information here is taken from several Linux HOWTOs that contain additional useful information on booting Linux:

- *BootPrompt-HOWTO*
- *CD-Writing-HOWTO*
- *CDROM-HOWTO*
- *Ethernet-HOWTO*
- *Ftape-HOWTO*
- *Hardware-HOWTO*
- *Multi-Disk-HOWTO*
- *PCI-HOWTO*
- *PCMCIA-HOWTO*

C.2 Boot Floppies

Even if you don't want to boot Linux from a floppy diskette, you should create and keep on hand a Linux boot floppy. If something goes wrong with your system, preventing you from booting in the normal way, you may be able to boot your system by using the floppy. Then, you can diagnose and repair the problem and get back to business as usual.

C.2.1 Creating a Boot Floppy

You can easily create a boot diskette that can be used to start your system. To do so, insert a blank floppy diskette into your system's floppy drive. Log on as *root* and issue the following command:

```
# /sbin/mkbootdisk version
```

For *version*, supply the version number of your kernel. If you don't recall the version, execute the following command, which reports it:

```
# uname -a
```

The version number resembles the version number of an RPM package. For example, a typical kernel version number is 2.4.18-14.

The **mkbootdisk** command creates a boot floppy that uses the same kernel running when the command is issued. It also configures the boot floppy to load any necessary SCSI modules, so that your SCSI drives will be accessible after booting from the floppy.

C.2.2 Using a Boot Floppy

Insert the boot floppy into your system's floppy drive. If your system is turned off, power up your system. If your system is turned on, first shut down the active operating system in the proper manner, then restart the system. Linux should then boot from the floppy.



To use your boot floppy, your system's CMOS must be configured to allow booting from the floppy drive. If your system boots from its hard drive even when the boot floppy is present, you must change your system's CMOS configuration. The relevant option is generally named Boot Sequence, Boot Order, or something similar. The value you want is generally labeled *A:*, *C:*, or something similar. Consult your system's documentation for further information.

C.3 The GRUB Loader

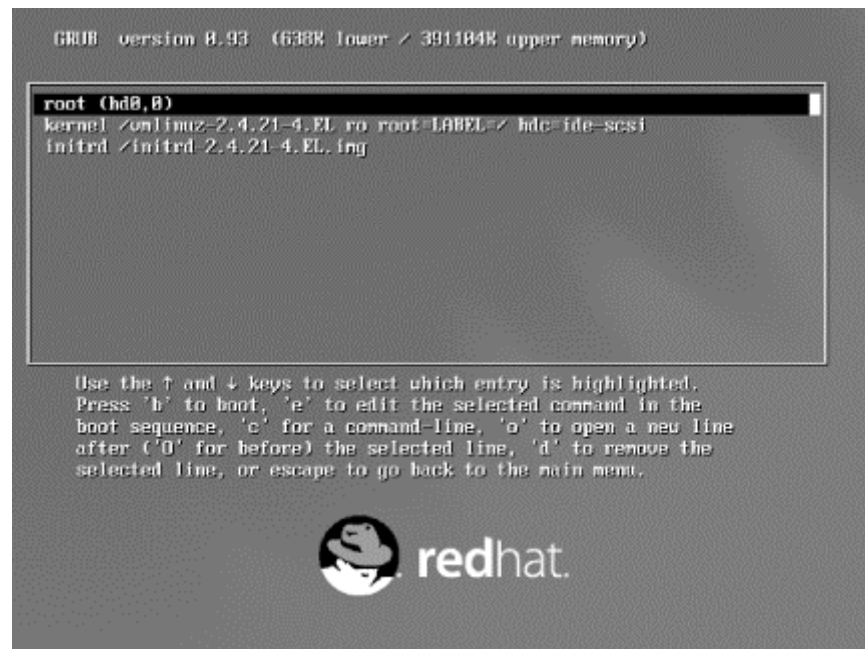
Most PCs can be booted from a floppy drive or hard drive; most recently manufactured computers can be booted from a CD-ROM drive. The first sector of a disk, diskette, or partition is known as the *boot sector*. The boot sector associated with a disk or diskette (the

first sector of the disk or diskette) is known as the *Master Boot Record* (MBR). In order for a diskette or disk to be bootable, it must contain a boot loader, which can reside in:

- The boot sector of the floppy diskette
- The MBR of the first hard disk or the first CD-ROM drive, if the PC supports booting from a CD-ROM
- The boot sector of a Linux filesystem partition on the first hard disk
- The boot sector of an extended partition on the first hard disk

GRUB is a sophisticated boot loader that can load Linux, Microsoft Windows 3.x and 9x, NT, 2000, XP, and other popular operating systems. Most users install GRUB on the MBR of their system's first hard disk. That way, when the system is started, it boots GRUB, which can be used to load Linux, Microsoft Windows, or another operating system. Unless you direct otherwise, the Red Hat Linux installation procedure automatically installs GRUB. So you don't need to install GRUB; you just need to configure it. Similarly, when you boot by using GRUB, you can also boot parameters to control the boot process; you can specify GRUB's boot parameters by selecting an operating system from GRUB's menu and pressing **e**. In response, GRUB displays an editor screen that shows the commands associated with the selected operating system, as shown in Figure C-1.

Figure C-1. The GRUB editor screen



Commands that can be used in the editor screen are listed in Table C-1.

Table C-1. GRUB editor commands	
Command	Meaning
B	Boot the currently selected operating system.

Table C-1. GRUB editor commands

Command	Meaning
E	Edit the currently selected GRUB command.
C	Open a screen for interactively entering and executing GRUB commands.
O	Enter a new command after the currently selected command.
O	Enter a new command before the currently selected command.
D	Delete the currently selected command.
Esc	Return to the main GRUB menu.

The principal GRUB commands are:

chainloader

Used to load a Microsoft operating system, including DOS, Windows 3.x, 9x, NT, or 2000.

initrd

Specifies the file containing an initial RAM disk used in loading Linux. This command is necessary, for instance, when booting Linux from a SCSI drive.

kernel

Specifies the file containing the Linux kernel to be booted.

rootrootnoverify

Specifies the partition to be mounted as the root partition. The **root** command causes the filesystem to be verified before the partition is mounted.

GRUB refers to hard disks using the syntax (hd*n*), where *n* specifies the device number assigned by the system BIOS. For example, (hd0) refers to the first hard drive. Partitions are designated by the related syntax (hd*n,m*), where *m* is the number of the partition. For example, (hd1,0) refers to the first partition of the second hard drive.

GRUB can refer to the disk blocks that comprise a file by using a special syntax known as a *blocklist*. A blocklist consists of a comma-separated list of block ranges, each of which consists of a starting block number, followed by a plus sign (+), followed by the number of blocks in the range. For example, the blocklist 0+1,10+20 refers to a block range beginning at block 0 and including 1 block and a block range beginning at block 10 and including 20 blocks.

If the starting block number of a block range is omitted, the block range is implied to begin with block 0. For example, the block range +1 begins at block 0 and includes 1 block.

A blocklist can appear after a reference to a partition; if no partition is identified, the partition identified by the preceding **root** command is implied. For example, in:

```
root (hd0,0)
chainlist +1
```

the blocklist +1 is implicitly associated with (hd0,0) and is equivalent to the blocklist (hd0,0)+1.

A typical GRUB entry for booting Linux resembles the following:

```
root (hd0,0)
kernel /boot/vmlinuz-2.4.18-14 root=LABEL=/
initrd /initrd-2.4.18-14.img
```

This entry boots the specified kernel residing on partition 0 of the first hard drive, mounting the partition labelled "/" as the root partition. Linux boot parameters, such as those described in the upcoming Section C.4, can be specified by using the **kernel** command. If the kernel requires access to special drivers residing on a RAM disk, a RAM disk can be identified as in this entry:

```
root (hd0,0)
kernel /boot/vmlinuz-2.4.18-14 ro root=LABEL=/
initrd /initrd-2.4.18-14.img
```

Notice that the name of the RAM disk file is specified as though the file resides in the root directory, whereas it actually resides in the */boot* directory. This is necessary because GRUB initially mounts the */boot* filesystem, as specified in the preceding **root** command. The mounted filesystem is treated by GRUB as its root filesystem.

A typical GRUB entry for booting a Microsoft operating system has this form:

```
rootnoverify (hd0,0)
chainloader +1
```

This entry boots the Microsoft operating system residing on partition of the first hard drive. The blocklist +1 points to the first block of the root partition. Microsoft operating systems place their boot file at this location.

To boot your system, GRUB uses your system's BIOS, which may not be able to load a Linux kernel (or other program) stored beyond cylinder 1023 of your hard drive. Booting a kernel stored beyond cylinder 1023 requires a motherboard that supports *logical byte addressing* (LBA). Most motherboards manufactured in the last several years are supposed to support LBA. Unfortunately, some motherboards that claim to support LBA do not. If you're installing Linux on a preexisting hard drive, you may not be able to place your Linux kernel in an appropriate location. In that case, you won't be able to use GRUB to boot your system.

To learn more about GRUB, see the following resources:

- The manpage for GRUB
- The GRUB home page, <http://www.gnu.org/software/grub>
- The Multiboot-with-GRUB mini HOWTO, <http://www.redhat.com/mirrors/LDP/HOWTO/mini/Multiboot-with-GRUB.html>

C.4 Boot Parameters

Boot parameters are specified using a three-part directive that includes the name of the parameter and an optional list of options, which consists of an equal sign (=) followed by a comma-separated list of option values

No spaces may appear in the directive. As an example, the following directive specifies the identity of the Linux root partition:

```
root=/dev/hda1
```

The installation program generally refers to partitions by using labels, so that you can boot a system even if you move the partitions around. A root directive referring to a label looks like this:

```
root=LABEL=/usr
```

You can specify multiple directives by separating them with a space. For example, the following specifies the identity of the Linux root partition and that the root partition is initially mounted read-only, so that a thorough check of its filesystem can be performed:

```
root=/dev/hda1 ro
```

Most directives are interpreted by the kernel, though GRUB is also capable of processing directives. If you specify a directive that neither the kernel nor GRUB understands (assuming you're using GRUB), a directive that includes an equal sign is passed to the *init* process as an environment variable. You learned about environment variables in [Chapter 12](#). A nonkernel directive that doesn't include an equal sign is passed to the *init* process. An example of this usage is specifying the directive `single`, which causes *init* to start your system in single-user mode:

```
root=/dev/hda1 ro single
```

C.4.1 General Boot Arguments

[Table C-2](#) describes some of the most popular and useful boot arguments. These arguments apply to your system as a whole; in subsequent sections, you'll learn about other boot arguments that apply to specific devices or functions. In addition to boot arguments previously introduced, the table describes the `reserve` argument, which is helpful in avoiding system memory conflicts.

Table C-2. Selected general boot arguments	
Argument	Description and options
<code>init=</code>	Specifies arguments passed by the kernel to the <code>init</code> process.
<code>mem=</code>	Specifies the amount of physical memory available to Linux; lets you instruct Linux to avoid high memory areas used by some systems for BIOS or caching. You can specify the amount as a hexadecimal number or as a decimal number followed by <code>k</code> or <code>M</code> , denoting kilobytes or megabytes, respectively.

Table C-2. Selected general boot arguments	
Argument	Description and options
<code>reserve=</code>	Specifies I/O ports that must not be probed. The port number is specified by using a hexadecimal number, and the range is specified by using a decimal number. For example, <code>reserve=0x320,32</code> specifies that I/O ports 320-33f must not be probed.
<code>ro</code>	Initially mounts the root filesystem in read-only mode, so that a more effective filesystem check can be done.
<code>root=</code>	<p><i>Identifies the root filesystem: /dev/fdn</i> Floppy disk <i>n</i> (0 or 1)</p> <p><i>/dev/hdxn</i> Partition <i>n</i> of IDE drive <i>x</i> (a to d)</p> <p><i>/dev/sdxn</i> Partition <i>n</i> of SCSI drive <i>x</i> (a to e)</p>
<code>rw</code>	Initially mounts the root filesystem in read/write mode; does not perform a filesystem check.
<code>vga=</code>	Specifies the default display mode set before booting. Specifying <code>vga=ask</code> will cause <i>lilo</i> to list the available video modes. You can then specify the desired mode in place of <code>ask</code> . (This argument is interpreted by <i>lilo</i> and will have no effect if another loader is used.)

C.4.2 RAM Disk Boot Arguments

[Table C-3](#) describes four boot arguments used in working with RAM disks. You won't likely need to specify any of these, but knowing about them may help you understand boot specifications written by others, including those used by Red Hat Linux.

Table C-3. Select RAM disk arguments	
Argument	Description and options
<code>load_ramdisk=</code>	Specifies that a RAM disk is not to be loaded (0) or is to be loaded (1).
<code>prompt_ramdisk=</code>	Specifies whether to provide a prompt instructing the user to insert a floppy containing a RAM disk (1) or provide no such prompt (0).
<code>ramdisk_size=</code>	Specifies the amount of RAM to be allocated to a RAM disk. If not specified, the default is 4 MB.
<code>ramdisk_start=</code>	Specifies the offset (in disk blocks from the start of the boot media) of the RAM disk data; lets a kernel and RAM disk data occupy the same

Table C-3. Select RAM disk arguments	
Argument	Description and options
	floppy disk.

C.4.3 SCSI Host Adapter Boot Arguments

[Table C-4](#) describes the most often used boot arguments related to SCSI host adapters. [Table C-5](#) describes the options used by the SCSI host adapter boot arguments and other boot arguments.

Table C-4. Select SCSI host adapter arguments	
Argument	Description and options
<code>advansys=</code>	Advansys SCSI host adapter: <code>iobase,[iobase,[iobase,[iobase]]]</code>
<code>aha152x=</code>	Adaptec aha151x, ada152x, aic6260, aic6360, and SB16-SCSI SCSI host adapters: <code>iobase[,irq[,scsi_id[,reconnect[,parity]]]]</code>
<code>aha1542=</code>	Adaptec aha154x SCSI host adapter: <code>iobase[,buson,busoff[,dmaspeed]]</code>
<code>aic7xxx=</code>	Adaptec aha274x, aha284x, aic7xxx SCSI host adapters: <code>extended,no_reset</code>
<code>AM53C974=</code>	AMD AM53C974-based SCSI host adapters: <code>scsi-id,dev_id,dmaspeed,offset</code> See the file <i>linux/drivers/scsi/README.AM53C974</i> .
<code>buslogic=</code>	BusLogic SCSI controller. Many options are available. See the BootPrompt-HOWTO.
<code>eata=</code>	EATA SCSI host adapter: <code>iobase,[iobase,[iobase,[iobase]]]</code>
<code>fdomain=</code>	Future Domain SCSI controller: <code>iobase,irq[,scsi_id]</code>
<code>in2000=</code>	Always SCSI host adapter. The driver for the Always SCSI controller accepts options in somewhat different format than other drivers. See the Boot Prompt-HOWTO.

Table C-4. Select SCSI host adapter arguments	
Argument	Description and options
<code>max-scsi-luns=</code>	Specifies the maximum number of SCSI logical units to be probed; lets you avoid probing devices that might lock up the SCSI bus.
<code>ncr5380=</code>	NCR 5380-based SCSI host adapters: <code>iobase, irq, dma</code> <code>membase, irq, dma</code>
<code>ncr53c400=</code>	NCR 53c400-based SCSI host adapters: <code>iobase, irq, pio</code> <code>membase, irq, pio</code>
<code>ncr53c406a=</code>	NCR 53c406a-based SCSI host adapters: <code>iobase, irq, pio</code> <code>membase, irq, pio</code>
<code>pas16=</code>	Pro Audio Spectrum SCSI host adapter: <code>iobase, irq</code>
<code>ppa=</code>	Iomega parallel port SCSI adapter for ZIP drive: <code>iobase, speed_high, speed_low, nybble</code>
<code>st0x=</code>	Seagate ST-0x SCSI host adapter: <code>membase, irq</code>
<code>t128=</code>	Trantor T128 SCSI host adapter: <code>membase, irq</code>
<code>tmc8xx=</code>	Future Domain TMC-8xx and TMC-950 SCSI host adapters: <code>membase, irq</code>
<code>u14-34f=</code>	Ultrastor SCSI host adapter: <code>iobase, [iobase, [iobase, [iobase]]]</code>
<code>wd7000=</code>	Western Digital WD7000 SCSI host adapter: <code>irq, dma, iobase</code>

For example, from [Table C-4](#) you can learn that Adaptec aha154x SCSI host adapters use a boot argument having the form:

`iobase[, buson, busoff[, dmaspeed]]`

[Table C-5](#) can help you understand the form of **iobase** and other options. The **iobase** option, for example, lets you specify the I/O port associated with the SCSI host adapter. For example, you can specify a boot argument for an Adaptec aha154x SCSI host adapter by writing only an **iobase** option; the remaining options are optional. However, as indicated by the square brackets, if you include a **buson** option, you must include a **busoff** option. Similarly, to include the **dmasspeed** option, you must include each of the other options. Here's an example of a complete boot argument:

```
aha1542=0x300,11,4
```

Table C-5. Select boot prompt arguments

Option	Description and options
<code>busoff</code>	The interval (number of microseconds) during which the device will relinquish the ISA bus, specified as a decimal integer, for example, 4.
<code>buson</code>	The interval (number of microseconds) during which the device will dominate the ISA bus, specified as a decimal integer, for example, 11.
<code>ctl</code>	The I/O port used for control, specified as a hexadecimal number, for example, 0x300.
<code>cyl,head,sect</code>	The geometry of the storage device, specified as three integers denoting the number of cylinders, heads, and sectors, respectively.
<code>dev_id</code>	A SCSI device with which the host adapter communicates, specified as a decimal integer, for example, 2.
<code>dma</code>	The DMA (direct memory access) channel by used by the device, specified as a decimal integer, for example, 3.
<code>dmasspeed</code>	The rate (in MB/sec.) at which DMA transfers are performed, specified as a decimal integer, for example, 5.
<code>extended</code>	Whether extended translation for large disks is enabled (1) or not (0).
<code>magic_number</code>	Whether the driver attempts to work, even if the firmware version is unknown (79); other values are ignored.
<code>no_reset</code>	Whether the driver should reset the SCSI bus when setting up the host adapter at boot (1) or not (0).
<code>iobase</code>	An I/O port, specified as a hexadecimal number, for example, 0x300.
<code>irq</code>	A hardware interrupt number, specified as a decimal integer, for example, 5.
<code>is_pas_card</code>	Whether a Pro Audio Spectrum (PAS) card is used; otherwise, do not specify this option.

Table C-5. Select boot prompt arguments	
Option	Description and options
<code>membase</code>	The base address of a memory region used for memory-mapped I/ O, specified as a hexadecimal number, for example, 0x2000.
<code>parity</code>	Whether the SCSI host adapter uses parity (1) or does not use parity (0).
<code>pio</code>	Whether <code>insl</code> and <code>outsl</code> multibyte instructions (1) or <code>inb</code> and <code>outb</code> single-byte instructions (0) are used.
<code>reconnect</code>	Whether the SCSI host adapter is allowed to disconnect and reconnect (1) or holds a connection until the operation is complete (0).
<code>scsi_id</code>	The ID by which the SCSI host adapter identifies itself, specified as a decimal integer, for example, 7.

To determine a proper value for options described in [Table C-5](#), you must often know something about the hardware structure of your system. The procedures described in [Chapter 2](#) will help you.

C.4.4 IDE Hard Drive and CD-ROM Boot Arguments

[Table C-6](#) describes the most commonly used boot arguments associated with IDE hard drives and CD-ROM drives. Refer to [Table C-5](#) to determine the form of the italicized options.

Table C-6. Select IDE hard drive arguments	
Argument	Description and options
<code>hdx=</code>	IDE hard drive or CD-ROM (<i>x</i> denotes the physical device and must be a letter from a to h):
	<i>autotune</i> Specifies that the driver should attempt to tune the interface to the fastest possible mode and speed
	<i>cdrom</i> Specifies that the drive is a CD-ROM drive
	<i>cyl,head,sect</i> Specifies the geometry of the drive
	<i>none</i> Specifies that the drive is not present—do not probe
	<i>noprobe</i>

Table C-6. Select IDE hard drive arguments

Argument	Description and options
	<p>Specifies that the driver should not probe for the device</p> <p><i>nowerr</i></p> <p>Specifies that the <code>WRERR_STAT</code> bit should be ignored on this drive</p>
<code>ide0=</code>	<p>IDE hard drive or CD-ROM:</p> <p><i>ali14xx</i></p> <p>Probe for and support the ali14xx interface</p> <p><i>cmd640_vlb</i></p> <p>Probe for and support the cmd640 chip (required for controllers using a VLB interface)</p> <p><i>dtc2279</i></p> <p>Probe for and support the dtc2278 interface</p> <p><i>ht6560b</i></p> <p>Probe for and support the ht6560b interface</p> <p><i>qd6580</i></p> <p>Probe for and support the qd6580 interface</p> <p><i>umc8672</i></p> <p>Probe for and support the umc8672 interface</p>
<code>idx=</code>	<p>IDE hard drive or CD-ROM (<i>x</i> specifies the physical device and must be a digit from 0 to 3):</p> <p><i>autotune</i></p> <p>Specifies that the driver should attempt to tune the interface to the fastest possible mode and speed</p> <p><i>iobase</i></p> <p>Specifies the I/O port used by the drive</p> <p><i>iobase,ctl</i></p> <p>Specifies the I/O port and control port used by the drive</p> <p><i>iobase,ctl,irq</i></p> <p>Specifies the I/O port, control port, and IRQ used by the drive</p> <p><i>noautotune</i></p> <p>Specifies that the driver should not attempt to tune the interface for</p>

Table C-6. Select IDE hard drive arguments	
Argument	Description and options
	fastest mode and speed
	<i>noprobe</i> Specifies that the driver should not probe for the device
	<i>serialize</i> Specifies that I/O operations should not be overlapped

C.4.5 Non-IDE CD-ROM Drive Boot Arguments

[Table C-7](#) describes the most common boot arguments for non-IDE CD-ROM drives. Refer to [Table C-5](#) to determine the form of the italicized options.

Table C-7. Select CD-ROM arguments	
Argument	Description and options
aztcd=	Aztech CD-ROM <i>iobase[,magic_number]</i> ^[1]
cdu31a=	Sony CDU-31A or CDU-33A CD-ROM <i>iobase,[irq[,is_pas_card]</i>
sonycd535=	Sony CDU-535 CD-ROM: <i>iobase[,irq]</i>
gsacd=	Goldstar CD-ROM: <i>iobase</i>
isp16=	ISP16 CD-ROM: <i>[port[,irq[,dma]]][[,]drive_type]</i>
mcd=	Mitsumi CD-ROM: <i>iobase,[irq[,wait_value]]</i>
optcd=	Optical Storage CD-ROM: <i>iobase</i>
cm206=	Phillips CD206 CD-ROM: <i>[iobase][[,irq]</i>

Table C-7. Select CD-ROM arguments	
Argument	Description and options
<code>sjcd=</code>	Sanyo CD-ROM: <code>iobase[,irq[,dma_channel]]</code>
<code>sbpcd=</code>	SoundBlaster Pro CD-ROM: <code>iobase,type</code>

^[1] Don't include the square brackets in your boot argument; they merely indicate which options must be present.

C.4.6 Floppy Drive Boot Arguments

A few systems require special boot arguments to best use their floppy drives. [Table C-8](#) describes the most common boot arguments related to floppy drives. Floppy drives that are not well behaved may malfunction if you specify the `daring` option, which you should use only with care. For additional boot arguments related to floppy drives, see `/usr/src/linux/Documentation/floppy.txt`.

Table C-8. Select floppy disk arguments	
Argument and option	Description
<code>floppy=asus_pci</code>	Specifies that only units 0 and 1 are allowed, to work around problem with BIOS of certain ASUS motherboards.
<code>floppy=daring</code>	Specifies that the floppy controller is well behaved, allowing more efficient operation.
<code>floppy=0,daring</code>	Specifies that the floppy controller may not be well behaved (default).
<code>floppy=thinkpad</code>	Specifies that the system is an IBM ThinkPad.
<code>floppy=no_unexpected_interrupts</code> or <code>floppy=L40SX</code>	Specifies that a message should be printed when an unexpected interrupt is received. This is required by IBM L40SX laptops in certain video modes.

C.4.7 Bus Mouse Boot Arguments

Two boot arguments provide bus mouse support. The first supports the Microsoft bus mouse:
`msmouse=irq`

The second supports any non-Microsoft bus mouse:

```
bmouse=irq
```

Each argument accepts a single option specifying the IRQ associated with the mouse.

C.4.8 Parallel Port Printer Boot Arguments

The Linux printer driver claims all available parallel ports. If you want to access a device other than a printer attached to a parallel port, you must instruct the printer driver to reserve only the ports associated with printers. To do so, use the `lp` boot argument, which takes as its options a list of ports and IRQs used to support printers. For example, the following boot argument specifies two printers:

```
lp=0x3bc,0,0x378,7
```

The first printer is on port 0x3bc and the second is on port 0x378. The first printer uses a special IRQ-less mode known as polling, so its IRQ is specified as 0. The second printer uses IRQ 7.

To disable all printers, specify `lp=0`.

C.4.9 Loadable Ethernet Drivers

Early versions of Linux used a so-called monolithic kernel. At that time, Linux distributions typically included several kernels, offering support for a variety of devices that might be needed to boot and install a Linux system. Devices not needed to boot and install a system—so-called special devices—had second-class status. To access special devices, users had to compile customized kernels that included support for those devices. When adding a device to a system, users often had to compile a new kernel, which was something of an inconvenience.

More recent versions of Linux feature a modular kernel, which allows drivers to be dynamically loaded on command. This makes it much easier than before to configure your Linux system to support Ethernet cards and other special devices. Red Hat Linux is generally able to configure your primary Ethernet card automatically, by probing for it during installation of Linux.

However, the autoprobe doesn't always succeed. Moreover, if you have more than one Ethernet card, the installation program sets up only the first card it finds. To set up additional cards, you need to know a bit about Linux's loadable modules.

C.4.10 Dynamically Loading a Modular Driver

To dynamically load a modular driver, issue the following command:

```
# modprobe driver
```

where *driver* specifies the module to be loaded. As an example, the command:

```
# modprobe ne2k-pci
```

loads the modular driver for the PCI-based NE2000 Ethernet card.

To find out what network adapters are supported by Red Hat Linux or to find out what driver to use with a particular adapter, see the Red Hat Linux Hardware Compatibility List, <http://hardware.redhat.com>.

When a driver is loaded, it generally probes to locate the supported device. In case an autoprobe fails, most drivers let you specify the I/O port and IRQ by using a command like the following:

```
# modprobe ne2k=pci io=0x280 irq=11
```

Some cards support additional options; these are documented in the file */usr/src/linux/Documentation/networking/net-modules.txt*.

C.4.11 Loading Modular Drivers at Boot Time

The Linux kernel automatically loads modules specified in the module configuration file, */etc/modules.conf*. So, once you've determined the proper module and options required by your Ethernet card, you can add a line or two to the module configuration file so that your card will be made ready to operate each time you boot your system.

The **alias** directive associates a logical module name with an actual module. Logical module names specify types of devices; for example, *eth0* specifies the first Ethernet card in a system, and *eth1* specifies the second Ethernet card in a system. Suppose your system includes two Ethernet cards: a non-PCI-based NE2000 and an SMC EtherPower, which is based on DEC's TULIP chip. You could use the following directives to automatically load these modules at boot time:

```
alias eth0 ne
alias eth1 tulip
```

If a driver requires options, you can specify them by using an options directive, which has the following form:

```
options driver argument=value[,value,...] argument=value[,value,...] ...
```

For example, you might specify the I/O port and IRQ used by the NE2000 card like this:

```
options ne io=0x280 irq=12
```

Most ISA modules accept parameters like *io=0x340* and *irq=12* on the **insmod** command line. You should supply these parameters to avoid probing for the card. Unlike PCI and EISA devices, ISA devices sometimes cannot be safely autoprobeed.

C.4.12 Administering Modular Drivers

The **lsmod** command, which takes no arguments, lists the loaded modular drivers. To unload a modular driver, specify the driver as the argument of the **modprobe** command and specify the **-r** argument. For example, to remove the *ne* driver, issue the command:

```
# modprobe -r ne
```

To unload every unused module—that is, every module not associated with an operational device—invoke the **rmmod** command and specify the **-a** argument:

```
# rmmod -a
```


You can't remove a module that's in use; therefore, you must shut down the device before removing it. To shut down an Ethernet device, you can use Neat. Or you can issue the following command:

```
# ifconfig ethn down
```

where *ethn* specifies the logical device (for example, eth0 or eth1).

Appendix D. Linux Command Quick Reference

The following list describes some of the most useful and popular Linux commands. Consult the manpage for each command to learn about additional arguments and details of operation.

adduser *userid*

Creates a new *userid*, prompting for necessary information (requires *root* privs).

alias **name**='command'

Defines *name* as an alias for the specified command.

apropos *keyword*

Searches the manual pages for occurrences of the specified keyword and prints short descriptions from the beginning of matching manual pages.

at *time*

at -f *file time*

Executes commands entered via STDIN (or by using the alternative form, the specified file) at the specified time. The time can be specified in a variety of ways; for example, in hour and minute format (*hh:mm*) or in hour, minute, month, day, and year format (*hh:mm mm/dd/yy*).

atq

Displays descriptions of jobs pending via the **at** command.

atrm *job*

Cancels execution of a job scheduled via the **at** command. Use the **atq** command to discover the identities of scheduled jobs.

bg

bg *jobs*

Places the current *job* (or by using the alternative form, the specified jobs) in the background, suspending its execution so that a new user prompt appears immediately. Use the **jobs** command to discover the identities of background jobs.

cal *month year*

Displays a calendar for the specified month of the specified year.

cat *files*

Displays the contents of the specified files.

cd

cd *directory*

Changes the current working directory to the user's home directory or the specified directory.

chgrp *group files*

chgrp -R *group files*

Changes the group of the specified files to the specified group. The alternative form of the command operates recursively, changing the group of subdirectories and files beneath a specified directory. The group must be named in the */etc/groups* file, maintained by the **newgroup** command.

chmod *mode files*

chmod -R *mode files*

Changes the access mode of the specified files to the specified mode. The alternative form of the command operates recursively, changing the mode of subdirectories and files beneath a specified directory.

chown *userid files*

chown -R *userid files*

Changes the owner of the specified files to the specified *userid*. The alternative form of the command operates recursively, changing the owner of subdirectories and files beneath a specified directory

clear

Clears the terminal screen.

cmp *file1 file2*

Compares two files, reporting all discrepancies. Unlike the **diff** command, **cmp** can compare multiple files and binary files.

cp *file1 file2*

cp *files directory*

cp -R *files directory*

Copies a file to another file or directory or copies a subdirectory and all its files to another directory.**date**

date *date*

Displays the current date and time or changes the system date and time to the specified value, of the form *MMddhhmmyy* or *MMddhhmmyyyy*.

df

Displays the amount of free disk space on each mounted filesystem.

diff *file1 file2*

Compares two files, reporting all discrepancies. Similar to the **cmp** command, though the output format differs.

dmesg

Displays the messages resulting from the most recent system boot.

du

du *directories*

Displays the amount of disk space used by the current directory (or the specified directories) and its (their) subdirectories.

echo *string*

echo -n *string*

Displays the specified text on the standard output stream. The **-n** option causes omission of the trailing newline character.

fdformat *device*

Formats the media inserted in the specified floppy disk drive. The command performs a low-level format only; it does not create a filesystem. To create a filesystem, issue the **mkfs** command after formatting the media.

fdisk *device*

Edits the partition table of the specified hard disk.

fg

fg *jobs*

Brings the current job (or the specified jobs) to the foreground.

file *files*

Determines and prints a description of the type of each specified file.

find *path -name pattern -print*

Searches the specified path for files with names matching the specified pattern (usually enclosed in single quotes) and prints their names. The find command has many other arguments and functions; see the online documentation.

finger *users*

Displays descriptions of the specified users.

free

Displays the amount of used and free system memory.

ftp *hostname*

Opens an FTP connection to the specified host, allowing files to be transferred. The FTP program provides subcommands for accomplishing file transfers; see the online documentation.

grep *pattern files*

grep -i *pattern files*

grep -n *pattern files*

grep -v *pattern files*

Searches the specified files for text matching the specified pattern (usually enclosed in single quotes) and prints matching lines. The **-i** option specifies that matching is performed without regard to case. The **-n** option specifies that each line of output is preceded by the filename and line number. The **-v** option reverses the matching, causing nonmatched lines to be printed.

gzip *files*

gunzip *files*

Expands or compresses the specified files. Generally, a compressed file has the same name as the original file, followed by **.gz**.

head *files*

Displays the first several lines of each specified file.

hostname

hostname *name*

Displays (or sets) the name of the host.

info

Launches the GNU Texinfo help system.

init *runlevel*

Changes the system runlevel to the specified value (requires *root* privileges).

insmod *module*

Dynamically loads the specified module (requires *root* privileges).

ispell *files*

Checks the spelling of the contents of the specified files.

jobs

Displays all background jobs.

kill *process_ids*

kill -l

kill -signal *process_ids*

Kills the specified processes, prints a list of available signals, or sends the specified processes the specified signal (given as a number or name).

killall *program*

killall -signal *program*

Kills all processes that are instances of the specified program or sends the specified signal to all processes that are instances of the specified program.

less *file*

Lets the user peruse a file too large to be displayed as a single screen (page) of output. The **less** command, which is more powerful than the **more** command, provides many subcommands that let the user navigate the file. For example, the spacebar moves forward one page, the **b** key moves back one page, and the **q** key exits the program.

links *URL*

Views the specified web page.

ln *old new*

ln -s *old new*

Creates a hard (or soft) link associating a new name with an existing file or directory.

locate *pattern*

Locates files with names containing the specified pattern. Uses the database maintained by the **updatedb** command.

lpq

Displays the entries of the print queue.

lpr *files*

Displays the specified files.

lprm *job*

Cancels printing of the specified print queue entries. Use **lpq** to determine the contents of the print queue.

ls

ls *files*

ls -a *files*

ls -l *files*

ls -lR *files*

Lists (nonhidden) files in the current directory or the specified files or directories. The **-a** option lists hidden files as well as nonhidden files. The **-l** option causes the list to include descriptive information, such as file size and modification date. The **-R** option recursively lists the subdirectories of the specified directories.

mail

Launches a simple mail client that permits sending and receiving email messages.

man *title*

man *section title*

Displays the specified manpage.

mkdir *directories*

mkdir -p *directories*

Creates the specified directories. The **-p** option causes creation of any parent directories needed to create a specified directory.

mkfs -t *type device*

Creates a filesystem of the specified type (such as *ext3* or *msdos*) on the specified *device* (requires *root* privileges).

mkswap *device*

Creates a Linux swap space on the specified hard disk partition (requires *root* privileges).

more *file*

Lets the user peruse a file too large to be displayed as a single screen (page) of output. The **more** command provides many subcommands that let the user navigate

the file. For example, the spacebar moves forward one page, the **b** key moves back one page, and the **q** key exits the program.

mount

mount *device directory*

mount -o option -t type device directory

Displays the mounted devices or mounts the specified device at the specified mount point (generally a subdirectory of */mnt*). The mount command consults */etc/fstab* to determine standard options associated with a device. The command generally requires *root* privileges. The **-o** option allows specification of a variety of options, for example, **ro** for read-only access. The **-t** option allows specification of the filesystem type (for example, *ext3*, *msdos*, or *iso9660*, the filesystem type generally used for CD-ROMs).

mv paths target

Moves the specified files or directories to the specified target.

nano

nano file

Launch nano to edit or create specified file, if any.

newgroup group

Creates the specified group.

passwd

passwd user

Changes the current user's password or that of the specified user (requires *root* privileges). The command prompts for the new password.

ping -n ip_address

ping host

Sends an echo request via TCP/IP to the specified host. A response confirms that the host is operational.

pr files

Formats the specified files for printing, by inserting page breaks and so on. The command provides many arguments and functions.

ps

ps -aux

Displays the processes associated with the current *userid* or displays a description of each process.

pwd

Displays the absolute path corresponding to the current working directory.

reboot

Reboots the system (requires *root* privileges).

reset

Clears the terminal screen and resets the terminal status.

rm *files*

rm -f *files*

rm -i *files*

rm -if *files*

rm -rf *files*

Deletes the specified files or (when the **-r** option is specified) recursively deletes all subdirectories of the specified files and directories. The **-f** option suppresses confirmation; the **-i** option causes the command to prompt for confirmation. Because deleted files cannot generally be recovered, the **-f** option should be used only with extreme care, particularly when used by the *root* user.

rmdir *directories*

rmdir -p *directories*

Deletes the specified empty directories or (when the **-p** option is specified) the empty directories along the specified path.

scp *host1:file host2:*

Copies *file* from *host1* to *host2*, via SSH.

shutdown *minutes*

shutdown -h *minutes*

shutdown -r *minutes*

Shuts down the system after the specified number of minutes elapses (requires *root* privileges). The **-r** option causes the system to be rebooted once it has shut down. If the **-r** option is absent, the system is halted and powered off; the **-h** option also halts and shuts down the system. Alternatively, **now** can be used instead of *minutes*, which forces an immediate reboot or halt of the system.

sleep *time*

Causes the command interpreter to pause for the specified number of seconds.

sort *files*

Sorts the specified files. The command has many useful arguments; see the online documentation.

split *file*

Splits a file into several smaller files. The command has many arguments; see the online documentation.

ssh *host -l userid*

Logs in to *host* via SSH, using the specified *userid*.

su**su** *user***su -****su -** *user*

Changes the current *userid* to *root* or to the specified *userid* (the latter requires *root* privileges). The - option establishes a default environment for the new *userid*.

swapoff *device*

Disables use of the specified device for swapping (requires *root* privileges).

swapon *device*

Enables use of the specified device for swapping (requires *root* privileges).

sync

Completes all pending input/output operations (requires *root* privileges).

tail *file***tail -f** *file***tail -n** *file*

Displays the last several lines of the specified files. The -f option causes the command to continuously print additional lines as they are written to the file. The -n option specifies the number of lines to be printed.

talk *user*

Launches a program that allows a chatlike dialog with the specified user.

tar cvf *tar_file files*

tar zcvf *tar_file files*

Creates a tar file with the specified name, containing the specified files and their subdirectories. The **z** option specifies that the tarfile will be compressed.

tar xvf *tar_file*

tar zxvf *tar_file*

Extracts the contents of the specified tarfile. The **z** option specifies that the tarfile has been compressed.

telnet *host*

Opens a login session on the specified host.

time

Times the execution of a job.

top

Displays a display of system processes that's continually updated until the user presses the **q** key.

touch *file*

Changes *file* access time. If the specified file does not exist, the command creates an empty (new) file.

traceroute *host*

Uses echo requests to determine and print a network path to the host.

umask *mask*

Specifies default permissions assigned to created directories and files.

umount *device*

Unmounts the specified filesystem (generally requires *root* privileges).

uname -a

Displays information about the system.

unzip *file*

Unzips a compressed file.

uptime

Displays the system uptime.

w

Displays the current system users.

wall

Displays a message to each user except those who've disabled message reception. Type **Ctrl-D** to end the message.

wc *files*

Displays the number of characters, words, and lines in the specified files.

who

Displays information about system users.

zip *file*

Compresses the specified file.

[Table D-1](#) identifies Linux commands that perform functions similar to MS-DOS commands. The operation of the Linux command is not generally identical to that of the corresponding MS-DOS command. See the index to this book or the Linux online documentation for further information about Linux commands.

Table D-1. MS-DOS commands and their related Linux commands	
MS-DOS	Linux
ATTRIB	chmod
CD	cd
CHKDSK	df, du
DELTREE	rm -R
DIR	ls -l
DOSKEY	(built-in; no need to launch separately)
EDIT	nano, vi, and so on
EXTRACT	tar
FC	cmp, diff
FDISK	fdisk
FIND	grep
FORMAT	fdformat
MORE	more
MOVE	mv
SORT	sort
START	at, bg
XCOPY, XCOPY32	cp

Colophon

Our look is the result of reader comments, our own experimentation, and feedback from distribution channels. Distinctive covers complement our distinctive approach to technical topics, breathing personality and life into potentially dry subjects.

The cover image of a man wearing a wide-brimmed hat is a 19th-century engraving from *Marvels of the New West: A Vivid Portrayal of the Stupendous Marvels in the Vast Wonderland West of the Missouri River*, by William Thayer (The Henry Bill Publishing Co., 1888).

Philip Dangler was the production editor and copyeditor for *Learning Red Hat Enterprise Linux and Fedora, Fourth Edition*. Darren Kelly, Matt Hutchinson, and Claire Cloutier provided quality control. Angela Howard wrote the index.

Edie Freedman designed the cover of this book. Emma Colby produced the cover layout with QuarkXPress 4.1 using Adobe's ITC Garamond font. David Futato designed the CD labels.

David Futato designed the interior layout. The chapter opening images are from the Dover Pictorial Archive, *Marvels of the New West*, and *The Pioneer History of America: A Popular Account of the Heroes and Adventures*, by Augustus Lynch Mason, A.M. (The Jones Brothers Publishing Company, 1884). This book was converted by Julie Hawks to FrameMaker 5.5.6 with a format conversion tool created by Erik Ray, Jason McIntosh, Neil Walls, and Mike Sierra that uses Perl and XML technologies. The text font is Linotype Birka; the heading font is Adobe Myriad Condensed; and the code font is LucasFont's TheSans Mono Condensed. The illustrations that appear in the book were produced by Robert Romano and Jessamyn Read using Macromedia FreeHand 9 and Adobe Photoshop 6. The tip and warning icons were drawn by Christopher Bing.

The online edition of this book was created by the Safari production group (John Chodacki, Becki Maisch, and Madeleine Newell) using a set of Frame-to-XML conversion and cleanup tools written and maintained by Erik Ray, Benn Salter, John Chodacki, and Jeff Liggett.