Red Hat Linux 6.2

The Official Red Hat Linux Reference Guide

Trend Micro, Inc.
EX1011-1

ISBN: 1-58569-020-1

Red Hat, Inc.
2600 Meridian Parkway Durham NC 27709 US 919-547-0012 1-888-733-4281 919-547-0024
docs@redhat.com 13588 Research Triangle Park NC 27713

© 2000 Red Hat, Inc.

RefGuide(EN)-6.2-Print-RHI (02/00)

Red Hat is a registered trademark and the Red Hat Shadow Man logo, RPM, the RPM logo, and Glint are trademarks of Red Hat, Inc.

Linux is a registered trademark of Linus Torvalds.

Motif and UNIX are registered trademarks of The Open Group.

Alpha is a trademark of Digital Equipment Corporation.

SPARC is a registered trademark of SPARC International, Inc. Products bearing the SPARC trademark are based on an architecture developed by Sun Microsystems, Inc.

Netscape is a registered trademark of Netscape Communications Corporation in the United States and other countries.

TrueType is a registered trademark of Apple Computer, Inc.

Windows is a registered trademark of Microsoft Corporation.

All other trademarks and copyrights referred to are the property of their respective owners.

Copyright © 2000 by Red Hat, Inc. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, V0.4 or later (the latest version is presently available at http://www.opencontent.org/openpub/).

Distribution of substantively modified versions of this document is prohibited without the explicit permission of the copyright holder.

Distribution of the work or derivative of the work in any standard (paper) book form for commercial purposes is prohibited unless prior permission is obtained from the copyright holder.

Printed in Canada, Ireland, and Japan

Trend Micro, Inc.
EX1011-2

Contents

Red Hat Linux 6.2

Introduction vii
- Welcome vii
- Getting the Documentation That's Right for You vii
- More to Come xi
- Sign Up for Support xi

Part I System-Related Reference 13

Chapter 1 Red Hat Linux 6.2 New Features 15
- 1.1 Installation-related Enhancements 15
- 1.2 System-Related New Features 15
- 1.3 Miscellaneous New Features 19

Chapter 2 System Administration 23
- 2.1 Filesystem Structure 23
- 2.2 Special Red Hat File Locations 27
- 2.3 Users, Groups and User-Private Groups 28
- 2.4 Configuring Console Access 32
- 2.5 The floppy Group 36
- 2.6 User Authentication with PAM 36
- 2.7 Shadow Utilities 41
- 2.8 Building a Custom Kernel 42
- 2.9 Sendmail 48
- 2.10 Controlling Access to Services 50
- 2.11 Anonymous FTP 50
- 2.12 NFS Configuration 51
- 2.13 The Boot Process, Init, and Shutdown 52
- 2.14 Rescue Mode 71

Chapter 3 System Configuration 75
3.1 System Configuration with linuxconf 75
3.2 System Configuration with the Control Panel 115

Chapter 4 PowerTools 133
4.1 PowerTools Packages 133
4.2 Reading the Contents of the CD-ROM 133
4.3 Installing PowerTools Packages 133

Chapter 5 Working with Update Agent 137
5.1 Starting Update Agent 137
5.2 Configuring Update Agent 138
5.3 Using Update Agent 144

Chapter 6 Package Management with RPM 149
6.1 RPM Design Goals 149
6.2 Using RPM 151
6.3 Impressing Your Friends with RPM 157
6.4 Other RPM Resources 159

Chapter 7 Gnome-RPM 161
7.1 Starting Gnome-RPM 163
7.2 The Package Display 164
7.3 Installing New Packages 166
7.4 Configuration 169
7.5 Package Manipulation 175

Chapter 8 Lightweight Directory Access Protocol (LDAP) 183
8.1 What is LDAP? 183
8.2 Pros and Cons of LDAP 184
8.3 Uses for LDAP 184
8.4 LDAP Terminology 185
8.5 OpenLDAP Files 185

8.6 OpenLDAP Daemons and Utilities 187
8.7 Modules for Adding Extra Functionality to LDAP 188
8.8 LDAP How To: A Quick Overview 188
8.9 Configuring Your System to Authenticate Using OpenLDAP 189
8.10 LDAP Resources on the Web 192

Part II Installation-Related Reference 195

Chapter 9 Preparing for a Text Mode Installation 197
9.1 Things You Should Know 197

Chapter 10 Installing Red Hat Linux via Text Mode 205
10.1 The Installation Program User Interface 205
10.2 Starting the Installation Program 209
10.3 Choosing a Language 211
10.4 Selecting a Keyboard Type 211
10.5 Selecting an Installation Method 212
10.6 Identify Disk Partition to Install From 214
10.7 Installing over a Network 215
10.8 Welcome 219
10.9 Upgrading or Installing 220
10.10 Partitioning Your Disk for Red Hat Linux 226
10.11 Installing LILO 239
10.12 Naming Your Computer 245
10.13 Configuring a Network Connection 247
10.14 Configuring Your Mouse 248
10.15 Configuring the Time Zone 250
10.16 Setting a Root Password 251
10.17 Creating a User Account 253
10.18 Authentication Configuration 254
10.19 Select Packages to Install 256
10.20 Configuring Your Video Adapter 259
10.21 Package Installation 261
10.22 Configuring the X Window System 263

10.23 Creating a Boot Disk 269
10.24 Finishing Up 271

Part III Appendixes 273

Appendix A General Parameters and Modules 275
A.1 A Note About Kernel Drivers 275
A.2 CD-ROM Module Parameters 276
A.3 SCSI parameters 279
A.4 Ethernet parameters 285

Appendix B An Introduction to Disk Partitions 295
B.1 Hard Disk Basic Concepts 295

Appendix C Driver Disks 321
C.1 Why Do I Need a Driver Disk? 321

Appendix D How to Create a Dual-Boot System 323
D.1 If Your Computer Already Has An Operating System 323
D.2 Setting Up a Dual-Boot Environment 325
D.3 Partitioning with FIPS 328

Appendix E RAID (Redundant Array of Independent Disks) 335
E.1 What is RAID? 335

Appendix F Kickstart Installations 343
F.1 What are Kickstart Installations 343
F.2 How Do You Perform a Kickstart Installation? 343
F.3 Starting a Kickstart Installation 345
F.4 The Kickstart File 347
F.5 Kickstart Commands 348

Introduction

Welcome

Welcome to the *Official Red Hat Linux Reference Guide*.

The *Official Red Hat Linux Reference Guide* contains useful information about your Red Hat Linux system. In fact, much of the information you'll find within can be extended to just about any Linux distribution. From fundamental concepts such as using RPM and Gnome-RPM to the finer points of using disk partitioning, we hope you'll find this book to be a valuable resource.

This guide is for you if you want to learn a bit more about how your Red Hat Linux system works. Among the featured entries, you'll learn about:

- Partitioning concepts -- Both an introduction to disk partitions and the strategies behind "finding a home" for more than one operating system on hard drives.
- Text mode installation -- Despite Red Hat Linux's GUI installation, you may want the control of a text mode install. Here's what you'll find, and what to expect.
- RPM -- From both the Gnome-RPM front-end to using RPM at the console.
- RAID concepts -- Take one disk drive, add another, and another… Make them appear as a single logical unit, and you've got power and performance.
- Post-installation configuration -- Want to do a little tweaking after the installation? Here's where you can get a start.

Getting the Documentation That's Right for You

While the *Official Red Hat Linux Reference Guide* digs into more of the nuts and bolts of your Red Hat Linux system, it is critical to make sure you have documentation that is appropriate to your level of Linux expertise. Regardless of your experience with Linux, it can be easy to feel overwhelmed without the right documentation.

Let's take a look at three categories of people using Red Hat Linux, and try to be more explicit in terms of the documentation you'll need. Let's start by figuring out your experience level. Here are the three basic categories:

New to Linux

Has never used any Linux (or Linux-like) operating system before, or has had only limited exposure to Linux. May or may not have experience using other operating systems (such as Windows). Is this you? If so, please turn to *Documentation For First-Time Linux Users*.

Some Linux Experience

Has installed and successfully used Linux (but not Red Hat Linux) before. Or, may have equivalent experience with other Linux-like operating systems. Does this describe you? If so, please turn to *For the More Experienced*.

Old-Timer

Has installed and successfully used Red Hat Linux before. Are you an old-timer? If so, please turn to *Documentation for Linux Gurus*.

Documentation For First-Time Linux Users

"A journey of a thousand miles begins with a single step." This old saying can be applied to learning about your Red Hat Linux system. Learning to use a Linux system effectively can be a long, rewarding journey, in which you find that you can easily do things about which people with other operating systems can only dream. But like all journeys, you've got to start somewhere, and take that first step.

First, get yourself some documentation! This cannot be stressed enough; without documentation you will only become frustrated at your inability to get your Red Hat Linux system working the way you want.

Here's the sort of Linux documentation you should get your hands on:

- A brief history of Linux -- Many aspects of Linux are the way they are because of historical precedent. There is also a Linux culture that, again, is based to a great deal on past history. A bit of knowledge about the history of Linux will serve

you well, particularly as you interact with more experienced Linux users on the Internet.

- An explanation of how Linux works -- While it's not necessary to delve into the most arcane aspects of the Linux kernel, it's a good idea to know something about how Linux is put together. This is particularly important if you've been working with other operating systems; some of the assumptions you hold about how computers work may not transfer from that operating system to Linux. A few paragraphs that discuss how Linux works (and particularly how it differs from the operating system you're used to) can be invaluable in getting off to a good start with your Red Hat Linux system.
- An introductory command overview (with examples) -- This is probably the most important thing to look for in Linux documentation. The design philosophy behind Linux is that it's better to use many small commands connected together in different ways than it is to have a few large (and complex) commands that do the whole job themselves. Without some examples that illustrate the Linux approach to doing things, you will find yourself intimidated by the sheer number of commands available on your Red Hat Linux system.

Here is some additional direction that may help to match all of your requirements:

- Books -- *Linux for Dummies*, by John "maddog" Hall, published by IDG; *Using Linux*, by William H. Ball, published by Que; *Running Linux*, by Matt Welsh and Lar Kaufman, published by O'Reilly & Associates; *Red Hat Linux Secrets*, by Naba Barkakati, published by IDG.
- Red Hat's website -- At our very own website (http://www.redhat.com), you'll find links to the Linux Documentation Project (LDP), the *Official Red Hat Linux Installation Guide*, the *Official Red Hat Linux Getting Started Guide* FAQs (Frequently Asked Questions), a database which can help you search for a Linux Users Group near you, a knowledgebase of information, and more. In short, you'll find a wealth of information to help you get started.
- Newsgroups -- Linux users are second to none when it comes to helping new users understand Linux. You can find dozens of Linux-related newsgroups on the Usenet, but a quick search through Deja.com (http://www.deja.com) shows:

- linux.help
- linux.redhat
- linux.redhat.digest
- linux.redhat.misc
- linux.redhat.rpm

Also, from the Deja.com website, you can frequently search for specific information from Linux newsgroups.

Continue reading the next section to find out more about the kinds of documentation that will help you at that point.

For the More Experienced

If you've used other Linux distributions, you probably already have a basic grasp of the most frequently used commands. You may have installed your own Linux system, and maybe you've even downloaded and built software you found on the Internet. What sorts of information will you need?

- Task-oriented items -- Many times, you will find that you would like to configure your Red Hat Linux system in a certain way, but you're not sure where to begin. In this case, it's often a big help to see what others in similar circumstances have done. This is where the Linux Documentation Project (also known as the LDP) can come in handy. Each of their HOWTOs document a particular aspect of Linux, from low-level kernel esoterica, to using Linux for amateur radio station work.

  If you selected one of the various `HOWTO` packages when you installed Red Hat Linux, you'll find the HOWTOs on your system in `/usr/doc/HOWTO`.

Documentation for Linux Gurus

If you're a long-time Red Hat Linux user, you probably already know that the following pretty much says it all when it comes to documentation:

Use the Force -- Read the source!

There are times when you'll just have to sit there and look at the sources to understand things. Fortunately, because of the freely available nature of Linux, it's easy to get the sources. Now if it were only that easy to understand them…

More to Come

The *Official Red Hat Linux Reference Guide* is part of the Red Hat's growing commitment to provide useful and timely support to Red Hat Linux users. Future editions will feature expanded information on system administration, console tools and other resources to help you extend the power of your Red Hat Linux system -- and yourself.

That's also where you come in.

Send in Your Feedback

If you'd like to make suggestions about the *Official Red Hat Linux Reference Guide*, please mention this guide's identifier:

RefGuide(EN)-6.2-Print-RHI (02/00)

You can send mail to:

`docs@redhat.com`

Sign Up for Support

If you have an official edition of Red Hat Linux 6.2, please remember to sign up for the benefits you're entitled to as a Red Hat customer.

You'll be entitled to any or all of the following benefits, depending upon the Official Red Hat Linux product you purchased:

- Official Red Hat support -- Get help with your installation questions from Red Hat, Inc.'s support team.
- Priority FTP access -- No more late-night visits to congested mirror sites. Owners of Red Hat Linux 6.2 receive free access to priority.redhat.com, Red Hat's preferred customer FTP service, offering high bandwidth connections day and night.

- Red Hat Update Agent -- Receive e-mail directly from Red Hat as soon as updated RPMs are available. Use Update Agent filters to receive notification and quickly download updated packages about those subjects that interest you. Also receive -- automatically -- kernel updates, security updates and other packages.
- Under the Brim: The Official Red Hat E-Newsletter -- Every month, get the latest news and product information directly from Red Hat.

To sign up, go to http://www.redhat.com/now. You'll find your **Personal Product ID** on a red and white card in your Official Red Hat Linux box.

To read more about technical support for Official Red Hat Linux, refer to the Appendix in the Official Red Hat Linux Installation Guide.

Good luck, and thank you for choosing Red Hat Linux!!

The Red Hat Documentation Team

Part I System-Related Reference

1 Red Hat Linux 6.2 New Features

This chapter describes features that are new to Red Hat Linux 6.2.

1.1 Installation-related Enhancements

The Red Hat Linux 6.2 installation program includes a number of new features. For more information, please refer to the *Official Red Hat Linux Installation Guide*.

1.2 System-Related New Features

There are many new features to Red Hat Linux 6.2 that are not part of the installation process. Some new features are tools or applications that you can use, others are new versions of the kernel or desktop environments. This list will provide a little more information about what to expect from Red Hat Linux 6.2 once you are actually using the OS itself.

Network services split into client, server packages:

The following network services have been split into client and server packages --

- telnet
- finger
- talk
- rsh
- rusers
- rwall
- tftp

For example, on the Intel-compatible platform telnet is now packaged in two RPMs --

- `telnet-0.16-5.i386.rpm`

- `telnet-server-0.16-5.i386.rpm`

sysctl **now controls system settings:**

In Red Hat Linux 6.2, kernel options such as IPv4 forwarding and enabling and disabling of the "magic sysrq" keys is done via the sysctl program, as opposed to being controlled by the contents of files in `/etc/sysconfig`. The sysctl settings are stored in `/etc/sysctl.conf`, and are loaded at each boot by the command:

```
sysctl -p /etc/sysctl.conf
```

Here is a sample `/etc/sysctl.conf`:

```
                # Disables IPv4 packet forwarding
net.ipv4.ip_forward = 0
# Enables source route verification
# This drops packets that come in over interfaces they shouldn't;
# (for example, a machine on an external net claiming to be one on your
# local network)
net.ipv4.conf.all.rp_filter = 1
# Disables automatic defragmentation
# Automatic defragmentation is needed for masquerading and Linux
# Virtual Server use; it is not needed otherwise.
net.ipv4.ip_always_defrag = 0
# Disables the magic-sysrq key
kernel.sysrq = 0
# Disables stop-a on the sparc
kernel.stop-a = 0
```

Many other kernel tunable parameters can be set; to see the full list, run `sysctl -a`, or look in the file `/usr/doc/kernel-doc-<version>/sysctl`.

Please Note

If you are upgrading your Red Hat Linux system to Red Hat Linux 6.2, any changes you have made to files in `/etc/sysconfig` will be migrated to `/etc/sysctl.conf`. You should confirm this by reviewing the file.

Linux 2.2.14 kernel:

Red Hat Linux 6.2 includes the latest stable version of the 2.2.*x* Linux kernel.

ident **service now run as daemon:**

The ident service is now run as a stand-alone service (called "identd"), and is controlled by settings in the `/etc/identd.conf` file.

Workstation-class installation now more secure:

Workstation-class installations no longer install the inetd "super server." This means that the following network-related services will not be available if you perform a workstation-class installation:

- ftp
- telnet
- shell
- login
- talk
- finger

If you require the above network-related services, you should consider an installation type other than workstation-class.

XFree86 **version 3.3.6:**

Red Hat Linux 6.2 contains the latest version of XFree86 (version 3.3.6 which supports many new drivers).

Services no longer run by default:

To permit a more fine-tuned system configuration, Red Hat Linux 6.2 no longer runs the following services by default --

- The automount daemon amd (in the `am-utils` RPM)
- The bind name server
- The dhcpd DHCP server
- The inn news server
- The knfsd NFS server
- The samba CIFS server used to support Windows-based file and print services

To enable these services, either use `chkconfig --level 35 <service-name> on`, or edit the list of services to start with the ntsysv or tksysv utilities.

Mesa graphics library now included:

The Mesa 3-D graphics library (version 3.2) is now included. Mesa is compatible with the OpenGL graphics API.

Sawmill **window manager now included:**

The sawmill window manager is now included in Red Hat Linux 6.2. Based on a Lisp-like language, sawmill is extensible, and GNOME-aware.

Man pages now compressed:

All man pages are now compressed (using gzip) to save disk space.

Starting programs at X startup:

By placing scripts in `/etc/X11/xinitrc/xinitrc.d`, it is now possible to automatically start programs whenever X starts.

Fonts recognized automatically:

X fonts that have been added to your Red Hat Linux 6.2 system will now be recognized automatically, when the font server starts. This can be done during an X session by issuing the following command as root:

```
/etc/rc.d/init.d/xfs restart
```

Encryption-related changes:

Due to relaxation of U.S. encryption laws, encryption-related changes have been made to the following packages:

- Kerberos authentication has been added to mutt, pine, fetchmail, cvs, and imap.
- The GNU Privacy Guard (GnuPG) is now included in all editions of Red Hat Linux 6.2.
- For those platforms that support it, Netscape Communicator built with 128-bit encryption is included in Red Hat Linux 6.2.

1.3 Miscellaneous New Features

These new features defy categorization:

Packages moved to PowerTools --

The following packages have been moved to PowerTools:

- The dosemu DOS emulator
- Version 1 of the fvwm window manager
- The aKtion and xanim movie viewers
- The mxp fractal generator
- The xwpick window grabber
- The xearth eye candy application

Changes to termcap **and** terminfo **entries --**

The termcap and terminfo entries have been changed to make the actions of the following keys more consistent:

- [Backspace]
- [Delete]
- [Home]
- [End]

Changes may be made by editing your `.inputrc` file.

DocBook support --

Red Hat Linux 6.2 includes support for editing and processing SGML documents written to use the DocBook DTD.

New documentation CD-ROM --

Red Hat Linux 6.2 boxed sets now include a documentation CD-ROM. The CD-ROM can be used in two ways:

- To install RPM-packaged documentation on your Red Hat Linux system.
- To read the documentation directly from the CD-ROM. For more information, please see the `README` on the documentation CD-ROM.

Colorized `ls` command --

The `ls` command is now colorized by default. To turn off this feature, add the command `unalias ls` in your `.bashrc` file, or (to disable color on a system-wide basis) delete the `colorls.*` files in `/etc/profile.d/`.

Deprecated features and packages --

The following features and packages are deprecated, and may not be supported or included in future releases of Red Hat Linux:

- The AnotherLevel environment

- The wmconfig dynamic window manager configuration tool
- The svgalib graphics library
- The Red Hat Linux version 5.2 compatibility development environment
- The mars-nwe NetWare server emulator
- The BSD lpr printing system
- The libc5 compatibility runtime libraries
- Version 1.*x* of the Qt library
- The libjpeg6a Red Hat Linux 5.*x* compatibility library
- The iBCS program compatibility technology

2 System Administration

This chapter provides an overview of the Red Hat Linux system. Here, you will learn aspects that you may not know about the system and things that are somewhat different from other UNIX systems.

2.1 Filesystem Structure

Red Hat is committed to the **Filesystem Hierarchy Standard** (**FHS**), a collaborative document that defines the names and locations of many files and directories. We will continue to track the standard to keep Red Hat Linux compliant.

The current FHS document is the authoritative reference to any FHS compliant filesystem, but the standard leaves many areas undefined or extensible. In this section we provide an overview of the standard and a description of the parts of the filesystem not covered by the standard.

The complete standard can be viewed at:

http://www.pathname.com/fhs/

While compliance with the standard means many things, the two most important are compatibility with other compliant systems, and the ability to mount the `/usr` partition as read-only because it contains common executables and is not meant to be changed by users. Because of this, `/usr` can be mounted from the CD-ROM or from another machine via read-only NFS.

2.1.1 Overview of the FHS

The directories and files noted here are a small subset of those specified by the FHS document. Check the latest FHS document for the most complete information.

The `/dev` Directory

The `/dev` directory contains filesystem entries which represent devices that are attached to the system. These files are essential for the system to function properly.

The /etc Directory

The /etc directory is reserved for configuration files that are local to your machine. No binaries are to be put in /etc. Binaries that were in the past put in /etc should now go into /sbin or possibly /bin.

The X11 and skel directories should be subdirectories of /etc:

```
/etc
  |- X11
  +- skel
```

The X11 directory is for X11 configuration files such as XF86Config. The skel directory is for "skeleton" user files, which are files used to populate a home directory when a user is first created.

The /lib Directory

The /lib directory should contain only those libraries that are needed to execute the binaries in /bin and /sbin.

The /proc Directory

The /proc directory contains special files that either extract information or send information to the kernel. It is an easy method of accessing information about the operating system using the cat command.

The /sbin Directory

The /sbin directory is for executables used only by the root user, and only those executables needed to boot and mount /usr and perform system recovery operations. The FHS says:

> "/sbin typically contains files essential for booting the system in addition to the binaries in /bin. Anything executed after /usr is known to be mounted (when there are no problems) should be placed in /usr/sbin. Local-only system administration binaries should be placed into /usr/local/sbin."

At a minimum, the following programs should be in /sbin:

```
arp, clock, getty, halt, init, fdisk,
fsck.*, ifconfig, lilo, mkfs.*, mkswap, reboot,
```

```
route, shutdown, swapoff, swapon, update
```

The /usr Directory

The /usr directory is for files that can be shared across a whole site. The /usr directory usually has its own partition, and it should be mountable read-only. The following directories should be subdirectories of /usr:

```
/usr
  |- X11R6
  |- bin
  |- dict
  |- doc
  |- etc
  |- games
  |- include
  |- info
  |- lib
  |- libexec
  |- local
  |- man
  |- sbin
  |- share
  +- src
```

The X11R6 directory is for the X Window System (XFree86 on Red Hat Linux), bin is for executables, doc is for random, non-man-page documentation, etc is for site-wide configuration files, include is for C header files, info is for GNU info files, lib is for libraries, man is for man pages, sbin is for system administration binaries (those that do not belong in /sbin), and src is for source code.

The /usr/local Directory

The FHS says:

> "The /usr/local hierarchy is for use by the system administrator when installing software locally. It needs to be safe from being overwritten when the system software is updated. It may be used for programs and data that are shareable amongst a group of machines, but not found in /usr."

The `/usr/local` directory is similar in structure to the `/usr` directory. It has the following subdirectories, which are similar in purpose to those in the `/usr` directory:

```
/usr/local
       |- bin
       |- doc
       |- etc
       |- games
       |- info
       |- lib
       |- man
       |- sbin
       +- src
```

The `/var` Directory

Since the FHS requires that you be able to mount `/usr` read-only, any programs that write log files or need `spool` or `lock` directories probably should write them to the `/var` directory. The FHS states `/var` is for:

> "…variable data files. This includes spool directories and files, administrative and logging data, and transient and temporary files."

The following directories should be subdirectories of `/var`:

```
/var
  |- catman
  |- lib
  |- local
  |- lock
  |- log
  |- named
  |- nis
  |- preserve
  |- run
  +- spool
       |- anacron
       |- at
       |- cron
       |- fax
       |- lpd
       |- mail
       |- mqueue
       +- news
```

```
        |- rwho
        |- samba
        |- slrnpull
        |- squid
        |- up2date
        |- uucp
        |- uucppublic
        |- vbox
        |- voice
    |- tmp
```

System log files such as `wtmp` and `lastlog` go in `/var/log`. The `/var/lib` directory also contains the RPM system databases. Formatted man pages go in `/var/catman`, and lock files go in `/var/lock`. The `/var/spool` directory has subdirectories for various systems that need to store data files.

2.1.2 `/usr/local` in Red Hat Linux

In Red Hat Linux, the intended use for `/usr/local` is slightly different from that specified by the FHS. The FHS says that `/usr/local` should be where you store software that is to remain safe from system software upgrades. Since system upgrades from Red Hat are done safely with the RPM system and Gnome-RPM, you don't need to protect files by putting them in `/usr/local`. Instead, we recommend you use `/usr/local` for software that is local to your machine.

For instance, let's say you have mounted `/usr` via read-only NFS from *beavis*. If there is a package or program you would like to install, but you are not allowed to write to *beavis*, you should install it under `/usr/local`. Later perhaps, if you've managed to convince the system administrator of *beavis* to install the program on `/usr`, you can uninstall it from `/usr/local`.

2.2 Special Red Hat File Locations

In addition to the files pertaining to the RPM system that reside in `/var/lib/rpm` (see Chapter 6, *Package Management with RPM* for more information on RPM), there are two other special locations that are reserved for Red Hat Linux configuration and operation.

The control-panel and related tools puts many scripts, bitmaps and text files in `/usr/lib/rhs`. There is probably nothing here that you would want to edit.

The other location, `/etc/sysconfig`, stores configuration information. The major users of the files in this directory are the scripts that run at boot time. It is possible to edit these by hand, but it would be better to use the proper control-panel tool.

2.3 Users, Groups and User-Private Groups

Managing users and groups has traditionally been tedious, but Red Hat Linux has a few tools and conventions that make users and groups easier to manage.

While you can use `useradd` to create a new user from the shell prompt, the easiest way to manage users and groups is through Linuxconf (see Chapter 3, *System Configuration*).

Next, we'll discuss the basic structure behind managing users and groups.

2.3.1 Standard Users

In Table 2–1, *Standard Users*, you'll find the standard users set up by the installation process (this is essentially the `/etc/passwd` file). The **Group ID** (GID) in this table is the *primary group* for the user. See Section 2.3.3, *User Private Groups* for details on how groups are used.

Table 2–1 Standard Users

User	UID	GID	Home Directory	Shell
root	0	0	`/root`	`/bin/bash`
bin	1	1	`/bin`	
daemon	2	2	`/sbin`	
adm	3	4	`/var/adm`	
lp	4	7	`/var/spool/lpd`	
sync	5	0	`/sbin`	`/bin/sync`
shutdown	6	0	`/sbin`	`/sbin/shutdown`

User	UID	GID	Home Directory	Shell
halt	7	0	`/sbin`	`/sbin/halt`
mail	8	12	`/var/spool/mail`	
news	9	13	`/var/spool/news`	
uucp	10	14	`/var/spool/uucp`	
operator	11	0	`/root`	
games	12	100	`/usr/games`	
gopher	13	30	`/usr/lib/go-pher-data`	
ftp	14	50	`/home/ftp`	
nobody	99	99	`/`	

2.3.2 Standard Groups

In Table 2–2, *Standard Groups*, you'll find the standard groups as set up by the installation process (this is essentially the `/etc/group` file).

Table 2–2 Standard Groups

Group	GID	Members
root	0	root
bin	1	root, bin, daemon
daemon	2	root, bin, daemon
sys	3	root, bin, adm
adm	4	root, adm, daemon
tty	5	
disk	6	root
lp	7	daemon, lp

Group	GID	Members
mem	8	
kmem	9	
wheel	10	root
mail	12	mail
news	13	news
uucp	14	uucp
man	15	
games	20	
gopher	30	
dip	40	
ftp	50	
nobody	99	
users	100	
floppy	19	

2.3.3 User Private Groups

Red Hat Linux uses a **user private group** (UPG) scheme, which makes UNIX groups much easier to use. The UPG scheme does not add or change anything in the standard UNIX way of handling groups. It simply offers a new convention for handling groups. Whenever you create a new user, by default, he or she has a unique group. The scheme works as follows:

User Private Group

Each user has their own primary group, of which only they are a member.

umask = 002

The traditional UNIX umask is 022, which prevents other users *and other members of a user's primary group* from modifying a user's files. Since every user has their own private group in the UPG scheme, this "group protection" is not needed. A umask of 002 will prevent users from modifying other users' private files. The umask is set in `/etc/profile`.

setgid bit on Directories

If you set the setgid bit on a directory (with `chmod g+s` *`directory`*), files created in that directory will have their group set to the directory's group.

Most IT organizations like to create a group for each major project and assign people to the groups they should be in. Managing files traditionally has been difficult, though, because when someone creates a file it is owned by the primary group he or she belongs to. When a single person works on multiple projects, it becomes hard to associate the right files to the right ownership group. In the UPG scheme, groups are automatically assigned to files on a project-by-project basis, which makes managing group projects very simple.

Let's say you have a big project called *devel*, with many people editing the devel files in a `devel` directory. Make a group called `devel`, `chgrp` the `devel` directory to `devel`, and add the all the devel users to the `devel` group. Now, all devel users will be able to edit the devel files and create new files in the `devel` directory, and these files will always retain their `devel` group. Thus, they will always be edit-able by other devel users.

If you have multiple projects like *devel*, and users who are working on multiple projects, these users will never have to change their umask or group when they move from project to project. The setgid bit on each project's main directory "selects" the proper group.

Since each user's home directory is owned by the user and their private group, it is safe to set the setgid bit on the home directory. However, by default, files are created with the primary group of the user, so the setgid bit would be redundant.

User Private Group Rationale

Although UPG is not new to Red Hat Linux 6.2, many people still have questions about it, such as why UPG is necessary. The following is the rationale for the scheme.

- You'd like to have a group of people work on a set of files in say, the `/usr/lib/emacs/site-lisp` directory. You trust a few people to mess around in there, but certainly not everyone.
- So you enter:

  ```
  chown -R root.emacs /usr/lib/emacs/site-lisp
  ```

  and you add the proper users to the group.
- To allow the users to actually create files in the directory you enter:

  ```
  chmod 775 /usr/lib/emacs/site-lisp
  ```

- But when a user creates a new file it is assigned the group of the user's default group (usually `users`). To prevent this you enter:

  ```
  chmod 2775 /usr/lib/emacs/site-lisp
  ```

  which causes everything in the directory to be created with the "emacs" group.
- But the new file needs to be mode 664 for another user in the emacs group to be able to edit it. To do this you make the default umask 002.
- Well, this all works fine, except that if your default group is "users," every file you create in your home directory will be writable by everybody in "users" (usually everyone).
- To fix this, you make each user have a "private group" as their default group.

At this point, by making the default umask 002 and giving everyone a private default group, you can easily set up groups that users can take advantage of without doing any magic. Just create the group, add the users, and do the above `chown` and `chmod` on the group's directories.

2.4 Configuring Console Access

When normal (non-root) users log in to a computer locally, they are given two types of special permission: they can run certain programs that they would not otherwise

be able to run, and they can access certain files (normally special device files used to access diskettes, CD-ROMS, and so on) that they would not otherwise be able to access.

Since there are multiple consoles on a single computer, and multiple users can be logged into the computer locally at the same time, one of the users has to "win" the fight to access the files. The first user to log in at the console owns those files. Once the first user logs out, the next user who logs in will own the files.

In contrast, *every* user who logs in at the console will be allowed to run programs normally restricted to the root user. By default, those programs will ask for the user's password. This will be done graphically if X is running which makes it possible to include these actions as menu items in a graphical user interface. As shipped, the console-accessible programs are `shutdown`, `halt`, and `reboot`.

2.4.1 Disabling Console Program Access

In environments where the console is otherwise secured (BIOS and LILO passwords are set, [Ctrl]-[Alt]-[Delete] is disabled, the power and reset switches are disabled, etc.), it may not be desirable to allow arbitrary users at the console to run `shutdown`, `halt`, and `reboot`.

In order to disable all access by console users to console programs, you should run the command:

```
rm -f /etc/security/console.apps/*
```

2.4.2 Disabling All Console Access

In order to disable all console access, including program and file access, in the `/etc/pam.d/` directory, comment out all lines that refer to `pam_console.so`. The following script will do the trick:

```
cd /etc/pam.d
for i in * ; do
sed '/[^#].*pam_console.so/s/^/#/' < $i > foo && mv foo $i
done
```

2.4.3 Defining the Console

The `/etc/security/console.perms` file defines the console group. The syntax of that file is very flexible; you can edit the file so that these instructions no longer apply. However, the default file has a line that looks like this:

```
<console>=tty[0-9][0-9]* :[0-9]\.[0-9] :[0-9]
```

When users log in, they are attached to some sort of named terminal, either an X server with a name like `:0` or `mymachine.example.com:1.0`; or a device like `/dev/ttyS0` or `/dev/pts/2`. The default is to define that local virtual consoles and local X servers are considered local, but if you want to consider the serial terminal next to you on port `/dev/ttyS1` to also be local, you can change that line to read:

```
<console>=tty[0-9][0-9]* :[0-9]\.[0-9] :[0-9] /dev/ttyS1
```

2.4.4 Making Files Console-Accessible

In `/etc/security/console.perms`, there is a section with lines like:

```
<floppy>=/dev/fd[0-1]*
<cdrom>=/dev/cdrom
<jaz>=/dev/zip
```

You can also add your own lines:

```
<scanner>=/dev/sga
```

(Of course, make sure that `/dev/sga` is really your scanner and not, say, your hard drive.)

That's the first part. The second part is to define what is done with those files. Look in the last section of `/etc/security/console.perms` for lines similar to:

```
<console> 0660 <floppy> 0660 root.floppy
<console> 0600 <cdrom>  0600 root.disk
<console> 0600 <jaz>    0660 root.disk
```

and add a line like:

```
<console> 0600 <scanner> 0600 root
```

Then, when you log in at the console, you will be given ownership of the `/dev/sga` device and the permissions will be 0600 (readable and writable by you only). When you log out, the device will be owned by root and still have 0600 (now: readable and writable by root only) permissions.

2.4.5 Enabling Console Access for Other Applications

If you wish to make other applications besides `shutdown`, `reboot`, and `halt` accessible to console users, you will have to do just a little bit more work.

First of all, console access *only* works for applications which reside in `/sbin` or `/usr/sbin`, so the application that you wish to run must be there.

Create a link from the name of your application to the `/usr/bin/console-helper` application:

```
cd /usr/bin
ln -s consolehelper foo
```

Create the file `/etc/security/console.apps/foo`:

```
touch /etc/security/console.apps/foo
```

Create a PAM configuration file for the *foo* service in `/etc/pam.d/`. We suggest that you start with a copy of the shutdown service, then change it if you want to change the behavior:

```
cp /etc/pam.d/shutdown /etc/pam.d/foo
```

Now, when you run `/usr/bin/foo`, it will call `consolehelper`, which, with the help of `/usr/sbin/userhelper` will authenticate the user (asking for the user's password if `/etc/pam.d/foo` is a copy of `/etc/pam.d/shutdown`; otherwise, it will do precisely what is specified in `/etc/pam.d/foo`) and then run `/usr/sbin/foo` with root permissions.

2.5 The `floppy` Group

If, for whatever reason, console access is not appropriate for you, and you need to give non-root users access to your system's diskette drive, this can be done using the `floppy` group. Simply add the user(s) to the `floppy` group using the tool of your choice. Here's an example showing how `gpasswd` can be used to add user fred to the `floppy` group:

```
[root@bigdog root]# gpasswd -a fred floppy
Adding user fred to group floppy
[root@bigdog root]#
```

User fred will now be able to access the system's diskette drive.

2.6 User Authentication with PAM

Programs which give users access to privileges of any sort need to be able to authenticate the users. When you log into a system, you provide your name and password, and the login process uses those to authenticate the login -- to verify that you are who you say you are. Forms of authentication other than passwords are possible, and it is possible for the passwords to be stored in different ways.

PAM, which stands for **Pluggable Authentication Modules**, is a way of allowing the system administrator to set authentication policy without having to recompile authentication programs. With PAM, you control how the modules are plugged into the programs by editing a configuration file.

Most Red Hat Linux users will never need to touch this configuration file. When you use RPM to install programs that require authentication, they automatically make the changes that are needed to do normal password authentication. However, you may want to customize your configuration, in which case you must understand the configuration file.

2.6.1 PAM Modules

There are four types of modules defined by the PAM standard.

- `auth` modules provide the actual authentication, perhaps asking for and checking a password, and they set "credentials" such as group membership or kerberos "tickets."
- `account` modules check to make sure that the authentication is allowed (the account has not expired, the user is allowed to log in at this time of day, and so on).
- `password` modules are used to set passwords.
- `session` modules are used once a user has been authenticated to allow them to use their account, perhaps mounting the user's home directory or making their mailbox available.

These modules may be *stacked*, so that multiple modules are used. For instance, `rlogin` normally makes use of at least two authentication methods: if `rhosts` authentication succeeds, it is sufficient to allow the connection; if it fails, then standard password authentication is done.

New modules can be added at any time, and PAM-aware applications can then be made to use them. For instance, if you have a one-time-password calculator system, and you can write a module to support it (documentation on writing modules is included with the system in `/usr/doc/pam*`), PAM-aware programs can use the new module and work with the new one-time-password calculators without being recompiled or otherwise modified in any way.

2.6.2 Services

Each program using PAM defines its own "service" name. The login program defines the service type `login`, ftpd defines the service type `ftp`, and so on. In general, the service type is the name of the program used to *access* the service, not (if there is a difference) the program used to *provide* the service.

2.6.3 The Configuration Files

The directory `/etc/pam.d` is used to configure all PAM applications. (This used to be `/etc/pam.conf` in earlier PAM versions; while the `pam.conf` file is still

read if no /etc/pam.d/ entry is found, its use is deprecated.) Each application (really, each *service*) has its own file. A file looks like this:

```
#%PAM-1.0
auth      required  /lib/security/pam_securetty.so
auth      required  /lib/security/pam_pwdb.so shadow nullok
auth      required  /lib/security/pam_nologin.so
account   required  /lib/security/pam_pwdb.so
password  required  /lib/security/pam_cracklib.so
password  required  /lib/security/pam_pwdb.so shadow nullok use_authtok
session   required  /lib/security/pam_pwdb.so
```

The first line is a comment. (Any line that starts with a # character is a comment.) Lines two through four stack up three modules to use for login authorization. Line two makes sure that *if* the user is trying to log in as root, the tty on which they are logging in is listed in the /etc/securetty file *if* that file exists. Line three causes the user to be asked for a password and the password checked. Line four checks to see if the file /etc/nologin exists, and if it does, displays the contents of the file, and if the user is not root, does not let him or her log in.

Note that all three modules are checked, *even if the first module fails*. This is a security decision -- it is designed to prevent the user from knowing why their authentication was disallowed, because knowing why it was disallowed might allow them to break the authentication more easily. You can change this behavior by changing required to requisite; if any requisite module returns failure, PAM fails immediately without calling any other modules.

The fifth line causes any necessary accounting to be done. For example, if shadow passwords have been enabled, the pam_pwdb.so module will check to see if the account has expired, or if the user has not changed his or her password and the grace period for changing the password has expired.

The sixth line subjects a newly changed password to a series of tests to ensure that it cannot, for example, be easily determined by a dictionary-based password cracking program.

The seventh line (which may be wrapped) specifies that if the login program changes the user's password, it should use the pam_pwdb.so module to do so. (It will do

so only if an `auth` module has determined that the password needs to be changed --- for example, if a shadow password has expired.)

The eighth and final line specifies that the `pam_pwdb.so` module should be used to manage the session. Currently, that module doesn't do anything; it could be replaced (or supplemented by stacking) by any necessary module.

Note that the order of the lines within each file matters. While it doesn't really matter much in which order required modules are called, there are other *control flags* available. While optional is rarely used, and never used by default on a Red Hat Linux system, sufficient and requisite cause order to become important.

Let's look at the `auth` configuration for `rlogin`:

```
auth  required    /lib/security/pam_securetty.so
auth  sufficient  /lib/security/pam_rhosts_auth.so
auth  required    /lib/security/pam_pwdb.so shadow nullok
auth  required    /lib/security/pam_nologin.so
```

That looks *almost* like the `login` entry, but there's an extra line specifying an extra module, and the modules are specified in a different order.

First, `pam_securetty.so` keeps root logins from happening on insecure terminals. This effectively disallows all root `rlogin` attempts. If you wish to allow them (in which case we recommend that you not be Internet-connected or be behind a good firewall), you can simply remove that line.

Second, if `pam_rhosts_auth.so` authenticates the user, PAM immediately returns success to `rlogin` without any password checking. If `pam_rhosts_auth.so` fails to authenticate the user, that failed authentication is ignored.

Third, if `pam_rhosts_auth.so` has failed to authenticate the user, the `pam_pwdb.so` module performs normal password authentication.

Finally `pam_nologin.so` checks `/etc/nologin`, as specified above.

Note that if you do not want to prompt for a password if the securetty check fails, you can change the `pam_securetty.so` module from required to requisite.

2.6.4 Shadow Passwords

The `pam_pwdb.so` module will automatically detect that you are using shadow passwords and make all necessary adjustments. Please refer to Section 2.7, *Shadow Utilities* for more information.

2.6.5 Rexec and PAM

For security reasons, rexec is not enabled in Red Hat Linux 6.2. Should you wish to enable it, you will need to comment out one line in the file `/etc/pam.d/rexec`. Here is a sample of the file (note that your file may differ slightly):

```
#%PAM-1.0
auth       required     /lib/security/pam_pwdb.so shadow nullok
auth       required     /lib/security/pam_nologin.so
account    required     /lib/security/pam_pwdb.so
```

To enable rexec, the line referring to the `pam_nologin.so` module must be commented out:

```
#%PAM-1.0
auth       required     /lib/security/pam_pwdb.so shadow nullok
#auth      required     /lib/security/pam_nologin.so
account    required     /lib/security/pam_pwdb.so
```

After this file is modified, rexec will be enabled.

Please Note

If your `/etc/pam.d/rexec` file contains a line referring to the `pam_securetty.so` module, you will not be able to `rexec` as root. To do so, you must also comment out the line referring to the `pam_securetty.so` module.

More Information

This is just an introduction to PAM. More information is included in the `/usr/doc/pam*` directory, including a *System Administrators' Guide*, a *Module*

Writers' Manual, an *Application Developers' Manual*, and the PAM standard, DCE-RFC 86.0.

2.7 Shadow Utilities

Shadow passwords are a method of improving system security by moving the encrypted passwords (normally found in `/etc/passwd`) to `/etc/shadow` which is readable only by root. During the installation of Red Hat Linux, you were given the option of setting up shadow password protection on your system.

The `shadow-utils` package contains a number of utilities that support:

- Conversion from normal to shadowed passwords and back (`pwconv`, `pwunconv`)
- Verification of the password, group, and associated shadow files (`pwck`, `grpck`)
- Industry-standard methods of adding, deleting and modifying user accounts (`useradd`, `usermod`, and `userdel`)
- Industry-standard methods of adding, deleting, and modifying user groups (`groupadd`, `groupmod`, and `groupdel`)
- Industry-standard method of administering the `/etc/group` file (`gpasswd`)

Please Note

There are some additional points of interest concerning these utilities:

- The utilities will work properly whether shadowing is enabled or not.
- The utilities have been slightly modified to support Red Hat's user private group scheme. For a description of the modifications, please see the `useradd` man page. For more information on user private groups, please turn to Section 2.3.3, *User Private Groups*.
- The `adduser` script has been replaced with a symlink to `/usr/sbin/useradd`.

2.8 Building a Custom Kernel

Many people new to Linux often ask, "why should I build my own kernel?" Given the advances that have been made in the use of kernel modules, the most accurate response to that question is, "unless you know why you need to build your own kernel, you probably don't." So unless you have a specific reason to build a customized kernel (or you're just the curious sort), you may skip ahead to Section 2.9, *Sendmail*.

In the past, you would need to recompile the kernel if you added new hardware on your system. The kernel was, in other words, **static**. Improvements in the Linux 2.0.*x* kernels allowed for much of the drivers for hardware to be **modularized** into components that could only be inserted on demand. However, there were major problems with having multiple kernels on your system that had been compiled for different advancements (a good case being SMP versus UP kernels). Further advancements with the modularization of the Linux 2.2.x kernel have allowed for multiple kernels to more easily co-exist (though *not* share modules).

For information on handling kernel modules see Section 3.2.2, *Loading Kernel Modules*. Most of the changes are hidden except when recompiling a customized kernel for your system.

2.8.1 Building a modularized kernel

These instructions enable you to take advantage of the power and flexibility available through kernel modularization. If you do not wish to take advantage of modularization, please see Section 2.8.3, *Building a monolithic kernel* for an explanation of the different aspects of building and installing a monolithic kernel. It's assumed that you've already installed the `kernel-headers` and `kernel-source` packages and that you issue all commands from the `/usr/src/linux` directory.

The most important step is to make sure that you have a working emergency boot disk in case you make a mistake below. If you didn't make a boot disk during the installation, use the `mkbootdisk` command to make one. The standard command is similar to `mkbootdisk --device /dev/fd0 2.2.x`. Once done, test the boot disk to make sure that it will boot the system.

It is important to begin a kernel build with the source tree in a known condition. Therefore, it is recommended that you begin with the command `make mrproper`. This will remove any configuration files along with the remains of any previous builds that may be scattered around the source tree. Now you must create a configuration file that will determine which components to include in your new kernel. Available methods for kernel configuration are listed below:

- `make config` -- An interactive text program. Components are presented and you answer with **Y** (yes), **N** (no), or **M** (module).
- `make menuconfig` -- A graphical, menu driven program. Components are presented in a menu of categories, you select the desired components in the same manner used in the Red Hat Linux installation program. Toggle the tag corresponding to the item you want included; **Y** (yes), **N** (no), or **M** (module).
- `make xconfig` -- An X Window System program. Components are listed in different levels of menus, and are selected using a mouse. Again, select **Y** (yes), **N** (no), or **M** (module).

- `make oldconfig` -- This is a non-interactive script that will set up your Makefile to be the default settings. If you're using the Red Hat patched kernel, it will set up the configuration to be that of the kernel that is shipped for your box. This is useful in setting up your kernel to known working defaults and then turning off features that you don't want.

Please Note

In order to use `kmod` (see Section 3.2.2, *Loading Kernel Modules* for details) and kernel modules you must answer **`Yes`** to `kmod support` and `module version (CONFIG_MODVERSIONS) support` in the configuration.

If you wish to build a kernel with a configuration file (`/usr/src/linux/.config` -- this file is created once one of the above methods has been performed) that you have already created with one of the above methods, you can omit the `make mrproper` and `make config` commands and use the command `make dep` followed by `make clean` to prepare the source tree for the build.

The next step in making a modularized kernel is to simply edit `/usr/src/linux/Makefile` and compile the source code components into a working program that your machine can use to boot. The method described here is the easiest to recover from in the event of a mishap. If you are interested in other possibilities, details can be found in the Kernel-HOWTO or in the `Makefile` in `/usr/src/linux` on your Linux system.

- Edit the `Makefile` and change the line: `EXTRAVERSION =` to match a "unique" name (such as adding your initials to the end of the string, as in `EXTRAVERSION = -2.5.0sjs`). This will allow you to have the old working kernel and the new kernel on your system at the same time.
- Build the kernel with `make bzImage`.
- Build any modules you configured with `make modules`.

- Install the new modules (even if you didn't build any) with `make modules_install`. This will install the kernel modules into the filepath `/lib/modules/` using the path name that was specified in the `Makefile`. Our example would be `/lib/modules/2.2.15-2.5.0sjs/`.

If you have a SCSI adapter and made your SCSI driver modular, build a new `initrd` image (see Section 2.8.2, *Making an initrd image*; note that there are few practical reasons to make the SCSI driver modular in a custom kernel). Unless you have a specific reason to create an `initrd` image, do not create one and do not add it to `lilo.conf`.

In order to provide a redundant boot source to protect from a possible error in a new kernel you should keep the original kernel available. Adding a kernel to the LILO menu is as simple as renaming the original kernel in `/boot`, copying the new kernel to `/boot`, adding a few lines in `/etc/lilo.conf` and running `/sbin/lilo`. Here is an example of the default `/etc/lilo.conf` file shipped with Red Hat Linux:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
default=linux

linear

image=/boot/vmlinuz-2.2.15-2.5.0
     label=linux
     initrd=/boot/initrd-2.2.15-2.5.0.img
     read-only
     root=/dev/hda8

other=/dev/hda1
        label=dos
```

Now you must update `/etc/lilo.conf`. If you built a new `initrd` image you must tell LILO to use it. In this example of `/etc/lilo.conf` we have added four lines in the middle of the file to indicate another kernel to boot from. We have

renamed `/boot/vmlinuz` to `/boot/vmlinuz.old` and changed its label to `old`. We have also added an `initrd` line for the new kernel:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
default=linux

linear

image=/boot/vmlinuz-2.2.15-2.5.0
     label=linux
     initrd=/boot/initrd-2.2.15-2.5.0.img
     read-only
     root=/dev/hda8

image=/boot/vmlinuz-2.2.15-2.5.0sjs
        label=test
        initrd=/boot/initrd-2.2.15-2.5.0sjs.img
        read-only
        root=/dev/hda8

other=/dev/hda1
        label=dos
```

Now when the system boots and you press [Tab] at the LILO `boot:` prompt, available choices will be shown;

```
LILO boot:
linux  test  dos
```

To boot the old kernel (**`linux`**) simply press [Enter], or wait for LILO to time out. If you want to boot the new kernel (**`test`**), type **`test`** and press [Enter].

Here is a summary of the steps:

- Copy the resulting compiled kernel into your `/boot` directory using the name that resulted from your earlier changes to the `Makefile`. Here is an example:

```
cp -p
/usr/src/linux/arch/i386/boot/bzImage
```

```
/boot/vmlinuz-2.2.15-2.5.0sjs
```

- Edit `/etc/lilo.conf`.
- Make a new initial ramdisk, `initrd` image (see Section 2.8.2, *Making an initrd image*) if needed.
- Run `/sbin/lilo`. You can add a `-v` flag to `lilo` to get more verbose reporting if you think there might be a problem.

You can begin testing your new kernel by rebooting your computer and watching the messages to ensure your hardware is detected properly.

2.8.2 Making an initrd image

An `initrd` image is needed for loading your SCSI module at boot time. If you do not need an `initrd` image, do not make one and do not edit `lilo.conf` to include this image.

The shell script `/sbin/mkinitrd` can build a proper `initrd` image for your machine if the following conditions are met:

- The loopback block device is available.
- The `/etc/conf.modules` file has a line for your SCSI adapter; for example:

```
alias scsi_hostadapter BusLogic
```

To build the new `initrd` image, run `/sbin/mkinitrd` with parameters such as this:

```
/sbin/mkinitrd /boot/newinitrd-image 2.2.15
```

Where `/boot/newinitrd-image` is the file to use for your new image, and `2.2.15` is the kernel whose modules (from `/lib/modules`) should be used in the `initrd` image (not necessarily the same as the version number of the currently running kernel).

2.8.3 Building a monolithic kernel

To build a monolithic kernel you follow the same steps as building a modularized kernel with a few exceptions.

- When configuring the kernel only answer **`Yes`** and **`No`** to the questions (don't make anything modular). Also, you should answer **`No`** to `kmod support` and `module version (CONFIG_MODVERSIONS) support` in the configuration.
- Omit the steps:

```
make modules
make modules_install
```

- Edit `lilo.conf` and add the line `append=nomodules`.

2.9 Sendmail

A default `sendmail.cf` file will be installed in `/etc`. The default configuration should work for most SMTP-only sites (Simple Mail Transfer Protocol) . It will *not* work for UUCP (Unix to UNIX Copy Protocol) sites; you will need to generate a new `sendmail.cf` if you must use UUCP mail transfers.

Please Note

Although SMTP servers are supported automatically, **IMAP** (Internet Message Access Protocol) servers are not. If your ISP uses an IMAP server rather than an SMTP sever, you must install the IMAP package. Without it, your system won't know how to pass information to the IMAP server or retrieve your mail.

To generate a new `sendmail.cf`, you will need to install `m4` and the `sendmail` source package. Read the `README` file in the sendmail sources for more details on

creating sendmail configuration files. Also, O'Reilly & Associates publishes a good sendmail reference entitled *sendmail*, by Bryan Costales.

One common sendmail configuration is to have a single machine act as a mail gateway for all the machines on your network. For instance, at Red Hat we have a machine `mail.redhat.com` that does all our mail. On that machine we simply need to add the names of machines for which mail.redhat.com will handle mail to `/etc/sendmail.cw`. Here is an example:

```
# sendmail.cw - include all aliases for your machine
# here.
torgo.redhat.com
poodle.redhat.com
devel.redhat.com
```

Then on the other machines, `torgo`, `poodle`, and `devel`, we need to edit `/etc/sendmail.cf` to "masquerade" as `mail.redhat.com` when sending mail, and to forward any local mail processing to `redhat.com`. Find the `DH` and `DM` lines in `/etc/sendmail.cf` and edit them as such:

```
# who I send unqualified names to
# (null means deliver locally)
DRmail.redhat.com

# who gets all local email traffic
DHmail.redhat.com

# who I masquerade as (null for no masquerading)
DMredhat.com
```

With this type of configuration, all mail sent will appear as if it were sent from `redhat.com`, and any mail sent to `torgo.redhat.com` or the other hosts will be delivered to `mail.redhat.com`.

Please be aware that if you configure your system to masquerade as another any e-mail sent from your system to your system will be sent to the machine you are masquerading as. For example, in the above illustration, log files that are periodically sent to `root@poodle.redhat.com` by the `cron` daemon would be sent to `root@mail.redhat.com`.

2.10 Controlling Access to Services

As a security measure, most network services are managed by a protective program called a **TCP wrapper**. The protected services are those listed in `/etc/inetd.conf` that specify `/usr/sbin/tcpd`. `tcpd` can allow or deny access to a service based on the origin of the request, and the configuration in `/etc/hosts.allow` and `/etc/hosts.deny`.

By default Red Hat Linux allows all service requests. To disable or limit services you can edit `/etc/hosts.allow`. Here is an example `/etc/hosts.allow` file:

```
ALL: redhat.com .redhat.com
in.talkd: ALL
in.ntalkd: ALL
in.fingerd: ALL
in.ftpd: ALL
```

This configuration allows all connections from redhat.com and `*.redhat.com` machines. It also allows talk, finger, and FTP requests from all machines.

`tcpd` allows much more sophisticated access control, using a combination of `/etc/hosts.allow` and `/etc/hosts.deny`. Read the tcpd(8) and hosts_access(5) man pages for complete details.

2.11 Anonymous FTP

Setting up anonymous FTP is simple. All you need to do is install the anon-ftp RPM package (which you may have already done at install time). Once it is installed, anonymous FTP will be up and running.

There are a few files you might wish to edit to configure your FTP server.

`/etc/ftpaccess`

This file defines most of the access control for your FTP server. Some of the things that you can do are: set up logical "groups" to control access from different sites, limit the number of simultaneous FTP connections, configure transfer logging, and much more. Read the ftpaccess man page for complete details.

`/etc/ftphosts`

The `ftphosts` file is used to allow or deny access to certain accounts from various hosts. Read the ftphosts man page for details.

`/etc/ftpusers`

This file lists all the users that are *not* allowed to FTP into your machine. For example, `root` is listed in `/etc/ftpusers` by default. That means that you cannot FTP to your machine and log in as root. This is a good security measure, but some administrators prefer to remove `root` from this file.

2.12 NFS Configuration

NFS stands for *N*etwork *F*ile *S*ystem; it is a way to share files between machines as if they were on your local hard drive. Linux can be both an NFS server and an NFS client, which means that it can **export** filesystems to other systems, and **mount** filesystems exported from other machines.

2.12.1 Mounting NFS Filesystems

Use the `mount` command to mount an NFS filesystem from another machine:

```
mkdir /mnt/local # Only required if /mnt/local doesn't exist
mount bigdog:/mnt/export /mnt/local
```

In this command, bigdog is the hostname of the NFS fileserver, `/mnt/export` is the filesystem that bigdog is exporting, and `/mnt/local` is a directory on the local machine where we want to mount the filesystem. After the `mount` command runs (and if we have the proper permissions from `bigdog`) we can enter `ls /mnt/local` and get a listing of the files in `/mnt/export` on bigdog.

2.12.2 Exporting NFS Filesystems

The file that controls what filesystems you wish to export is `/etc/exports`. Its format is:

```
directory       hostname(options)
```

the (*`options`*) are discretional. For example:

```
/mnt/export     speedy.redhat.com
```

would allow speedy.redhat.com to mount `/mnt/export`, but:

```
/mnt/export     speedy.redhat.com(ro)
```

would just allow speedy to mount `/mnt/export` read-only.

Each time you change `/etc/exports`, you must tell the NFS daemons to examine it for new information. One simple way to accomplish this is to just stop and start the daemons:

```
/etc/rc.d/init.d/nfs stop
/etc/rc.d/init.d/nfs start
```

Or you can restart the daemons with this command:

```
/etc/rc.d/init.d/nfs restart
```

The following will also work:

```
killall -HUP rpc.nfsd rpc.mountd
```

See the following man pages for more details: nfsd(8), mountd(8), and exports(5). Another good reference is *Managing NFS and NIS Services*, by Hal Stern, published by O'Reilly & Associates.

2.13 The Boot Process, Init, and Shutdown

This section contains information on what happens when you boot or shut down your Red Hat Linux system.

2.13.1 Behind the Scenes of the Boot Process

When a computer is booted, the processor looks at the end of the system memory for the **BIOS** (Basic Input/Output System) and runs it. The BIOS program is written into read-only permanent memory, and is always ready to go. The BIOS provides the lowest level interface to peripheral devices and controls the first step of the boot process.

The BIOS tests the system, looks for and checks peripherals and then looks for a drive to boot from. Usually, it checks the floppy drive (or CD-ROM drive on many newer systems), if present, and then it looks on the hard drive. On the hard drive, the BIOS looks for a **Master Boot Record** (MBR) starting at the first sector on the first hard drive and starts the MBR running.

The MBR looks for the first active partition and reads the partition's boot record. The boot record contains instructions on how to load the boot loader, LILO (*LI*nux *LO*ader). The MBR then loads LILO and LILO takes over the process.

LILO reads the file `/etc/lilo.conf`, which spells out which operating system(s) to configure or which kernel to start and where to install itself (for example, `/dev/hda` for your hard drive). LILO displays a `LILO:` prompt on the screen and waits for a preset period of time (also set in the `lilo.conf` file) for input from the user. If your `lilo.conf` is set to give LILO a choice of operating systems, at this time you could type in the label for whichever OS you wanted to boot.

After waiting for a set period of time (five seconds is common), LILO proceeds to boot whichever operating system appears first in the `lilo.conf` file.

If LILO is booting Linux, it first boots the kernel, which is a `vmlinuz` file (plus a version number, for example, `vmlinuz-2.2.15-xx`) located in the `/boot` directory. Then the kernel takes over.

The kernel looks in several different places for init (`/sbin` is a common location) and runs the first one it finds. Init takes over.

Init starts (and becomes the parent or grandparent of) all of the processes which make up your Linux system. First, it runs `/etc/rc.d/rc.sysinit`, which sets your path, sets up networking if necessary, starts swapping, checks the filesystems, and so on. Basically, `rc.sysinit` is taking care of everything that your system needs to have done at system initialization. For example, on a networked system `rc.sysinit` uses the information in the `/etc/sysconfig/network` and `/etc/sysconfig/clock` files to initialize network processes and the clock. Itmay also run `rc.serial`, if you have serial port processes that need to be initialized.

Init looks at and implements the `/etc/inittab` file. The `/etc/inittab` file describes how the system should be set up in each runlevel and sets the default runlevel. This file states that `/etc/rc.d/rc` and `/sbin/update` should be run whenever a runlevel starts.

The `/sbin/update` file flushes dirty buffers back to disk.

Whenever the runlevel changes, `/etc/rc.d/rc` starts and stops services. First, rc sets the source function library for the system (commonly `/etc/rc.d/init.d/functions`), which spells out how to start/kill a program and how to find out the PID of a program. The rc file then finds out the current and the previous runlevel and tells linuxconf the appropriate runlevel.

The rc file starts all of the background processes necessary for the system to run, and looks for an rc directory for that runlevel (`/etc/rc.d/rc<x>.d`, where the <*x*> is numbered 0-6). rc kills all of the kill scripts (their file name starts with a K) in `/rc.d/`. Then it initializes all of the start scripts (their file names start with an S) in the appropriate runlevel directory (so that all services and applications are started correctly).

For example, for runlevel 5, rc looks into the `/etc/rc.d/rc5.d` directory and finds that it needs to kill rusersd, rwalld, rwhod, mcserv, mars-nwe, apmd, and pcmcia. In the bloody aftermath, rc looks into the same directory and finds start scripts for kmod, network, nfsfs, randomc, syslog, atd, crond, portmap, snmpd, inet, xntpd, lpd, nfs.rpmsave, dhcpd, ypbind, autofs, keytable, sendmail, gpm, sound, and smb.rpmsave. And life begins anew.

The `/etc/inittab` file forks a getty process for each virtual console (login prompts) for each runlevel (runlevels 2-5 get all six; runlevel 1, which is single user mode, only gets one console; runlevels 0 and 6 get no virtual consoles). `/etc/inittab` also states that gdm, the X logon manager, should start in runlevel 5.

Also, /etc/inittab describes how the system should handle translating [Ctrl]-[Alt]-[Delete] into something like the command `/sbin/shutdown -t3 -r now`. And finally, `/etc/inittab` states what the system should do in case of power failures.

At this point, you should be looking at a login prompt. All that, and it only took a few seconds.

Next, we'll discuss information on the files in `/etc/sysconfig`.

2.13.2 Sysconfig Information

The following information outlines the various files in `/etc/sysconfig`, their function, and their contents.

Files in `/etc/sysconfig`

The following files are normally found in `/etc/sysconfig`:

- `/etc/sysconfig/clock`
- `/etc/sysconfig/hwconf` (this should be ignored for editing)
- `/etc/sysconfig/mouse`
- `/etc/sysconfig/sendmail`
- `/etc/sysconfig/apmd`
- `/etc/sysconfig/init`
- `/etc/sysconfig/keyboard`
- `/etc/sysconfig/network`
- `/etc/sysconfig/pcmica`
- `/etc/sysconfig/soundcard` (which is written by sndconfig)

Let's take a look at each one.

`/etc/sysconfig/clock`

The `/etc/sysconfig/clock` file controls the interpretation of values read from the system clock. Earlier releases of Red Hat Linux used the following values (which are deprecated):

- `CLOCKMODE=`*`mode`*, where *mode* is one of the following:
  - `GMT` -- indicates that the clock is set to UTC.
  - `ARC` -- on Alpha only indicates the ARC console's 42-year time offset is in effect.

Currently, the correct values are:

- `UTC=`*`boolean`*, where *boolean* is the following:
  - `true` -- indicates that the clock is set to UTC. Any other value indicates that it is set to local time.
- `ARC=`*`boolean`*, where *boolean* is the following:
  - `true` -- (for Alpha-based systems only) Indicates the ARC console's 42-year time offset is in effect. Any other value indicates that the normal UNIX epoch is assumed.

- `ZONE="`*`filename`*`"` -- indicates the zonefile under `/user/share/zone-info` that `/etc/localtime` is a copy of, for example:

```
ZONE="US/Eastern"
```

`/etc/sysconfig/hwconf`

The `/etc/sysconfig/hwconf` file lists all the hardware that `kudzu` detected on your system, as well as the drivers used, vendor ID and device ID information. It is not meant to be edited. If you do edit it, devices could suddenly show up as being *added* or *removed*.

`/etc/sysconfig/mouse`

The `/etc/sysconfig/mouse` file is used to specify information about the available mouse. The following values may be used:

- `MOUSETYPE=`*`type`*, where `type` is one of the following:
  - `microsoft` -- A Microsoft mouse.
  - `mouseman` -- A MouseMan mouse.
  - `mousesystems` -- A Mouse Systems mouse.
  - `ps/2` -- A PS/2 mouse.
  - `msbm` -- A Microsoft bus mouse.

- `logibm` -- A Logitech bus mouse.
- `atibm` -- An ATI bus mouse.
- `logitech` -- A Logitech mouse.
- `mmseries` -- An older MouseMan mouse.
- `mmhittab` -- An mmhittab mouse.

- `XEMU3=`*`emulation`*, where *emulation* is one of the following:
  - `yes` -- Three mouse buttons should be emulated.
  - `no` -- The mouse already has three buttons.

In addition, `/dev/mouse` is a symlink that points to the actual mouse device.

`/etc/sysconfig/sendmail`

The `/etc/sysconfig/sendmail` allows messages to be sent to one or more recipients, routing the message over whatever networks are necessary. The file sets the default values for the sendmail program to run. Its default values are to run as a background daemon, and to check its queue once an hour in case something has backed up.

The following values may be used:

- `DAEMON=`*`answer`*, where *answer* is one of the following:
  - `yes` -- Sendmail should be configured. `yes` implies `-bd`.
  - `no` -- Sendmail should not be configured.

- `QUEUE=1h` which is given to sendmail as `-q$QUEUE`. The `-q` option is not given to sendmail if `/etc/sysconfig/sendmail` exists and QUEUE is empty or undefined.

/etc/sysconfig/apmd

The `/etc/sysconfig/apmd` is used by apmd, as a configuration for what things to start/stop/change on suspend or resume. It is set up to turn on or off apmd during startup, depending on whether your hardware supports Advanced Power Management (apm), or if you choose not to use it.

/etc/sysconfig/init

The `/etc/sysconfig/init` file controls how the system will look during bootup.

The following values may be used:

- `BOOTUP=<some bootup mode>`, where <*some bootup mode*> is one of the following:
  - `BOOTUP=color` means new (as of Red Hat Linux 6.0) boot display.
  - `BOOTUP=verbose` means old style display.
  - Anything else means new display, but without ANSI-formatting.

- `LOGLEVEL=<a number>`, where <*a number*> sets the initial console logging level for the kernel. The default is 7; 8 means everything (including debugging); 1 means nothing except kernel panics. syslogd will override this once it starts.
- `RES_COL=<a number>`, where <*a number*> is a column of the screen to start status labels at. Defaults to 60.
- `MOVE_TO_COL=<a command>`, where <*a command*> moves the cursor to $RES_COL. Defaults to ANSI sequences output by echo `-e`.
- `SETCOLOR_SUCCESS=<a command>`, where <*a command*> sets the color to a color indicating success. Defaults to ANSI sequences output by echo `-e`, setting the color to green.
- `SETCOLOR_FAILURE=<a command>`, where <*a command*> sets the color to a color indicating failure. Defaults to ANSI sequences output by echo `-e`, setting the color to red.

- `SETCOLOR_WARNING=<a command>`, where <*a command*> sets the color to a color indicating warning. Defaults to ANSI sequences output by echo `-e`, setting the color to yellow.
- `SETCOLOR_NORMAL=<a command>`, where <*a command*> sets the color to 'normal'. Defaults to ANSI sequences output by echo `-e`.
- `MAGIC_SYSRQ=an answer`, where *an answer* is one of the following:
  - `yes` -- Enables the magic sysrq key.
  - `no` -- Disables the magic sysrq key, and [Stop]-[A] (break on serial console) on SPARC.

- `PROMPT=an answer`, where *an answer* is one of the following:
  - `yes` -- Enables the key check for interactive mode.
  - `no` -- Disables the key check for interactive mode.

/etc/sysconfig/keyboard

The `/etc/sysconfig/keyboard` file controls the behavior of the keyboard. The following values may be used:

- `KEYTABLE=file`, where *file* is the name of a keytable file. For example: `KEYTABLE="/usr/lib/kbd/keytables/us.map"`
- `KEYBOARD=sun|pc`, which is used on SPARCs only. `sun` means a Sun keyboard is attached on `/dev/kbd`, `pc` means a PS/2 keyboard is on a PS/2 port.

/etc/sysconfig/network

The `/etc/sysconfig/network` file is used to specify information about the desired network configuration. The following values may be used:

- `NETWORKING=answer`, where *answer* is one of the following:
  - `yes` -- Networking should be configured.

– `no` -- Networking should not be configured.

- `HOSTNAME=`*`hostname`*, where *hostname* should be the FQDN (Fully Qualified Domain Name), but can be whatever hostname you want.

Please Note

For compatibility with older software that people might install (such as `trn`), the `/etc/HOSTNAME` file should contain the same value as here.

- `FORWARD_IPV4=`*`answer`*, where *answer* is one of the following:
  – `yes` -- Perform IP forwarding.
  – `no` -- Do not perform IP forwarding.

  (The current Red Hat Linux installation sets this to "no" by default [for RFC compliance], but if `FORWARD_IPV4` is not set at all, forwarding is *enabled* for compatibility with the configuration files used on Red Hat Linux versions 4.2 and earlier.)
- `GATEWAY=`*`gw-ip`*, where *gw-ip* is the IP address of the network's gateway.
- `GATEWAYDEV=`*`gw-dev`*, where *gw-dev* is the gateway device (e.g. eth0).
- `NISDOMAIN=`*`dom-name`*, where *dom-name* is the NIS domain name.

`/etc/sysconfig/pcmcia`

The `/etc/sysconfig/pcmcia` file is used to specify PCMCIA configuration information. The following values may be used:

- `PCMCIA=`*`answer`*, where *answer* is one of the following:
  – `yes` -- PCMCIA support should be enabled.

– `no` -- PCMCIA support should not be enabled.

- `PCIC=`*`pcic-type`*, where *pcic-type* is one of the following:
  – `i82365` -- The computer has an i82365-style PCMCIA socket chipset.
  – `tcic` -- The computer has a tcic-style PCMCIA socket chipset.

- `PCIC_OPTS=`*`option`*, where *option* is the socket driver (i82365 or tcic) timing parameters.
- `CORE_OPTS=`*`option`*, where *option* is the list of `pcmcia_core` options.
- `CARDMGR_OPTS=`*`option`*, where *option* is the list of options for the PCMCIA cardmgr (such as `-q`, quiet mode; `-m`, looks for loadable kernel modules in the specified director; and so on, read the cardmgr man page for more information).

`/etc/sysconfig/soundcard`

The `/etc/sysconfig/soundcard` file is generated by sndconfig and should not be modified. It is used by `/etc/rc.d/init.d/sound` to set up your system properly. The sole use of this is to determine what card entry in the menu to pop up by default the next time sndconfig is run.

It may contain the following:

- `CARDTYPE=`*`<a card>`*, where *<a card>* is seen as, for example, `CARDTYPE=SB16`.

Files in `/etc/sysconfig/network-scripts/`

The following files are normally found in `/etc/sysconfig/network-scripts`:

- `/etc/sysconfig/network-scripts/ifup`
- `/etc/sysconfig/network-scripts/ifdown`

- /etc/sysconfig/network-scripts/network-functions
- /etc/sysconfig/network-scripts/ifcfg-*<interface-name>*
- /etc/sysconfig/network-scripts/ifcfg-*<interface-name>-<clone-name>*
- /etc/sysconfig/network-scripts/chat-*<interface-name>*
- /etc/sysconfig/network-scripts/dip-*<interface-name>*
- /etc/sysconfig/network-scripts/ifup-post
- /etc/sysconfig/network-scripts/ifdhcpc-done

Let's take a look at each one.

/etc/sysconfig/network-scripts/ifup, /etc/sysconfig/network-scripts/ifdown

These are symbolic links to /sbin/ifup and /sbin/ifdown, respectively. These are the only two scripts in this directory that should be called directly; these two scripts call all the other scripts as needed. These symlinks are here for legacy purposes only -- they will probably be removed in future versions, so only /sbin/ifup and /sbin/ifdown should currently be used.

These scripts take one argument normally: the name of the device (e.g. "eth0"). They are called with a second argument of "boot" during the boot sequence so that devices that are not meant to be brought up on boot (ONBOOT=no, [see below]) can be ignored at that time.

/etc/sysconfig/network-scripts/network-functions

Not really a public file. Contains functions which the scripts use for bringing interfaces up and down. In particular, it contains most of the code for handling alternative interface configurations and interface change notification through netreport.

/etc/sysconfig/network-scripts/ifcfg-*<interface-name>*, /etc/sysconfig/network-scripts/ifcfg-*<interface-name>-<clone-name>*

The first file defines an interface, while the second file contains only the parts of the definition that are different in a "clone" (or alternative) interface. For example,

the network numbers might be different, but everything else might be the same, so only the network numbers would be in the clone file, while all the device information would be in the base `ifcfg` file.

The items that can be defined in an `ifcfg` file depend on the interface type.

The following values are common to all base files:

- `DEVICE=`*`name`*, where *name* is the name of the physical device (except dynamically-allocated PPP devices where it is the "logical name").
- `IPADDR=`*`addr`*, where *addr* is the IP address.
- `NETMASK=`*`mask`*, where *mask* is the netmask value.
- `NETWORK=`*`addr`*, where *addr* is the network address.
- `BROADCAST=`*`addr`*, where *addr* is the broadcast address.
- `GATEWAY=`*`addr`*, where *addr* is the gateway address.
- `ONBOOT=`*`answer`*, where *answer* is one of the following:
  - `yes` -- This device should be activated at boot-time.
  - `no` -- This device should not be activated at boot-time.
- `USERCTL=`*`answer`*, where *answer* is one of the following:
  - `yes` -- Non-root users are allowed to control this device.
  - `no` -- Non-root users are not allowed to control this device.

- `BOOTPROTO=`*`proto`*, where *proto* is one of the following:
  - `none` -- No boot-time protocol should be used.
  - `bootp` -- The BOOTP protocol should be used.
  - `dhcp` -- The DHCP protocol should be used.

The following values are common to all PPP and SLIP files:

- `PERSIST=`*`answer`*, where *answer* is one of the following:
  - `yes` -- This device should be kept active at all times, even if deactivated after a modem hang up.
  - `no` -- This device should not be kept active at all times.

- `MODEMPORT=`*`port`*, where *port* is the modem port's device name (for example, `"/dev/modem"`).
- `LINESPEED=`*`baud`*, where *baud* is the modem's linespeed (for example, `"115200"`).
- `DEFABORT=`*`answer`*, where *answer* is one of the following:
  - `yes` -- Insert default abort strings when creating/editing the script for this interface.
  - `no` -- Do not insert default abort strings when creating/editing the script for this interface.

The following values are common to all PPP files:

- `DEFROUTE=`*`answer`*, where *answer* is one of the following:
  - `yes` -- Set this interface as the default route.
  - `no` -- Do not set this interface as the default route.

- `ESCAPECHARS=`*`answer`*, where *answer* is one of the following:
  - `yes` -- Use the pre-defined asyncmap.
  - `no` -- Do not use the pre-defined asyncmap.

(This represents a simplified interface; it doesn't let people specify which characters to escape. However, almost everyone can use an asyncmap of `00000000` anyway, and it's possible to set `PPPOPTIONS` to use an arbitrary asyncmap if so desired.)

- `HARDFLOWCTL=`*`answer`*, where *answer* is one of the following:
  - `yes` -- Use hardware flow control.
  - `no` -- Do not use hardware flow control.

- `PPPOPTIONS=`*`options`*, where *options* is an arbitrary option string. It is placed last on the command line so it can override other options (such as asyncmap) that were specified previously.
- `PAPNAME=`*`name`*, where *name* is used as part of `"name $PAPNAME"` on the `pppd` command line.

  Note that the "`remotename`" option is always specified as the logical PPP device name, like "`ppp0`" (which might perhaps be the physical device ppp1 if some other PPP device was brought up earlier...), which makes it easy to manage PAP/CHAP files -- name/password pairs are associated with the logical PPP device name so that they can be managed together.

  In principle, there shouldn't anything that would keep the logical PPP device names from being "`worldnet`" or "`myISP`" instead of "`ppp0`" -- "`pppN`."
- `REMIP=`*`addr`*, where *addr* is the remote IP address (which is normally unspecified).
- `MTU=`*`value`*, where *value* is the value to be used as MTU.
- `MRU=`*`value`*, where *value* is the value to be used as MRU.
- `DISCONNECTTIMEOUT=`*`value`*, where *value* represents the number of seconds to wait before re-establishing the connection after a successfully-connected session terminated.

- `RETRYTIMEOUT=value`, where *value* represents the number of seconds to wait before re-attempting to establish a connection after a previous attempt has failed.

`/etc/sysconfig/network-scripts/chat-<interface-name>`

This file is a chat script for PPP or SLIP connections, and is intended to establish the connection. For SLIP devices, a DIP script is written from the chat script; for PPP devices, the chat script is used directly.

`/etc/sysconfig/network-scripts/dip-<interface-name>`

This write-only script is created from the chat script by `netcfg`. Do not modify this file. In the future, this file may disappear and instead will be created on-the-fly from the chat script.

`/etc/sysconfig/network-scripts/ifup-post`

This file is called when any network device (except a SLIP device) comes up. Calls `/etc/sysconfig/network-scripts/ifup-routes` to bring up static routes that depend on that device. Brings up aliases for that device. Sets the hostname if it is not already set and a hostname can be found for the IP for that device. Sends SIGIO to any programs that have requested notification of network events.

Could be extended to fix up name service configuration, call arbitrary scripts, and more, as needed.

`/etc/sysconfig/network-scripts/ifdhcpc-done`

This file is called by `dhcpcd` once DHCP configuration is complete; sets up `/etc/resolv.conf` from the version dhcpcd dropped in `/etc/dhcpc/resolv.conf`.

2.13.3 System V Init

This section is a brief description of the internals of the boot process. It discusses how the machine boots using SysV init, as well as the differences between the init used in older Linux releases, and SysV init.

The Init program is run by the kernel at boot time. It is in charge of starting all the normal processes that need to run at boot time. These include the getty processes that allow you to log in, NFS daemons, FTP daemons, and anything else you want to run when your machine boots.

SysV init is quickly becoming the standard in the Linux world to control the startup of software at boot time, because it is easier to use and more powerful and flexible than the traditional BSD init.

SysV init also differs from BSD init in that the configuration files are in a subdirectory of `/etc` instead of residing directly in `/etc`. In `/etc/rc.d`, you will find `rc.sysinit` and the following directories:

```
init.d
rc0.d
rc1.d
rc2.d
rc3.d
rc4.d
rc5.d
rc6.d
```

The `init.d` directory contains a variety of scripts. Basically, you must have one script for each service you may need to start at boot time or when entering another runlevel. Services include things like networking, `nfs`, `sendmail`, `httpd`, and so on. Services do not include things like `setserial` that must only be run once and then exited. Things like that should go in `rc.local` or `rc.serial`.

If you want `rc.local`, it should be in `/etc/rc.d`. Most systems include one even though it doesn't do much. You can also include an `rc.serial` file in `/etc/rc.d` if you need to perform serial port specific tasks at boot time.

The chain of events is as follows:

- The kernel looks in several places for init and runs the first one it finds
- init runs `/etc/rc.d/rc.sysinit`
- `rc.sysinit` handles most of the boot loader's processes and then runs `rc.serial` (if it exists)

- init runs all the scripts for the default runlevel.
- init runs `rc.local`

The default runlevel is decided in `/etc/inittab`. You should have a line close to the top like:

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

From this, you'd look in the second column and see that the default runlevel is 3. If you want to change it, you can edit `/etc/inittab` by hand. Be very careful when you are editing the `inittab` file. If you do mess up, you can fix it by rebooting and typing:

```
LILO boot:  linux single
```

This *should* allow you to boot into single user mode so you can re-edit `inittab` to its previous value.

Now, how does it run all the right scripts? If you enter `ls -l` on `rc3.d`, you might see something like:

```
lrwxrwxrwx 1 root root 17 3:11 S10network -> ../init.d/network
lrwxrwxrwx 1 root root 16 3:11 S30syslog -> ../init.d/syslog
lrwxrwxrwx 1 root root 14 3:32 S40cron -> ../init.d/cron
lrwxrwxrwx 1 root root 14 3:11 S50inet -> ../init.d/inet
lrwxrwxrwx 1 root root 13 3:11 S60nfs -> ../init.d/nfs
lrwxrwxrwx 1 root root 15 3:11 S70nfsfs -> ../init.d/nfsfs
lrwxrwxrwx 1 root root 18 3:11 S90lpd -> ../init.d/lpd.init
lrwxrwxrwx 1 root root 11 3:11 S99local -> ../rc.local
```

What you'll notice is that there are no "real" files in the directory. Everything there is a link to one of the scripts in the `init.d` directory. The links also have an "S" and a number at the beginning. The "S" means to start this particular script and a "K" would mean to stop it. The number in the file name is for ordering purposes. Init will start all the services based on the order in which they appear. You can duplicate numbers, but it will only confuse you somewhat. You only need to use a two digit number, along with an upper case "S" or "K" to start or stop the services you require.

How does init start and stop services? Simple. Each of the scripts is written to accept an argument which can be "`start`" and "`stop`". You can execute those scripts by hand, in fact, with a command like:

```
/etc/rc.d/init.d/httpd stop
```

This will stop the `httpd` server. init reads the name and if it has a "K," it calls the script with the "`stop`" argument. If it has an "S" it calls the script with a "`start`" argument.

Why all these runlevels? Some people want an easy way to set up machines to be multi-purpose. You could have a "server" runlevel that just runs `httpd`, `sendmail`, networking, etc. Then you could have a "user" runlevel that runs `gdm`, networking, and so on.

2.13.4 Init Runlevels

Generally, Red Hat Linux operates in runlevel 3 -- full multi-user mode. The following runlevels are defined in Red Hat Linux:

- `0` -- Halt
- `1` -- Single user mode
- `2` -- Multi-user mode, without NFS
- `3` -- Full multi-user mode
- `4` -- Not used
- `5` -- Full multi-user mode (with an X-based login screen)
- `6` -- Reboot

If your machine gets into a state where it will not boot due to a bad `/etc/inittab`, or will not let you log in because you have a corrupted `/etc/passwd` or have simply forgotten your password, boot into single user mode by typing `linux single` at the LILO boot prompt. A very bare system will boot and you will have a shell from which you can fix things.

2.13.5 Initscript Utilities

The `chkconfig` utility provides a simple command-line tool for maintaining the `/etc/rc.d` directory hierarchy. It relieves system administrators from having to directly manipulate the numerous symlinks in `/etc/rc.d`.

In addition, there is the `ntsysv` utility, that provides a screen-oriented interface, versus `chkconfig`'s command-line interface.

Please see the `chkconfig` and `ntsysv` man pages for more information.

2.13.6 Running Programs at Boot Time

The file `/etc/rc.d/rc.local` is executed at boot time, after all other initialization is complete, and whenever you change runlevels. You can add additional initialization commands here. For instance, you may want to start up additional daemons, or initialize a printer. In addition, if you require serial port setup, you can edit `/etc/rc.d/rc.serial`, and it will be executed automatically at boot time.

The default `/etc/rc.d/rc.local` simply creates a nice login banner with your kernel version and machine type.

2.13.7 Shutting Down

To shut down Red Hat Linux, issue the `shutdown` command. You can read the `shutdown` man page for complete details, but the two most common usages are:

```
shutdown -h now
shutdown -r now
```

Each will cleanly shutdown the system. After shutting everything down, the `-h` option will halt the machine, and the `-r` option will reboot.

Although the `reboot` and `halt` commands are now "smart" enough to invoke `shutdown` if run while the system is in runlevels 1-5, it is a bad habit to get into, as not all Linux-like operating systems have this feature.

2.14 Rescue Mode

When things go wrong, there are several ways to work on fixing them. However, they require that you understand the system well. We will present the ways that you can boot into rescue modes where you can use your own knowledge to rescue the system.

2.14.1 What is Rescue Mode?

Rescue mode is a term used to describe a method of booting a small Linux environment completely from a diskette, CD or other method.

What follows in this section may help you recover from a problem at some point.

As the name implies, rescue mode is there to rescue you from something. In normal operation, your Red Hat Linux system uses files located on your system's hard drive to do everything -- run programs, store your files, and more.

However, there may be times when you are unable to get Linux running completely enough to access its files on your system's hard drive. By using rescue mode, it's possible to access the files stored on your system's hard drive, even if you can't actually run Linux from that hard drive.

Normally, you'll need to get into rescue mode for one of two reasons:

- You are unable to boot Linux, and you'd like to fix it.
- You are having hardware or software problems, and you want to get a few important files off your system's hard drive.

Let's take a closer look at each of these scenarios.

Unable to boot Linux

Many times this is caused by the installation of another operating system after you've installed Red Hat Linux. Some other operating systems assume that you have no other operating systems on your computer, and overwrite the Master Boot Record (or MBR) that originally contained the LILO bootloader. If LILO is overwritten in this manner, you're out of luck -- unless you can get into rescue mode.

Hardware/software problems

There can be as many different situations under this category as there are systems running Linux. Things like failing hard drives and forgetting to run LILO after building a new kernel are just two instances that can keep you from booting Red Hat Linux. If you can get into rescue mode, you might be able to resolve the problem -- or at least get copies of your most important files.

To boot your system in rescue mode, enter the following parameter at the installation boot prompt:

```
boot: linux rescue
```

You can get to the installation boot prompt in one of these ways:

- By booting your system from the diskette or CD-ROM that came with your Red Hat Linux boxed set.
- By booting from a network or PCMCIA boot diskette. These methods assume your network connection is working and require you to identify the network host and transfer type. For an explanation of how to specify this information, see "Installing over the Network" in Chapter 10, *Installing Red Hat Linux via Text Mode*.

Once you have your system in rescue mode, a prompt appears:

```
bash#
```

From this prompt, you can run the commands listed below:

```
anaconda        gzip        mkfs.ext2   ps
badblocks       head        mknod       python
bash            hwclock     mkraid      python1.5
cat             ifconfig    mkswap      raidstart
chatter         init        mlabel      raidstop
chmod           insmod      mmd         rcp
chroot          less        mmount      rlogin
clock           ln          mmove       rm
collage         loader      modprobe    rmmod
cp              ls          mount       route
cpio            lsattr      mpartition  rpm
```

```
dd                lsmod       mrd         rsh
ddcprobe          mattrib     mread       sed
depmode           mbadblocks  mren        sh
df                mcd         mshowfat    sync
e2fsck            mcopy       mt          tac
fdisk             mdel        mtools      tail
fsck              mdeltree    mtype       tar
fsck.ext2         mdir        mv          touch
ftp               mdu         mzip        traceroute
genhdlist         mformat     open        umount
gnome-pty-helper  minfo       pico        uncpio
grep              mkdir       ping        uniq
gunzip            mke2fs      probe       zcat
```

However, if your root filesystem is undamaged, you can mount it and then run any standard Linux utility. For example, suppose your root filesystem is in `/dev/hda5`. Here's how to mount this partition:

```
mount -t ext2 /dev/hda5 /foo
```

Where */foo* is a directory that you have created.

Now you can run `chroot`, `fsck`, `man`, and other utilities. At this point, you are running Linux in single-user mode.

If you don't know the names of your Linux partitions, you can guess: mounting non-existent partitions will do no harm.

Booting Single-User Mode Directly

You may be able to boot single-user mode directly. If your system boots, but does not allow you to log in when it has completed booting, try rebooting and specifying one of these options at the LILO boot prompt:

```
LILO boot: linux single
LILO boot: linux emergency
```

In single-user mode, you computer boots to runlevel 1. Your local filesystems will be mounted but your network will not be activated. You get a usable system maintenance shell.

In emergency mode, you are booted into the most minimal environment possible. The root filesystem will be mounted read-only and almost nothing will be set up. The main advantage of this over `linux single` is that your init files are not loaded. If init is corrupted or not working, you can still mount filesystems to recover data that could be lost during a re-installation.

A Handy Trick

Have you ever rebuilt a kernel and, eager to try out your new handiwork, rebooted before running LILO? And you didn't have an entry for an older kernel in `lilo.conf`? Read on…

In many cases, it's possible to boot your Red Hat Linux/Intel system from the Red Hat Linux boot disk with your root filesystem mounted and ready to go. Here's how:

Enter the following command at the boot disk's `boot:` prompt:

```
linux single root=/dev/hdXX initrd=
```

(Replace the *XX* in `/dev/hd`*XX* with the appropriate letter and number for your root partition.)

What does this do? First, it starts the boot in single-user mode, with the root partition set to your root partition. The empty `initrd` specification bypasses the installation-related image on the boot disk, which will cause you to enter single-user mode immediately.

Is there a downside? Unfortunately, yes. Because the kernel on the Red Hat Linux boot disk only has support for IDE built-in, those of you with SCSI-based systems won't be able to use this trick. In that case, you'll have to use the boot/rescue disk combination mentioned above.

3 System Configuration

One of the main strengths of Red Hat Linux is that the operating system can be configured to do just about anything. In the past, this meant editing sometimes cryptic configuration files by hand, then restarting system services and crossing your fingers in the hope that your changes were correctly made. Additionally, the variety of configuration options can be bewildering to new users, who may not know where to look for a particular configuration file.

Red Hat Linux provides two system configuration utilities: linuxconf and the control panel. The control panel provides a launcher for various Red Hat system configuration tools, including linuxconf. Instructions on how to use the control panel and the tools it includes can be found in Section 3.2, *System Configuration with the Control Panel*.

The linuxconf configuration tool can ease some of the burdens of system configuration. Linuxconf is both:

- A configuration interface -- You type the values needed to configure your system into a user interface.
- A configuration activator -- When you're satisfied with your edits, you tell linuxconf to apply the changes you've made.

While linuxconf can do nearly everything the control panel tools can, there are two areas in which the control panel still holds the upper hand:

- Printer configuration
- Loading kernel modules to support new hardware

Let's take a look at linuxconf.

3.1 System Configuration with linuxconf

Linuxconf allows you to configure and control various aspects of your system, and is capable of handling a wide range of programs and tasks. Complete documentation of linuxconf could be a separate book in its own right and is certainly more than we can

cover in this chapter. Instead, we'll focus on common tasks such as adding new users and getting connected to a network.

More information on linuxconf, including its most recent release can be found at the linuxconf website:

http://www.solucorp.qc.ca/linuxconf/

The linuxconf website includes fairly extensive information, including a description, rationale, history, list of contacts, and a lot of other information in addition to the software. The website is maintained by linuxconf's creator and developer, Jacques Gelinas, so it includes the latest news about linuxconf.

This chapter will go into detail on just a few of linuxconf's capabilities. If you just need a quick reference to show you where to go in linuxconf for the tasks covered by this chapter, see Section 3.1.15, *Finding Your Way Through* linuxconf.

If you need more help with linuxconf, please try these sources:

- The linuxconf FAQ, which is available at http://www.xc.org/jonathan/linuxconf-faq.html.
- The archives of the linuxconf mailing list, which are available at http://hub.xc.org/scripts/lyris.pl?visit=linuxconf.
- After you've checked the linuxconf FAQ and the archives of the linuxconf list, you might try posting your question to the linuxconf list. Subscription information for the linuxconf list is available at the linuxconf website (http://www.solucorp.qc.ca/linuxconf/); click on the "Mailing lists" link.

  Please note that this list is for questions pertaining to linuxconf, and is not intended for general Linux questions.

3.1.1 Running linuxconf

You'll need to be root to run linuxconf, so if you're in your user account, `su` to become root.

Now, type `linuxconf` at the shell prompt to begin the program.

3.1.2 Linuxconf User Interfaces

Linuxconf has four user interfaces:

- Text-based -- Using the same user interface style as the Red Hat Linux text-mode installation program, the text-based interface makes it easy to navigate your way through linuxconf if you aren't running X. If you are running X, you can switch to a virtual console, log in as root, and type `linuxconf` to bring up text-mode linuxconf.

  Use the [Tab] and [arrow] keys to navigate the text-mode screens. A **down arrow** on a line indicates that a pull-down menu exists on that line. The [Ctrl]-[X] key combination will make pull-down menus appear.

- Graphical user interface (GUI) -- linuxconf can take advantage of X, and give you an easy-to-use "point and click" tree menu interface (see Section 3.1.3, *Tree Menu Interface* for more information). Red Hat Linux includes a GUI interface for linuxconf called gnome-linuxconf.

  This document will display linuxconf screens using the gnome-linuxconf interface, but you shouldn't have any trouble using the other interfaces with the instructions provided here.

- Web-based -- A Web-based interface makes remote system administration a breeze; it can also be displayed with the Lynx text-mode browser.

  To use the linuxconf Web interface, use your browser to connect to port 98 on the machine running linuxconf (i.e., http://your_machine:98).

  Before you use the Web-based interface, you'll need to configure linuxconf to allow connections from the machine running the browser. See Section 3.1.4, *Enabling Web-Based* linuxconf *Access* for instructions on enabling Web access to linuxconf.

- Command line -- linuxconf's command-line mode is handy for manipulating your system's configuration in scripts.

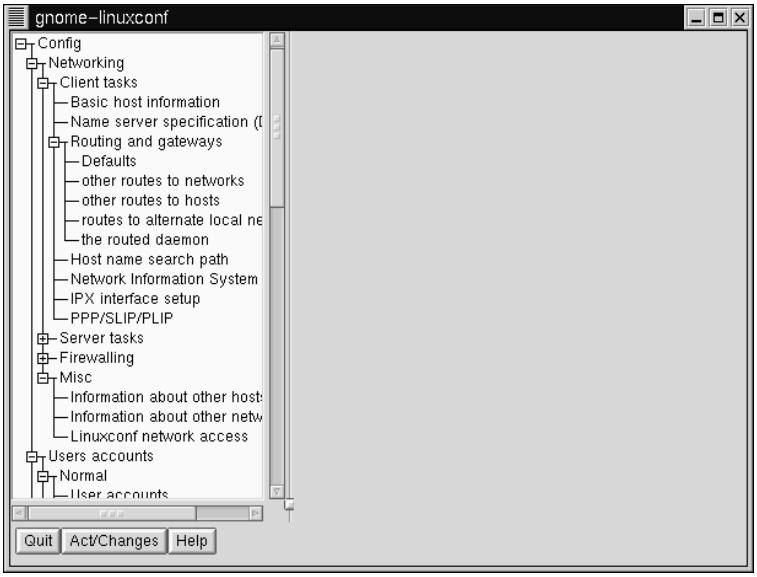
Linuxconf will start in either character-cell or X mode, depending on your **DISPLAY** environment variable. The first time you run linuxconf, an introductory message will be shown; although it is only displayed once, accessing help from the main screen will give you the same basic information.

Linuxconf includes some context-specific help. For information on any specific aspect of linuxconf, select **Help** from the screen you'd like help with. Note that not all help screens are complete at this time; as help screens are updated, they will be included in subsequent versions of linuxconf.

3.1.3 Tree Menu Interface

Linuxconf provides a tree menu of its hierarchical structure.

Figure 3–1 Linuxconf **Tree Menu View**



Finding the appropriate panel should be simple and fast. Collapse and expand sections by clicking on the + or - next to the menu item.

Selected entries will appear as tabs in the right-hand panel and will remain there until closed. If you end up with more tabs open than you like, just select **Cancel** on the bottom of each tab to close it without making any changes, or **Accept** to implement them.

Please Note

If you've grown fond of the older X Window System interface, it's still available. To return to it:

1. Open **Control** => **Control files and systems** => **Configure linuxconf modules**
2. De-select the treemenu check box
3. Click **Accept**
4. Click **Quit**
5. Restart linuxconf

3.1.4 Enabling Web-Based linuxconf Access

For security reasons, Web-based access to linuxconf is disabled by default. Before attempting to access linuxconf with a Web browser, you'll need to enable access. Here's how to do it:

1. Open **Config** => **Networking** => **Misc** => **Linuxconf network access**
2. In the **Linuxconf html access control** dialog box, enter the hostname of any computers that should be allowed access to Linuxconf. This includes your own system, if you wish to use the Web-based interface locally. Web accesses related to linuxconf may be logged to your system's `htmlaccess.log` file by selecting the check box.
3. Select the **Accept** button.

Web-based access should be enabled. To test it out, go to a system that you added to the access control list. Then, launch your Web browser, and enter the following URL:

```
http://<host>:98/
```

(Replace *<host>* with your system's hostname, of course.) You should see the main linuxconf page. Note that you will need to enter your system's root password to gain access beyond the first page.

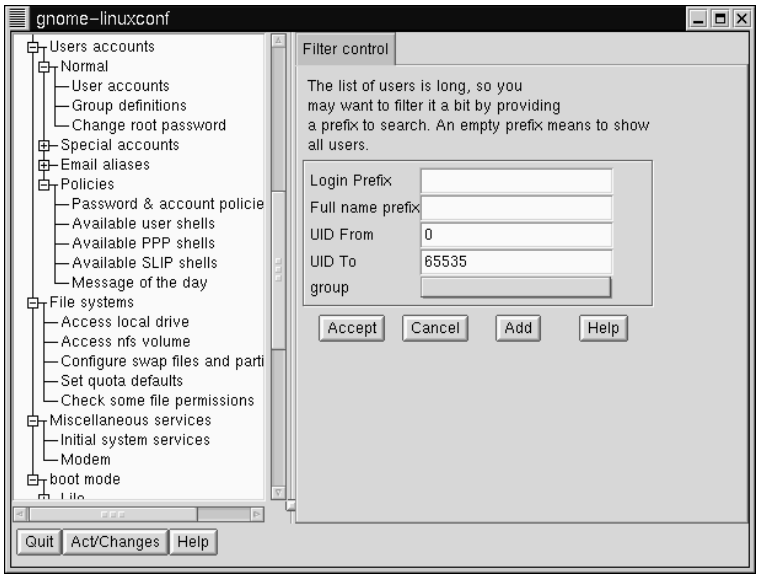
You can also enable network-wide access to linuxconf by following the same steps, by entering a network name instead of a hostname.

3.1.5 Adding a User Account

Adding a user is one of the most basic tasks you will encounter in administering your system. To add a user:

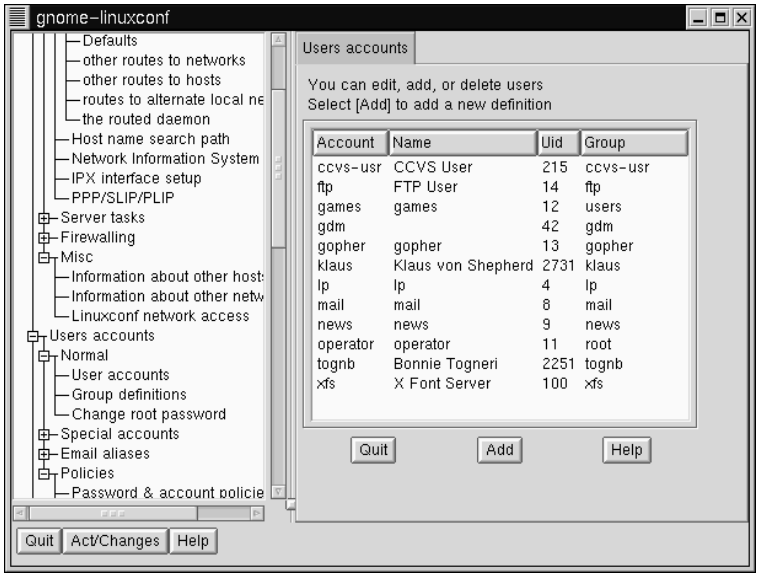
- Open **Config** => **Users accounts** => **Normal** => **User accounts.** Linuxconf may show you a filter screen (see Figure 3–2, *Filter Control Screen*).

Figure 3–2 Filter Control Screen



You can set the filter cutoff using **Control** => **Features.** The **Trigger for filter** field sets the number of entries that will pop up a filter screen. You can use the filter screen to select a smaller range of accounts than the full list. To get the full list, select **Accept** without changing any of the parameters. For detailed information on the various filters, select the **Help** button on the **Filter control** screen.Once you've applied or bypassed the filter, you'll see the **Users accounts** tab (see Figure 3–3, *Users Accounts Screen*).

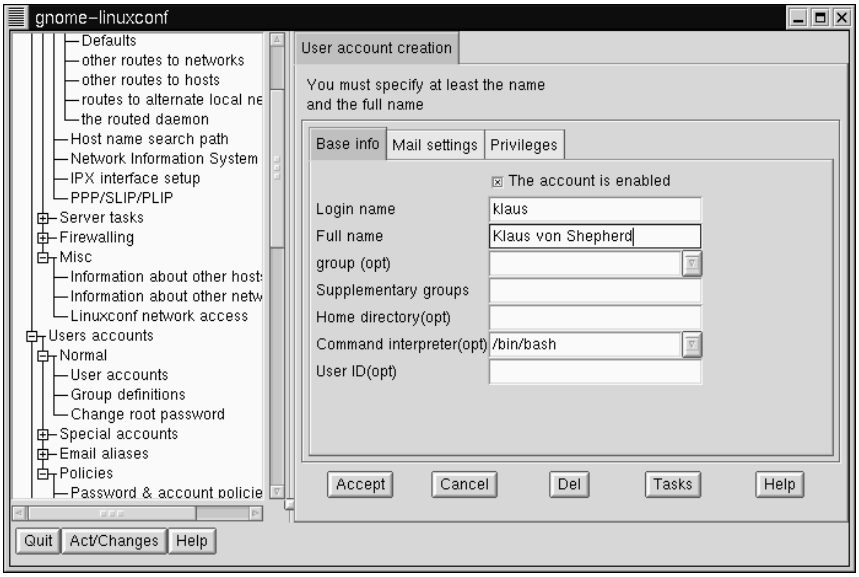
Figure 3–3 Users Accounts Screen



- Select **Add**. This will open the **User account creation** tab (see Figure 3–4, *User Account Creation*).

The **User account creation** screen includes the **Base info**, **Mail settings** and **Privileges** sections. Only the **Login name** is required, but you should be aware of the other fields, which you may or may not want to fill in.

Figure 3–4 User Account Creation



Base info for User Accounts

The **Login name** is the name of the account and is usually all lowercase letters. First or last names, initials or some combination thereof are fairly common login names. For a user named John T. Smith, **`smith`**, **`john`**, **`jts`**, or **`jsmith`** would be common user names. Of course **`spike`** or something else works just fine, too. You can also use numbers, so **`jts2`** would be fine if you had a second person with the same initials. There is no default for this field.

The **Full name** is the name of the user or the account. For an individual, it would be their name, **`John T. Smith`** for example. If the account represents a position rather than a person, the full name might be the title. So an account called **`webmaster`** might have a full name of **`Red Hat Webmaster`** or just **`Webmaster`**. There is no default for this field.

Since Red Hat Linux uses the User Private Group scheme, each user will be assigned to a default **group** consisting only of the user. For more information on User Private Groups, see Section 2.3.3, *User Private Groups*.

In the **Supplementary groups** field, you can specify additional groups. Group names should be separated by spaces. The default for this field is blank, meaning no supplementary groups are specified.

The **Home directory** specifies the home or login directory for the account. The default is `/home/`*`login`*, where *`login`* is replaced by the login name. A home directory is your starting point in the directory structure when you log in, or if in X, for each Xterm window opened. This is also where account specific preference files are stored.

The **Command interpreter** is the default shell for the account. The `bash` shell is the default shell for Red Hat Linux.

The **User ID** (UID) is the number associated with each user account. This is automatically generated by the system when the account is created, so just leave this field blank. The system uses the UID to identify an account.

Mail settings for User Accounts

The **Redirect messages to** field sets the system to forward e-mail messages sent to the user to this address instead.

An **E-mail alias** is a supplemental e-mail address that will be accepted by the system and forwarded on to the user. For example, you could add the alias **`jane_smith@yourdomain.com`** to user smitj's account. Then, e-mail sent to jane_smith@yourdomain.com would automatically be forwarded to smitj@yourdomain.com.

Privileges for User Accounts

In the **Privileges** section, you can grant access and/or control over various aspects of system configuration. As a default, regular users are denied all privileges on this screen. You may instead choose to grant or to silently grant them specific privileges. The difference between **Granted** and **Granted/silent** is that if the privilege is granted, linuxconf will ask for the user's password before allowing them the privilege. If the privilege is granted silently, linuxconf will not prompt for their password.

Generally, careful system administrators won't grant users any system configuration privileges unless it is absolutely necessary. If you do grant privileges, be careful when granting them silently. If a user with silently granted privileges logs in to his/her

machine and walks away, their privileges are wide open for the next person who sits down at their desk. Silently granted privileges are less risky if used on machines in a physically restricted area.

May use linuxconf: the user is allowed to access all of linuxconf's capabilities, and they can set up or change **linuxconf** parameters. Note that use of **linuxconf** is separate from the privilege of activating configuration changes. System administrators might want to grant the use of linuxconf, but deny the activation privilege, so that the sysadmin has a final "yes/no" on whether to activate any configuration changes.

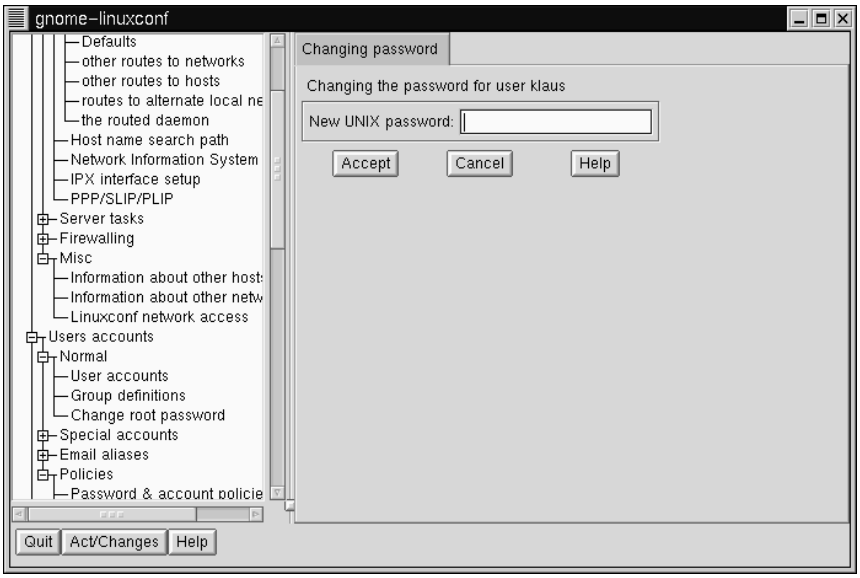
May activate config changes: After you change a parameter in linuxconf, at some point you'll have to indicate to linuxconf that the changes you made should be applied. Depending upon the flavor of linuxconf that you're using, you might do this by clicking on an **Activate the changes** button in GUI linuxconf, or clicking on an **Accept** button in Web-based linuxconf, or selecting an **Accept** button in text-mode linuxconf, etc.

You can grant the privilege of activating changes to a user. In that case, the user will be able to activate any changed system configuration parameters in linuxconf.

May shutdown: A user can be granted the right to shutdown the system. Note that Red Hat Linux is set in `/etc/inittab` to cleanly shutdown following the [Ctrl]-[Alt]-[Del] keystroke combination.

Once you have entered the login name and any other desired information select the **Accept** button at the bottom of the screen. If you decide against creating a new user, select **Cancel** instead.

Figure 3–5 Change Password Screen



When you click on **Accept** linuxconf will prompt you to enter the password, as in Figure 3–5, *Change Password Screen*. You'll have to re-type the password, to prevent unusable passwords caused by typos. Passwords must be at least six characters in length, but you can increase the required length and set other parameters for users' passwords at the **Users Accounts** => **Password & Account Policies** screen.

Good passwords contain a combination of letters, numbers, and special characters. It should use both upper case and lower case letters. Don't use your username, your anniversary, your social security number, your dog's name, your middle name or the word root. Don't use any variation of a word associated with your account or with yourself. Don't use a word that can be found in a dictionary; dictionary words are easy to crack.

A simple technique for creating a password is to use the first letters from each word of a phrase that is familiar to you (a line from a favorite song might be appropriate). Make a few letters uppercase, and insert a few numbers and/or special characters in place of letters and you'll have a decent password.

Press the **Accept** button again when finished. The system will let you know if it thinks the password is easy to crack; if you get a warning message, don't use the password.

3.1.6 Modifying a User Account

- Go to **Config** => **Users accounts** => **Normal** => **User accounts**, use the filter if necessary, and then select the account that you wish to modify.
- See Section 3.1.5, *Adding a User Account* if you need guidance for how to fill in the user accounts fields.

To implement the changes select **Accept**. If you decide against making any changes select **Cancel**. This guarantees that no changes are made.

3.1.7 Changing a User's Password

- Open **Config** => **Users accounts** => **Normal** => **User accounts**. This will open the **Users accounts** tab (see Figure 3–3, *Users Accounts Screen*).
- You may see a filter screen, depending upon the settings you've provided on **Control** => **Features**. If you want the full list, select `Accept` without changing any of the parameters. For detailed information on the various filters, select the `Help` button on the **Filter control** screen.
- Select the account whose password you wish to change. This will open the **User information** tab.
- Select **Passwd** from the options at the bottom of the screen.

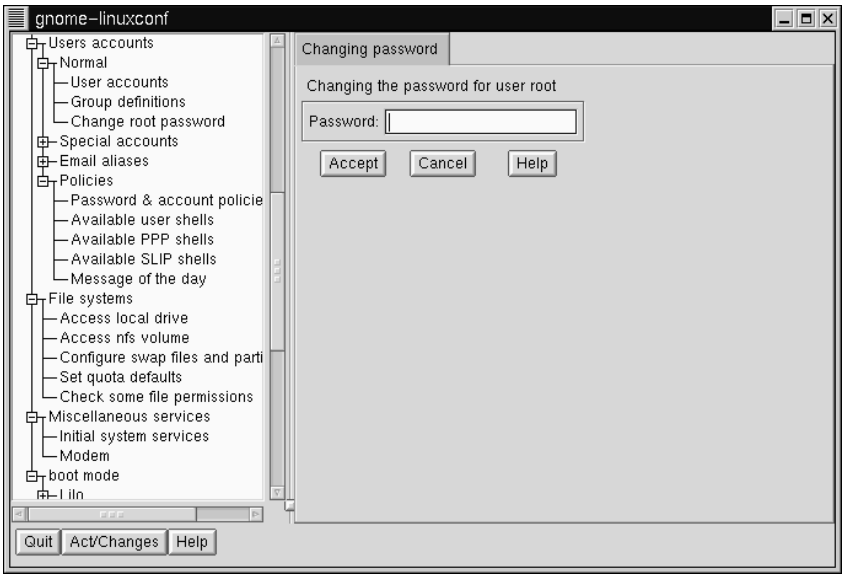
Linuxconf will prompt you to enter the new password. There is also a field called `Confirmation` where you will need to type the password again. This is to prevent you from mistyping the password. See Section 3.1.5, *Adding a User Account* for guidance on choosing a password. If you decide against changing the password, select `Cancel`. Once you have entered the new password select `Accept`.

3.1.8 Changing the Root Password

Because of the security implications of root access, linuxconf requires you to verify that you currently have access to the root account.

- Open **Config** => **Users accounts** => **Normal** => **Change root password**.

Figure 3–6 Root Password Verification Screen



You'll first need to enter the current root password to verify access to the root account.

Once you have entered root's current password, it will prompt you for a new password. In the **Confirmation** field, type the password again (see Figure 3–5, *Change Password Screen*). This is to prevent you from mistyping the password. See Section 3.1.5, *Adding a User Account* if you need guidance on choosing a password. Be sure to choose a good password! If you decide against changing the root password, just select **Cancel**. Once you have entered the new password select **Accept**.

3.1.9 Disabling a User Account

Disabling a user's account is preferable to deleting a user's account, unless you need the storage space or you're certain that his/her data will not be needed in the future. If a user's account is disabled, they will not be allowed to log in.

- Open **Config** => **Users accounts** => **Normal** => **User accounts**.

- De-select the check-box that states that `The account is enabled`. Select the `Accept` button at the bottom of the window and you're all set.

The account is disabled and can be enabled later using a similar method.

3.1.10 Enabling a User Account

By default, all newly created user accounts are enabled. If you need to enable an account, you can use linuxconf to do it.

- Open **Config** => **Users accounts** => **Normal** => **User accounts**. Select the **The account is enabled** checkbox.

3.1.11 Deleting a User Account

Please Note

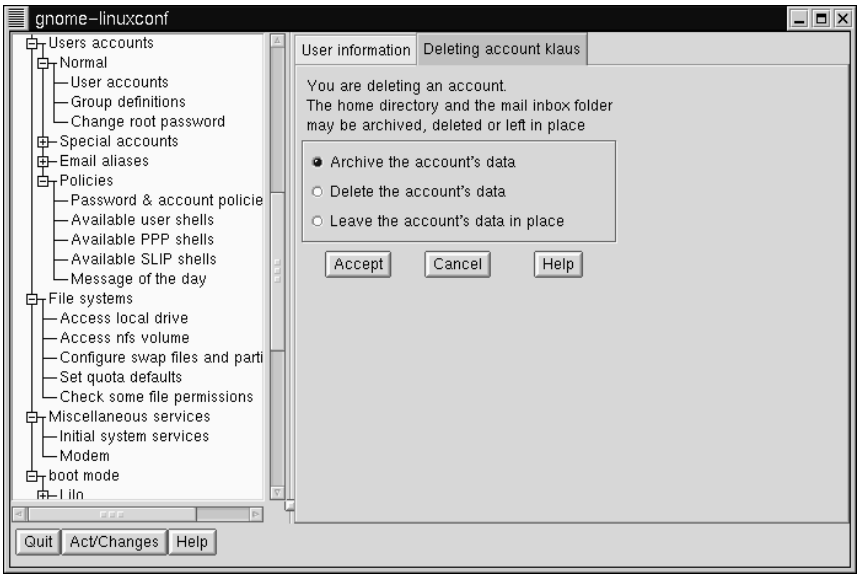
While there are options for retaining files associated with an account, any files that are deleted are gone and effectively unrecoverable. Take care when using this option!

To delete an account:

- Open **Config** => **Users accounts** => **User accounts**.
- On the **User accounts** screen (see Figure 3–3, *Users Accounts Screen*) select the account you wish to delete.
- At the bottom of the **User information** screen, select **Del** to delete the account.

Linuxconf will then prompt you with a list of options.

Figure 3–7 Deleting Account Screen



The default option is to archive the account's data. The archive option has the following effects:

1. Removes the user from the user accounts list;
2. Takes everything contained in the user's home directory and archives it (using tar and gzip compression), storing the resulting file in the `/default_home_directory/oldaccounts` directory. For an account named *`useraccount`* the filename would be similar to:

   `useraccount``-2000-01-10-497.tar.gz`

   The date indicates when the account was deleted, and the number following it is the ID of the process that actually performed the deletion. The `oldaccounts` directory is created in the same place as all of your user directories, and is created automatically the first time you remove a user account using this option.
3. Files not contained in the user's home directory, but owned by that user remain. The file is owned by the deleted account's user ID (UID). If you create a new

account and specifically assign it the UID of a deleted account, it will then become the owner of any remaining files.

Selecting **Delete the account's data** on the **Deleting account *<accountname>*** screen (see Figure 3–7, *Deleting Account Screen*) will:

1. Remove the user from the user accounts list;
2. Remove the user's home directory and all its contents.

Please Note

Files not contained in the user's home directory, but owned by that user will remain on the system. The file will still be owned by the deleted account's user ID (UID). If you create a new account and specifically assign it the UID of a deleted account, it will then become the owner of any such "orphaned" files.

Selecting **Leave the account's data in place** on the **Deleting account *<accountname>*** screen (see Figure 3–7, *Deleting Account Screen*) will:

1. Remove the user from the user accounts list;
2. Leave the user's home directory (with all its files) in place.

Please Note

Files and directories owned by the deleted account's user ID (UID) will remain on the system. If you create a new account and specifically assign it the UID of a deleted account, it will then become the owner of these "orphaned" files.

3.1.12 Groups

All users belong to one or more groups. Just as each file has a specific owner, each file belongs to a particular group as well. The group might be specific to the owner of the file, or may be a group shared by all users. The ability to read, write or execute a file can be assigned to a group; this is separate from the owner's rights. For example, the owner of a file will be able to write to a document, while other group members may only be able to read it.

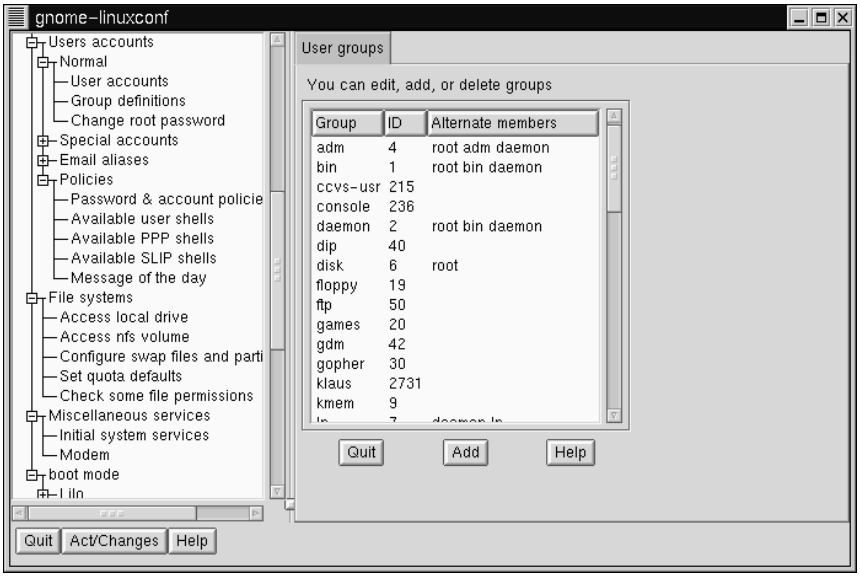
Creating a Group

To create a new group:

- Open **Config** => **Users accounts** => **Normal** => **Group definition.**

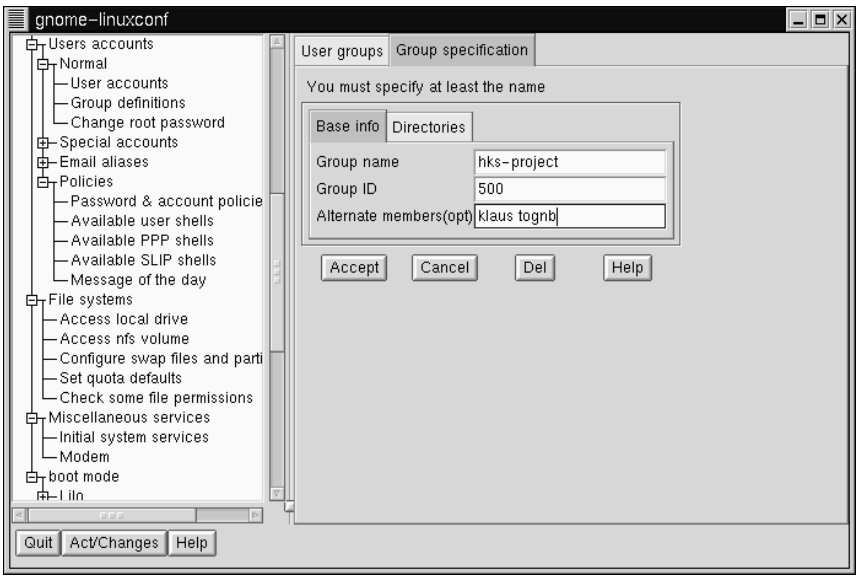
You may see a filter screen, depending upon the settings in **Control** => **Filters.** Either provide a filter, or select **Accept** to bypass the filter.

Figure 3–8 User Groups Screen



Select Add at the bottom of the **User groups** screen.

Figure 3–9 Group Specification Screen



Enter a group name. You may also wish to specify members of the group and can do so in the **Alternate members** field. The list of users should be space delimited, meaning that each username must have a space between it and the next one. Leave the **Group name** field blank, so that the system will assign a **Group ID** (GID) to your new group. When you're finished, select `Accept` and the group will be created.

Deleting a Group

To delete a group:

- Open **Config** => **Users accounts** => **Normal** => **Group definitions**.

  You may see a filter screen, depending upon the filter setting in **Control** => **Features**. You can use the filter to narrow your choice of groups by specifying a prefix.

- With or without a prefix select `Accept` at the bottom of the screen.
- On the **User groups** screen (see Figure 3–8, *User Groups Screen*) select the group you wish to delete.

- You'll be presented with the **Group specification** screen (see Figure 3–9, *Group Specification Screen*).
- Select `Del` to delete the group. Linuxconf will then prompt you to confirm the deletion. Choose **yes** to delete the group.

The group's files will still remain and their respective owners will still have sole control over them. The group name will be replaced with the deleted group's ID. The files may be assigned to a new group by using the `chgrp` command. More information on `chgrp` can be found by typing the command `info chgrp` or `man chgrp` at the shell prompt. If a new group is created and the deleted group's ID is specified then the new group will have access to the deleted group's files. Don't worry, linuxconf doesn't recycle old group numbers any more than it does old user IDs, so it won't happen by accident.

Modifying Group Membership

There are two ways to modify the list of users that belong to a group. You can either update each user account itself, or you can update the group definitions. In general, the fastest way is to update each of the group definitions. If you're planning on changing more information for each user than just the group information, then updating each user account may prove easier.

We'll start by detailing the group definitions method.

- Start linuxconf by typing `linuxconf` at the shell prompt.
- Open **Config** => **Users accounts** => **Normal** => **Group definitions**.

  Depending on the filter settings in **Control** => **Features**, you may see a filter screen. Use the filter to narrow the list, or just select **Accept** to bypass the filter.
- Select the group you wish to modify. This will open the `Group specification` screen (see Figure 3–9, *Group Specification Screen*).
- Add or remove each user from the **Alternate members** field. Make sure that all of the user names are separated by a space character.
- Select `Accept`, which can be found at the bottom of the screen.

This will automatically update each user account with the group showing up in the **Supplementary groups** field if added or absent if removed.

Adding and removing groups can also be done by modifying each individual user account.

- Start linuxconf by typing `linuxconf` at the shell prompt.
- Open **Config** => **Users accounts** => **Normal** => **User accounts**.

  You may see a filter screen, depending on the settings in **Control** => **Features**. Use the filter to narrow the list or select **Accept** to bypass the filter.
- On the **User accounts** screen (see Figure 3–3, *Users Accounts Screen*), select a user that you wish to update. You will be presented with the **User information** screen.
- Add or remove the desired groups from the **Supplementary groups** field. Each group should be separated by a space character.
- Once you've made all the changes you'd like, select `Accept` at the bottom of the screen.

This will automatically update the group definitions. Repeat the process for each user.

3.1.13 Filesystems

A filesystem is composed of files and directories, all starting from a single root directory. The root directory may contain any number of files and other directories, with each directory in turn following suit. The average filesystem often looks like an inverted tree with the directories as branches and the files as leaves. Filesystems reside on mass storage devices such as diskette drives, hard drives, and CD-ROMs.

For example, a diskette drive on DOS and Windows machines is typically referenced by `A:\`. This describes both the device (`A:`), and the root directory on that device (\). The primary hard drive on the same systems is typically referred to as the "C" drive because the device specification for the first hard drive is `C:`. To specify the root directory on the C drive, you would use `C:\`.

Under this arrangement, there are two filesystems -- the one on `A:`, and the one on `C:`. In order to specify *any* file on a DOS/Windows filesystem, you must either explicitly

specify the device on which the file resides, or it must be on the system's default drive (which is where DOS' C prompt comes from -- that's the default drive in a system with a single hard drive).

Under Linux, it is possible to link the filesystems on several mass storage devices together into a single, larger, filesystem. This is done by placing one device's filesystem "under" a directory on another device's filesystem. So while the root directory of a diskette drive on a DOS machine may be referred to as `A:\`, the same drive on a Linux system may be accessible as `/mnt/floppy`.

The process of merging filesystems in this way is known as **mounting**. When a device is mounted, it is then accessible to the system's users. The directory "under" which a mounted device's filesystem becomes accessible is known as the **mount point**. In the previous paragraph's example, `/mnt/floppy` was the diskette drive's mount point. Note that there are no restrictions (other than common conventions) as to the naming of mount points. We could have just as easily mounted the floppy to `/long/path/to/the/floppy/drive`.

One thing to keep in mind is that all of a device's files and directories are relative to its mount point. Consider the following example:

- A Linux System:
  - `/` -- system root directory
  - `/foo` -- mount point for the CD-ROM

- A CD-ROM:
  - `/` -- CD-ROM's root directory
  - `/images` -- A directory of images on the CD-ROM
  - `/images/old` -- A directory of old images

So, if the above describes the individual filesystems, and you mount the CD-ROM at `/foo`, the new operating system directory structure would be:

- A Linux System (with the CD-ROM mounted):
  - `/` -- System root directory
  - `/foo` -- CD-ROM root directory
  - `/foo/images` -- A directory of images on the CD-ROM
  - `/foo/images/old` -- A directory of old images

To mount a filesystem make sure to be logged in as root, or become root using the `su` command. For the latter, type `su` at the shell prompt and then enter the root password. Once you are root, type `mount` followed by the device and then the mount point. For example, to mount the first diskette drive on `/mnt/floppy`, you would type the command `mount /dev/fd0 /mnt/floppy`.

At installation, Red Hat Linux will create `/etc/fstab`. This file contains information on devices and associated mount points. The advantage to this file is that it allows you to shorten your mount commands and it controls which filesystems are automatically mounted when the system is booted.

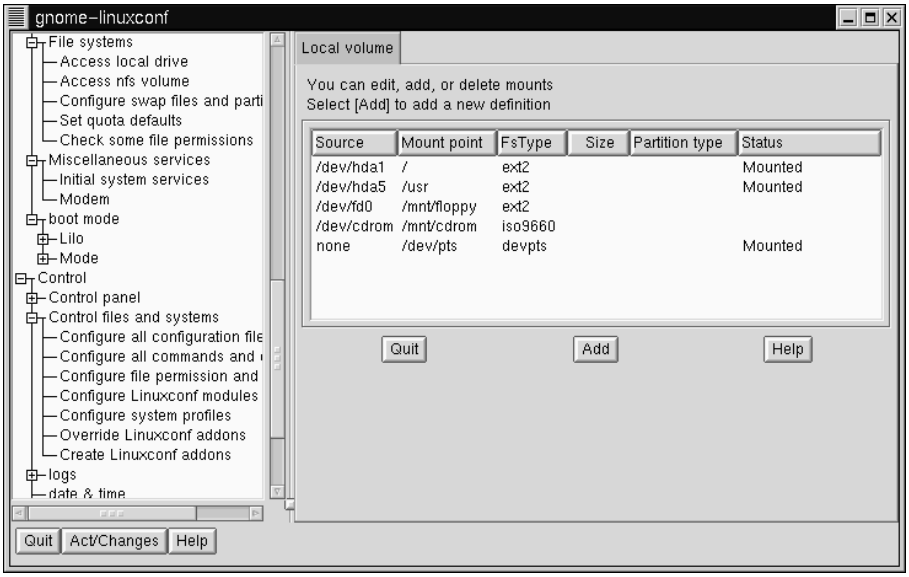
Using the information in `/etc/fstab`, you can type `mount` and then either the mount point or the device. The `mount` command will look for the rest of the information in `/etc/fstab`. It's possible to modify this file by hand, or by using linuxconf.

Reviewing Your Current Filesystem

We'll start by looking at your current directory structure.

- Open **Config** => **File systems** => **Access local drive.**

Figure 3–10 Local Volume Screen



The fields, as shown in Figure 3–10, *Local Volume Screen*, are:

- **Source**: The physical hardware; `hd` indicates an IDE hard drive, `fd` indicates a diskette drive, and `cdrom` typically indicates a CD-ROM drive. If your system has a SCSI drive, you will see an `sd` instead. More than one drive of a type are listed by letters, so `hda` represents the first IDE drive, while `hdb` would be the second. In some cases, you'll see numbers following these letters; on hard drives, the numbers represent the partitions on that drive, while for diskette drives, this number refers to the actual unit.
- **Mount point**: This is where in the system the drive is accessible from when mounted.
- **FsType**: The type of filesystem. A standard Linux partition uses the ext2 filesystem type. A filesystem type of vfat indicates a DOS filesystem with long filename support, while a fat filesystem type is for DOS filesystems supporting traditional 8.3 filenames. The iso9660 filesystem type indicates a CD-ROM drive.

Please Note

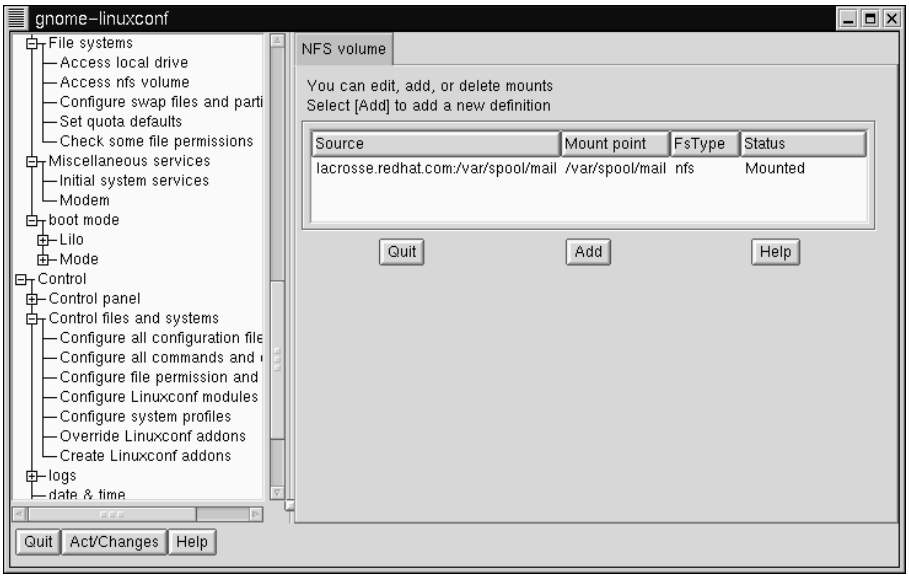
Red Hat Linux 6.2 can access FAT32 filesystems using the `vfat` filesystem type.

- **Size**: Size may indicate the size of the filesystem in megabytes (M), or it may not be filled in.
- **Partition type**: A description of the filesystem used on that partition (may not be filled in).
- **Status**: Whether the device is mounted or not.

Filesystems from other machines on a network may also be available. These can range from single small directories to entire volumes. No information on **Size** or **Partition type** is available for these partitions, either. Additional information on NFS filesystems (should you have any available) will be contained under:

Config => **File systems** => **Access nfs volume**

Figure 3–11 NFS Volume Screen



The screen (see Figure 3–11, *NFS Volume Screen* is similar to the **Local volume** screen, with some notable differences in the information provided for each entry:

- **Source**: This will be the name of the machine serving the filesystem, followed by the remote directory. For example: `foo:/var/spool/mail` where `foo` is the machine serving the directory, and `/var/spool/mail` is the directory being served.
- FsType -- This will always be "nfs."

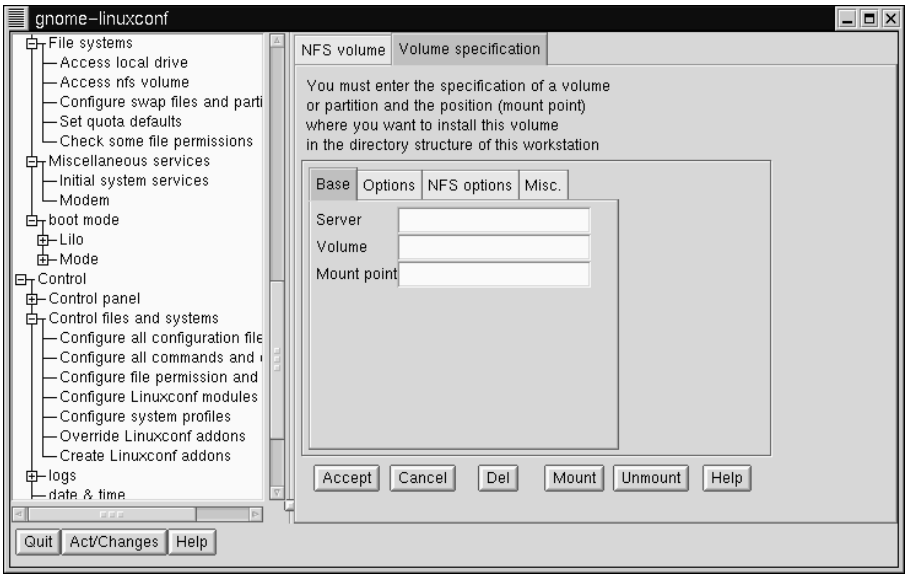
Adding NFS Mounts

NFS (Network File System) is a way for computers to share sections of their local filesystem across a network. These sections may be as small as a single directory, or include thousands of files in a vast hierarchy of directories. For example, many companies will have a single mail server with individuals' mail files served as an NFS mount to each users' local systems.

To add an NFS mount:

- Open **Config** => **File systems** => **Access nfs volume.**
- On the **NFS volume** screen, select `Add`.

Figure 3–12 Volume Specification Screen



The three fields on the **Base** tab are what you'll need to concern yourself with next (see Figure 3–12, *Volume Specification Screen*).

- **Server**: The host name of the machine the desired filesystem resides on. For example, foo.bar.com.
- **Volume**: The filesystem you wish to add. For example, `/var/spool/mail`.
- **Mount point**: Where in your system you want the remote file system accessible from. For example, `/mnt/mail`.

This is all you need to get the mount created. Linuxconf will update your `/etc/fstab` file accordingly. If you are aware of additional requirements, please read the help file on the **Volume specification** screen and see the `mount` man page for more information.

Once you have entered the information, select `Accept`.

3.1.14 Getting Connected with Linuxconf (Network Configuration)

The first thing to determine when getting hooked up is whether you're connecting to a local area network, such as a group of computers in an office, or a wide area network, such as the Internet. Before continuing, it's important to know what hardware you have and how you intend to connect. If you're going to dial into another computer, then make sure your modem is installed and that the cables are arranged properly. If you're using a network card, make sure it is installed properly and that the cables are correctly connected. Regardless of what network configuration you specify, if every phone line or cable is not in place, you'll never get connected. We'll start with modem connections and then move on to using network cards.

Adding Modem/PPP/SLIP connections

Red Hat Linux 6.2 uses the utility called RP3, or the RH PPP Dialer, a graphical tool which helps users configure and monitor network accounts -- particularly PPP accounts. (To read more about RP3, see the related chapter in the *Official Red Hat Linux Getting Started Guide*.) However, you can still depend on linuxconf to set up network configuration.

There are several pieces of information you will need to get from your ISP (Internet Service Provider) or systems administrator before getting your PPP or SLIP account working. In the case of some providers, you may have to sort through directions on how to set up a PPP connection on a Linux system. Some ISPs are ill-equipped to handle individuals using Linux. Don't worry, you can still get connected; you just need some additional information from your ISP.

The following is what you need for a connection with Red Hat Linux. The ISP representatives may respond that you don't need this information, or may suggest that you need more than this. Red Hat has streamlined the information needed using intelligent defaults and tools such as linuxconf to simplify this process for you. Unless they have a document specifically for Red Hat Linux, just request the information below and go from there. Specifically, you'll need:

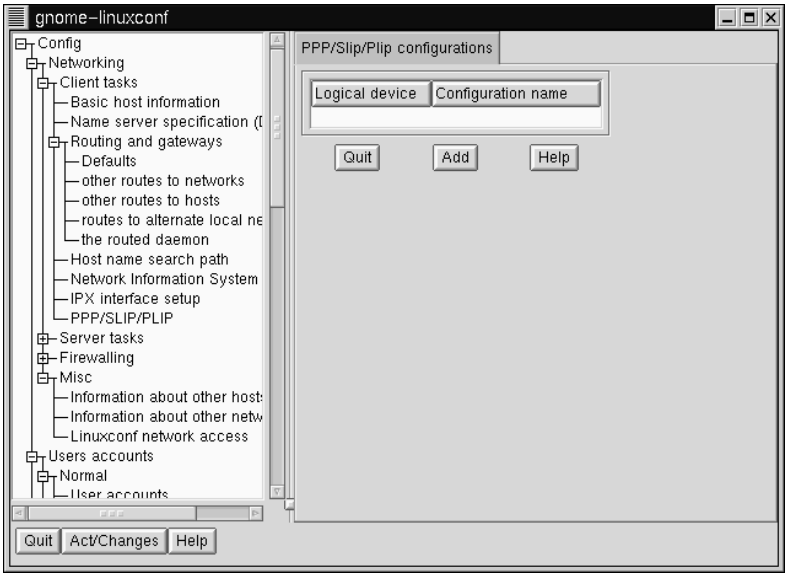
- The IP address for the ISP's domain name server
- The telephone number to dial

- Your login and password
- An IP address for your machine if the network you are connecting to isn't going to provide you with a dynamic one
- Information about whether your ISP uses an authentication method such as PAP, CHAP or MS-CHAP. If so, you will need a "secret" to enable authentication. The secret will be a word or sequence of characters. CHAP and MS-CHAP are not currently supported using linuxconf, and are rarely used.

Additional information which may be helpful, but isn't necessary includes a secondary name server address, and a search domain. Once you have all this information, you're ready to get connected.

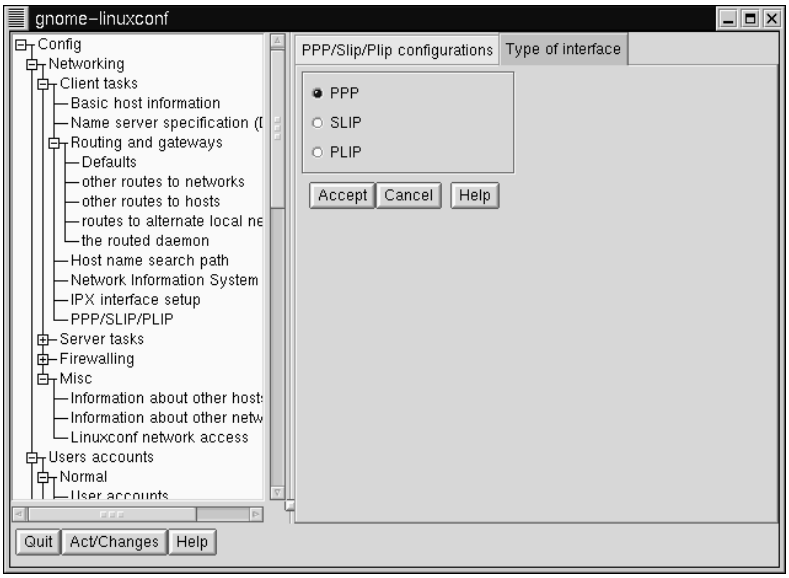
- Open **Config** => **Networking** => **Client tasks** => **PPP/SLIP/PLIP**.
- Select Add.

Figure 3–13 PPP/SLIP/PLIP Configurations Screen



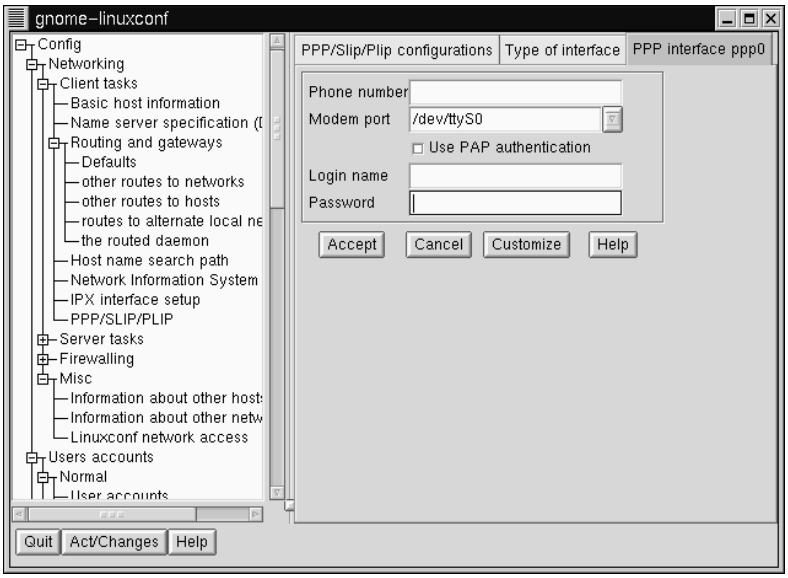
Initially there won't be any configurations specified. When you select Add you will be given a choice between PPP, SLIP and PLIP.

Figure 3–14 Type of Interface Screen



PPP is the most commonly used interface and is the default. To set up a PPP connection select `PPP` and select `Accept`.

Figure 3–15 PPP Interface Screen



You'll see the following fields:

- **Phone number** -- The number your modem will need to dial to access your ISP or the remote system
- **Modem port** -- Indicates where your modem is and should already be set
- **Use PAP authentication** check box -- Check if you know that the system you are dialing into requires this
- **Login name** -- Your login name for the PPP account
- **Password** -- Your password for the PPP account

Notice that the title bar is **PPP interface ppp0**. `ppp0` is the first PPP interface, `ppp1` would be the second and so on. It's important to keep track of which interface you're using if you have more than one. SLIP connections use `sl` instead of `ppp` for their interface prefix. With the exception of a PAP authentication option, the entry screens for adding a PPP or a SLIP account are identical.

Enter the complete phone number for the remote machine, and make sure to include any numbers required to access outside lines.

For example, if you need to dial 9 and then the number, and the computer you're connecting to has a telephone number of 555-0111, then you'd enter **95550111**.

The next information you're asked for is the modem port. This is a drop down box of available ports. If you're using a dual-boot Linux/Windows system and you know the COM port your modem is on, the following map may be of use:

Map to Windows COM ports are as follows:

- `ttyS0` -- COM1: under MS-DOS;
- `ttyS1` -- COM2: under MS-DOS;
- `ttyS2` -- COM3: under MS-DOS;
- `ttyS3` -- COM4: under MS-DOS.

The login name is the one for the PPP account. The password you enter will be shown in plain text, so be careful who you have around when you enter it! If you will be using PAP authentication, check the box; when you've entered the other required information, select the `Customize` button at the bottom of the screen.

If you're using PAP, select the **PAP** tab and enter your username and then the secret the ISP has provided you in the **Secret field**. The other defaults should be sufficient, but if you need to, you can edit the initial settings using the `Customize` option.

Modifying a PPP or SLIP Configuration

You can edit an existing configuration as well as delete it by selecting it from the list on the **PPP/SLIP/PLIP configurations** screen.

- Open **Config** => **Networking** => **Client tasks** => **PPP/SLIP/PLIP**.
- You will then be presented with the `PPP/SLIP/PLIP configurations` screen (see Figure 3–13, *PPP/SLIP/PLIP Configurations Screen*). Select the configuration you would like to modify or delete.

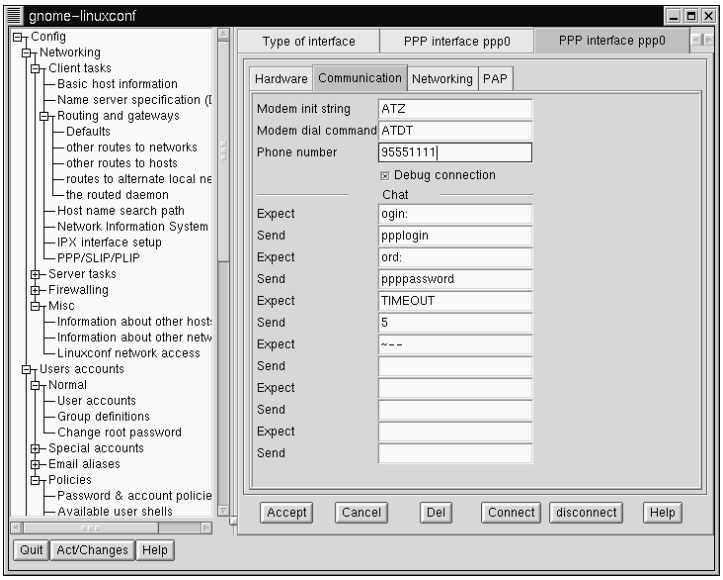
  This will open the appropriate interface screen for your configuration. If you wish to delete the configuration, select `Del` at the bottom of the screen. The Modem port is on the **Hardware** tab and is a drop down menu. If you want to change the

other settings you entered when you originally created the configuration, select the **Communication** tab. The first **Send** field contains your login, and the next **Send** field contains your password. The **Expect** fields correspond to the `login:` and `password:` prompts, which explains the `ogin:` and `ord:` entries.

Please Note

The `ogin:` and `ord:` entries may not be the same for your system. Different ISP/PPP servers may use different text and should be changed to fit the needs of whatever server type you are logging in to. Instead, you may see such prompts as `User ID` and `authentification`.

Figure 3–16 PPP Customization – The Communication Tab



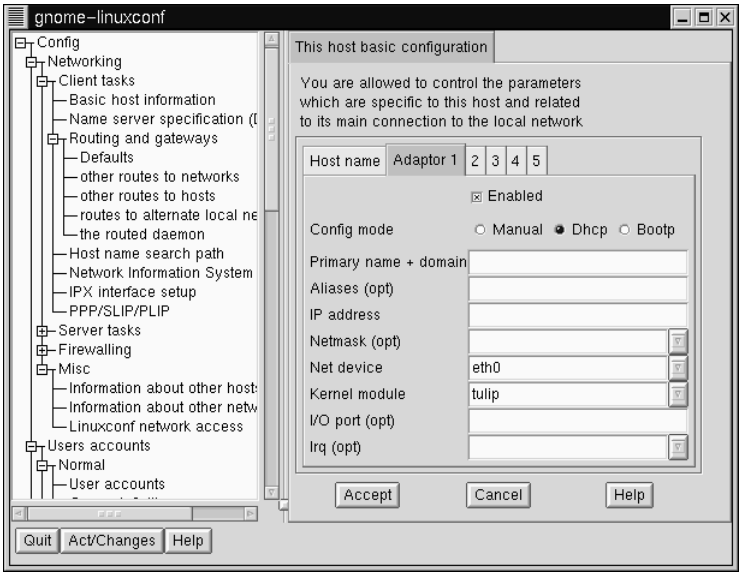
Once you have made your changes, you can test to see if your configuration is working. Select `Connect` from the bottom of the screen. This will attempt to connect you to the remote system using the information you've entered.

Other Network Connections

Setting up a network connection over Ethernet requires an entirely different type of setup. Network connections to Token Ring or ARCnet networks follow a similar procedure, but will not be discussed here.

- First you will need to have an Ethernet card installed.
- Start linuxconf by typing `linuxconf` at the shell prompt.
- Open **Config** => **Networking** => **Client tasks** => **Basic host information**. The `Host name` tab will request a host name, which should be specified by default unless you did not setup your networking during the installation process. If it is not already specified, please take the time now to configure it. It should be specified as `localhost.localdomain`. Skip this tab. Select the tab for `Adaptor 1`.

Figure 3–17 Adaptor 1



The first item on this screen is a check box to indicate whether this adaptor is enabled or not. It should be checked if this is the one you intend to use. Below that is a choice of **Config modes**. **Manual** means that you will be providing all the information and entering it yourself. **Dhcp** and **Bootp** mean that your machine will be getting its network configuration information from a remote DHCP or BOOTP server. If you're not sure what option to choose, talk to your network administrator.

Required fields for DHCP or BOOTP:

- **Net device** -- The type of network card you are using; for example, eth0 would be the appropriate entry to use the first Ethernet card.
- **Kernel module** -- The correct module based on your network card; for further information see the list below.

For DHCP and bootp configurations you only need to specify the **Net device** and the **Kernel module**. For the **Net device**, you will choose from a list where the **eth** prefix represents Ethernet cards, **arc** specifies an ARCnet card and **tr** specifies Token Ring cards. A complete list of network cards and their respective modules can be found in Appendix A, *General Parameters and Modules*. For the most up-to-date list, please see our website at:

`http://www.redhat.com/support/hardware`

The netmask information may be set by default, although depending on what kind of network you are setting up, or becoming a part of, you may need to specify this. If you are connecting to an ISP, ask them for the information. Most likely it will be 255.255.255.0.

Required fields for manual configuration:

- **Primary name + domain** -- The primary name is the name of your computer, while the domain is how your network is specified. For example, `foo.bar.com`; `foo` is the primary name and `bar.com` is the domain.
- **IP address** -- The address of the machine and will follow the pattern of *x.x .x.x*. For example, 192.168.0.13.
- **Net device** -- The type of network card you are using; eth0 would be the appropriate entry to use the first Ethernet card.

- **Kernel module** -- The correct module based on your network card.

Information on net devices and kernel modules is described above. The appropriate primary name + domain and IP address will depend on whether you are adding the computer to an existing network or creating a new network. For connecting to an existing network, contact your network administrator for the information. Getting a network connected to the Internet is beyond the scope of this book, and we recommend the following starting point:

TCP/IP Network Administration, 2nd Edition, by Craig Hunt (O'Reilly and Associates).

If you're setting up a private network that won't *ever* be connected to the Internet, then you can choose any primary name + domain name you would like and have several choices for IP addresses (See Table 3–1, *Addresses and Examples*).

Table 3–1 Addresses and Examples

Addresses available	Examples
10.0.0.0 - 10.255.255.255	10.5.12.14
172.16.0.0 - 172.31.255.255	172.16.9.1, 172.28.2.5
192.168.0.0 - 192.168.255.25	192.168.0.13

The three sets of numbers above correspond to class a, b, and c networks respectively. The classes are used to describe the number of IP addresses available as well as the range of numbers. The numbers above have been set aside for private networks.

Please Note

You should not use these IP addresses if you connect to the Internet since 192.168.0.* and 192.168.255.* are not reliably considered private. If you want your network to be connected to the Internet, or think you might want to at some point in the future, do yourself a favor and get yourself non-private addresses now.

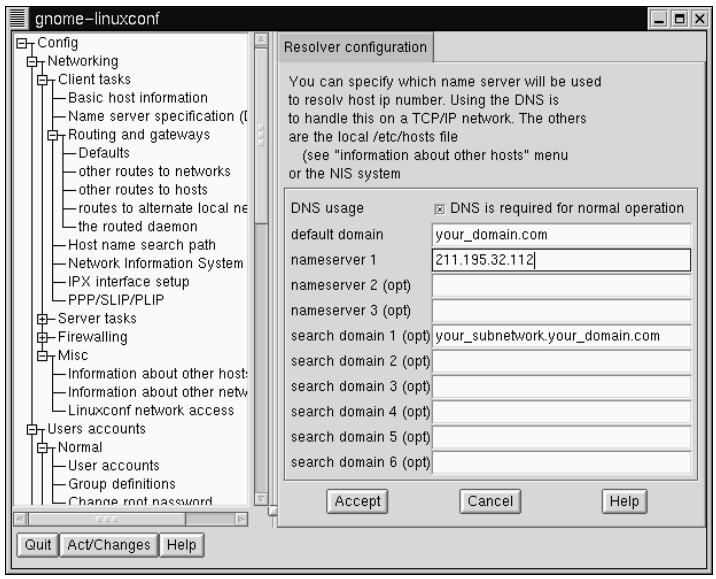
Name Server Specification

A name server and default domain are also needed to establish a network connection. The name server is used to translate host names such as `private.network.com` to their corresponding IP address such as `192.168.7.3`.

The default domain tells the computer where to look if a fully qualified hostname isn't specified. Fully qualified means that the full address is given, so `foo.redhat.com` is the fully qualified hostname, while the hostname is simply `foo`. If you specified your default domain as `redhat.com`, then you could use just the hostname to connect successfully. For example `ftp foo` would be sufficient if your search domain is `redhat.com`, while `ftp foo.redhat.com` would be required if it wasn't.

To specify the nameserver, open **Config** => **Networking** => **Client tasks** => **Name server specification (DNS).**

Figure 3–18 Resolver Configuration Screen



Nameservers are ranked according to the order in which they are accessed, so it's not unusual to see nameservers referred to as primary, secondary, tertiary and so on

down the list if more than one is specified. Each of these must be an IP address and not a name, since the computer has no way to resolve the name until it connects to a nameserver.

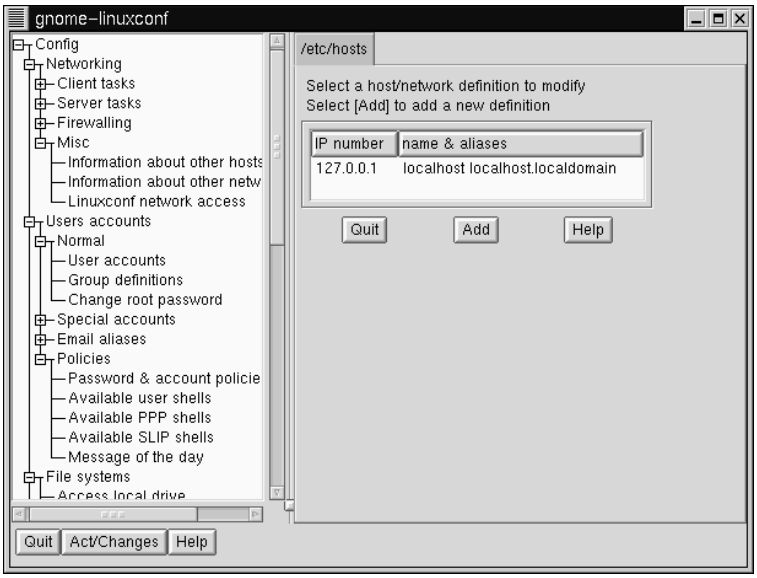
In addition to a default domain, you can also specify search domains. Search domains work differently; they progress from one to six in a similar manner to the nameserver. However, they all take precedence over the default domain! Keep this in mind when specifying search domains. Search domains are not commonly used.

The one item not yet covered is the check box for DNS usage. If you are running a small private network with no Internet connection, then using `/etc/hosts` files and keeping them all synchronized will work. As you add more and more machines, the complexity increases until it is easier to have a single machine run a DNS than to continue to sync `/etc/hosts` files.

Another reason for not using DNS is if your network is going to use NIS instead. Note that NIS can be used in conjunction with DNS. So to sum it all up, unless you know why using `/etc/hosts` or NIS would be best for your situation, DNS is probably going to be your best choice.

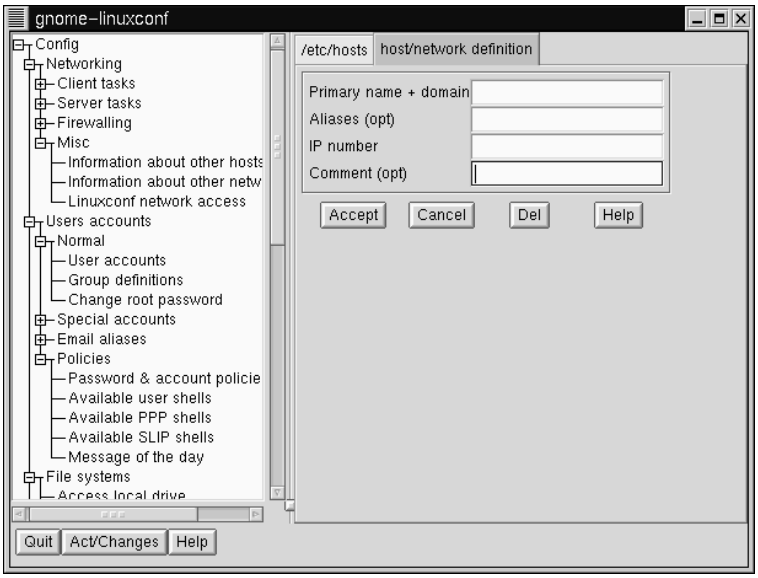
You can add, modify, or delete entries from the `/etc/hosts` file using linuxconf. Open **Config** => **Networking** => **Misc** => **Information about other hosts.**

Figure 3–19 `/etc/hosts` Screen



To modify or delete an entry select it. To delete the entry, select **Del** at the bottom of the **host/network definition** screen.

Figure 3–20 Host/Network Definition Screen



To modify it, change the information as necessary. To add a new entry, select `Add` at the bottom of the **/etc/hosts** screen. This will also open the **host/network definition** screen.

Required Fields:

- **Primary name + domain** -- The primary name is the name of the computer, while the domain is how the network it is attached to is specified. For example, given `foo.bar.com`, `foo` is the primary name and `bar.com` is the domain.
- **IP number** -- Also referred to as IP address; this is the address of the machine and will follow the pattern of *x.x. x.x.* For example, 192.168.0.13.

Optional Fields:

- **Aliases** -- A shorthand for the fully qualified domain name. This is often the same as the primary name. So, for example, if the fully qualified domain name is foo.bar.com, you could select foo as the alias.

- **Comment** -- A comment on the machine. For example, "The remote nameserver."

Once finished, select `Accept`.

3.1.15 Finding Your Way Through linuxconf

This table provides a quick reference for this chapter. Unfortunately, it doesn't provide a complete quick reference for linuxconf, which has many more capabilities than this documentation provides.

Table 3–2 Linuxconf **Quick Reference**

What do you want to do?	Where to find it in linuxconf
Add/modify/disable/delete a user account	**Config** => **Users accounts** => **Normal** => **User accounts**
Change a user's password	**Config** => **Users accounts** => **Normal** => **User accounts**
Change the root password	**Config** => **Users accounts** => **Normal** => **Change root password**
Configure networking	**Config** => **Networking** => **Client tasks** => **Basic host information**
Create/delete a group	**Config** => **Users accounts** => **Normal** => **Group definitions**
Create/modify modem/PPP/SLIP connections	**Config** => **Networking** => **Client tasks** => **PPP/SLIP/PLIP**
Edit parameters for passwords	**Users Accounts** => **Password & Account Policies**
Disable tree menu	**Control** => **Control files and systems** => **Configure linuxconf modules**
Enable Web-based access to linuxconf	**Config** => **Networking** => **Misc** => **Linuxconf network access**

What do you want to do?	Where to find it in linuxconf
Modify `/etc/hosts`	**Config** => **Networking** => **Misc** => **Information about other hosts**
Modify group membership	**Config** => **Users accounts** => **Normal** => **Group definitions** or **Config** => **Users accounts** => **Normal** => **User accounts**
Set filter parameters	**Control** => **Features**
Specify a nameserver (DNS)	**Config** => **Networking** => **Client tasks** => **Name server specification (DNS)**
View filesystem	**Config** => **File systems** => **Access local drive** or **Config** => **File systems** => **Access nfs volume**

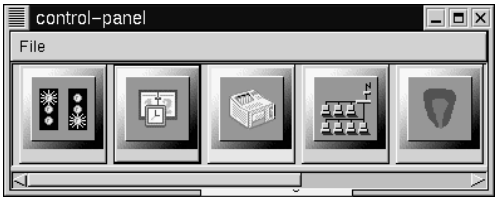
3.2 System Configuration with the Control Panel

Please Note

Most of what can be done with the control panel applications can also be done using linuxconf. In addition, linuxconf supports both character-cell *and* graphical user interfaces. Please refer to Section 3.1, *System Configuration with* linuxconf for an introduction to linuxconf.

The control panel is a launching pad for a number of different system administration tools (see Figure 3–21, *The Control Panel*). These tools make your life easier by letting you configure things without remembering configuration file formats and awkward command line options.

Figure 3–21 The Control Panel



To start the control-panel, start the X Window System as root with `startx` and type `control-panel` in an Xterm. You will need to be root to run the control-panel tools successfully. You can do this as well if you already have X running as a normal user. Just type `su -c control-panel` and then type the root password when prompted. If you plan to do other tasks as root, you could type `su` followed by the root password when prompted.

Please Note

If you are not running X as root, you may need to give root access to your system's X server. To do this, enter the following command on a *non-root* terminal window:

```
xhost +localhost
```

After starting the control panel, simply clicking on an icon starts up a tool. Please note that you are not prevented from starting two instances of any tool, but doing so is a very bad idea because you may try to edit the same files in two places and end up overwriting your own changes.

Please Note

If you do accidentally start a second copy of a tool, you should quit it immediately. Also, do not manually edit any files managed by the control-panel tools while the tools are running. Similarly, do not run any other programs (such as linuxconf) that may change those files while the tools are running.

3.2.1 Printer Configuration

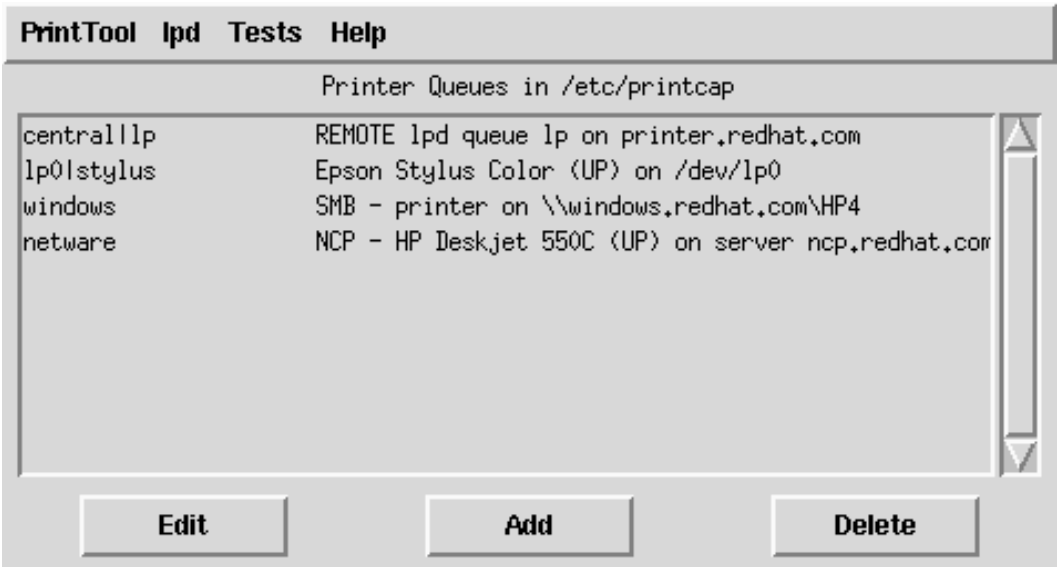
Please note that the *Official Red Hat Linux Getting Started Guide* contains more up-to-date documentation on printtool, so be sure to check that document before using printtool.

The printer configuration tool (printtool) maintains the `/etc/printcap` file, print spool directories, and print filters. The filters allow you to print many different types of files, including:

- plain text (ASCII) files
- PostScript files
- TeX `.dvi` files
- GIF, JPEG, TIFF, and other graphics formats
- RPMs

In other words, simply printing a GIF or RPM file using the `lpr` command will result in the printer doing "the right thing."

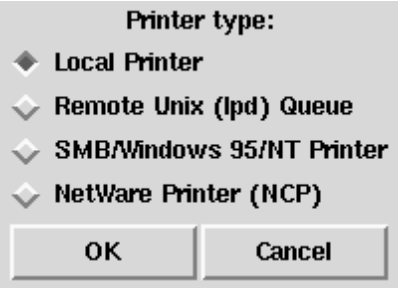
Figure 3–22 Print Tool



In order to create a new **print queue**, choose Add. Then, select what type of printer is being added. There are four types of print queues which can be configured with printtool:

- Local print queues are for printers attached to a printer or serial port on your Red Hat Linux system
- Remote print queues are attached to a different system which you can access over a TCP/IP network
- SMB print queues are attached to a different system which uses LAN-Manager-type (SMB) networking
- NCP print queues are attached to a different system which uses Novell's NetWare network technology

Figure 3–23 Selecting a Printer Type



After choosing the printer type, a dialog box requests further information about the print queue (see Figure 3–24, *Adding a Local Printer*). All types of print queues require the following information:

- Queue Name -- What the queue will be called. Multiple names can be specifed with the `|` (pipe) character separating entries.
- Spool Directory -- This is the directory on the local machine where files are stored before printing occurs. Be careful to not have more than one printer queue use a given spool directory.
- File Limit -- Maximum size print job accepted, in kilobytes (1 kb = 1024 bytes). A size of 0 indicates no limit should be imposed.
- Input Filter -- Filters convert printed files into a format the printer can handle. Press `Select` to choose the filter which best matches your printer (see Figure 3–25, *Configuring a Print Filter*).

  In addition to configuring print queues able to print graphical and PostScript output, you can configure a **text-only** printer, which will only print plain ASCII text. Most printer drivers are also able to print ASCII text without converting it to PostScript first; simply choose `Fast text printing` when you configure the filter.

Please Note

This only works for non-PostScript printers.

- Suppress Headers -- Check this if you don't want a header page printed at the beginning of each print job.

For **local** printers, the following information is also required:

- Printer Device -- Usually `/dev/lp1`; the name of the port which the printer is attached to. Serial printers are usually on `/dev/ttyS?` ports. Note that you will need to manually configure serial parameters.

Figure 3–24 Adding a Local Printer

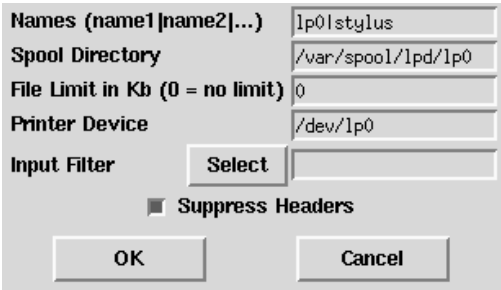
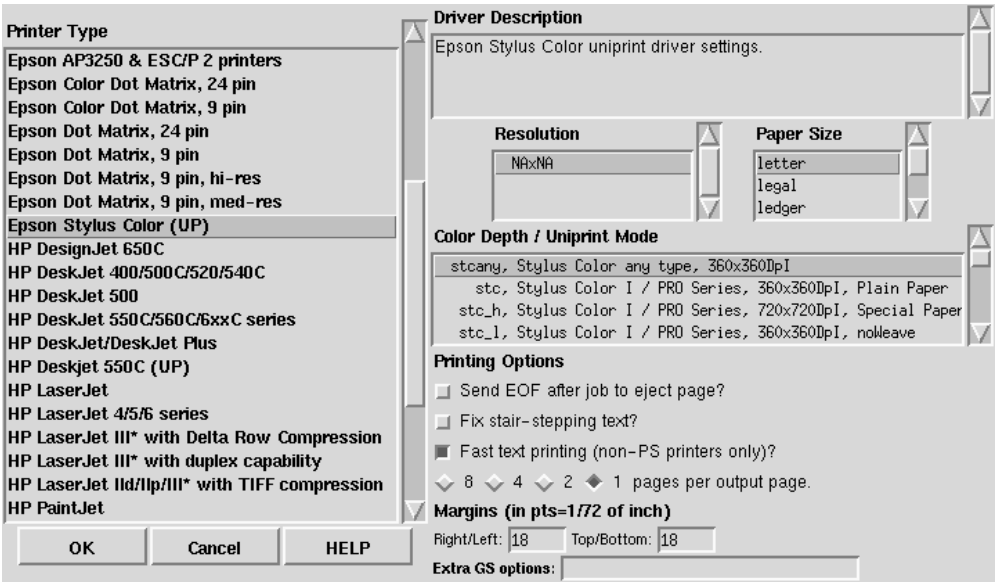


Figure 3–25 Configuring a Print Filter



For **remote** printers, the dialog box contains additional fields; fill in the following information:

- Remote Host -- Hostname of the remote machine hosting the printer.
- Remote Queue -- Name of the queue to print to on the remote machine.

The remote machine must be configured to allow the local machine to print on the desired queue. Typically `/etc/hosts.lpd` controls this.

Figure 3–26 Adding a Remote Printer

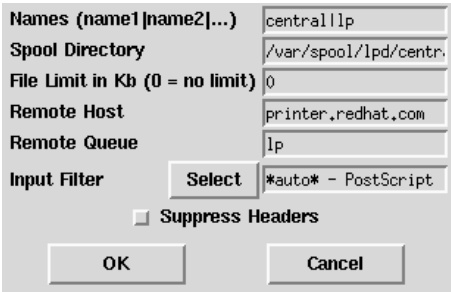


Figure 3–27 Adding an NCP Printer

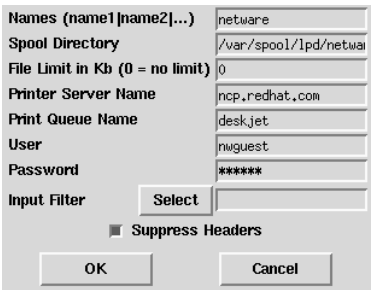
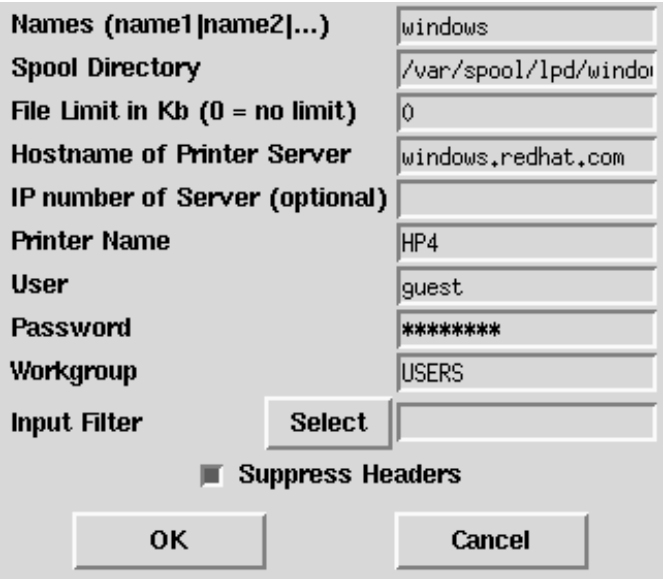


Figure 3–28 Adding an SMB Printer



For SMB and NCP printers, fill in the following information:

- Hostname of Printer Server -- Name of the machine to which the printer you want to use is attached.
- IP number of Server -- The IP address of the machine to which the printer you want to use is attached; this is optional and only relevant for SMB printers.
- Printer Name -- Name of the printer on which you want to print.
- User -- Name of user you must login as to access the printer (typically `guest` for Windows servers, or `nobody` for `samba` servers).
- Password -- Password (if required) to use the printer (typically blank). Someone should be able to tell you this if you do not already know it.

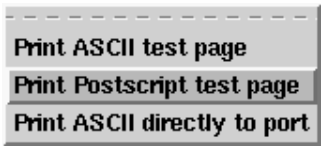
Please Note

If you require a username and password for an SMB (LAN Manager) or NCP (NetWare) print queue, they are stored unencrypted in a local script. Thus, it is possible for another person to learn the username and password. It is therefore recommended that the username and password for use of the printer to be different than that for a user account on the local Red Hat Linux system, so that the only possible security compromise would be unauthorized use of the printer. If there are file shares from the SMB server, it is recommended that they also use a different password than the one for the print queue.

After you have added your print queue, you may need to restart the printer daemon (`lpd`). To do so, choose `Restart lpd` from the `lpd` menu.

You may print a **test page** for any print queue you have configured. Select the type of test page you would like to print from the `Tests` menu.

Figure 3–29 Printing a Test Page



3.2.2 Loading Kernel Modules

The Linux kernel has a modular design. At boot time, only a minimal **resident** kernel is loaded into memory. Thereafter, whenever a user requests a feature that is not present in the resident kernel, a kernel **module** is dynamically loaded into memory. After a specified period of inactivity, the module may be removed from memory. This design promotes leanness and efficiency.

The mechanism that supports dynamic loading of modules is a kernel thread called `kmod`. When the kernel requests a module, kmod wakes up and calls `modprobe`(8) to get it.

When you install Red Hat Linux, the hardware on your system is probed and you provide information about how the system will be typically used and which programs should be loaded. Based on this probing and the provided usage information, the installation program decides which features to compile into the resident kernel and which to put in loadable modules, and sets up the dynamic loading mechanism to work transparently. But this is a highly configurable procedure. If you build your own custom kernel, you can make all of these decisions for yourself.

If you add new hardware after installation requiring support provided in a kernel module, you need to set up the dynamic loading mechanism. You do this by editing the module configuration file, `/etc/conf.modules`.

For example, if at the time you installed Red Hat Linux your system included a model SMC EtherPower 10 PCI network adapter, the module configuration file will contain this line: `alias eth0 tulip`. If, after installation, you install a second identical network adapter to your system, add this line to `/etc/conf.modules`:`alias eth1 tulip`.

See Appendix A, *General Parameters and Modules* for an alphabetical list of kernel modules and the hardware the modules support.

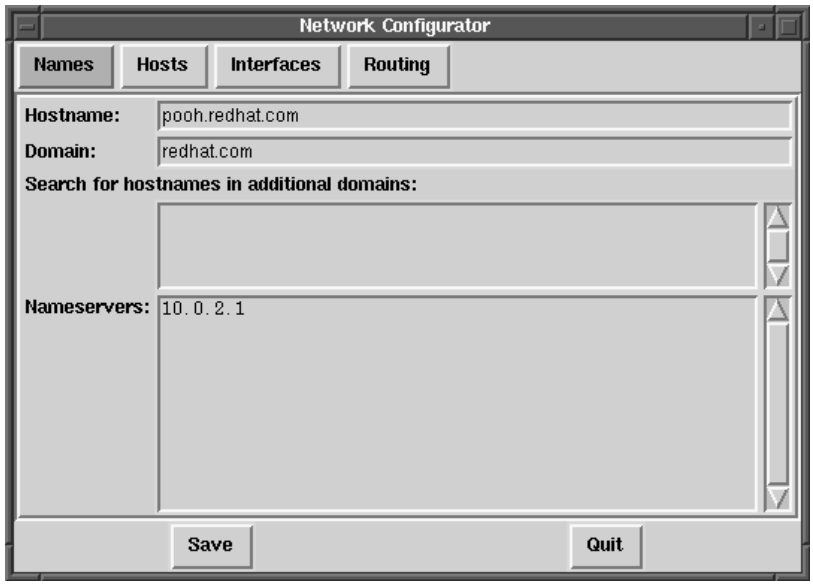
3.2.3 Network Configuration

Please Note

Documentation on network configuration using linuxconf can be found in Section 3.1.14, *Getting Connected with* Linuxconf *(Network Configuration)*.

The network configuration tool (`netcfg`) shown in Figure 3–30, *Network Configuration Panel* is designed to allow easy manipulation of parameters such as IP address, gateway address, and network address, as well as name servers and `/etc/hosts`.

Figure 3–30 Network Configuration Panel

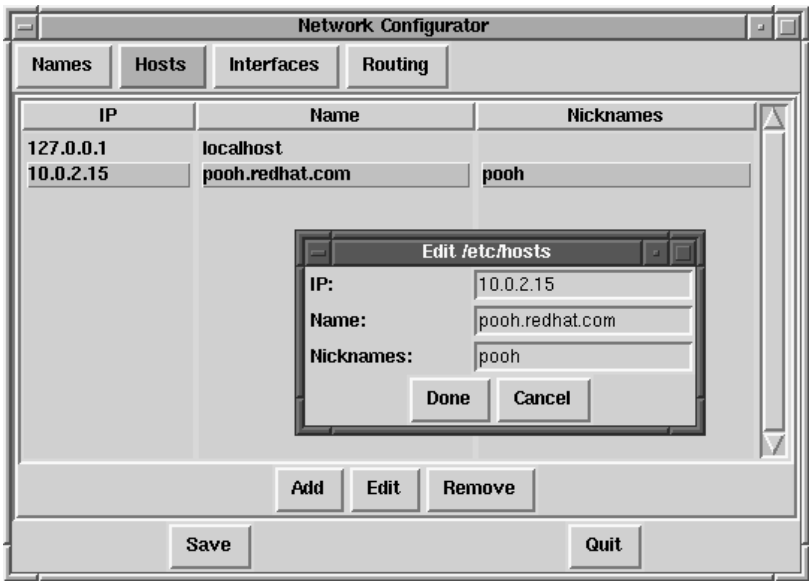


Network devices can be added, removed, configured, activated, deactivated and aliased. Ethernet, arcnet, token ring, pocket (ATP), PPP, SLIP, PLIP and loopback devices are supported. PPP/SLIP/PLIP support works well on most hardware, but some hardware setups may exhibit unpredictable behavior. When using the Network Configuration Tool click `Save` to write your changes to disk, to quit without making any changes select `Quit`.

Managing Names

The `Names` panel of the Network Configuration tool serves two primary purposes: setting the hostname and domain of the computer, and determining which name server will be used to look up other hosts on the network. The Network tool is not capable of configuring a machine as a nameserver. To edit a field or add information to a field, simply click on the field with the left mouse button and type the new information.

Figure 3–31 Adding/Editing Hosts



Managing Hosts

In the `Hosts` management panel you have the ability to add, edit, or remove hosts from the `/etc/hosts` file. Adding or editing an entry involves identical actions. An edit dialog box will appear, simply type the new information and click `Done` when you are finished. See Figure 3–31, *Adding/Editing Hosts* for an example.

Adding a Networking Interface

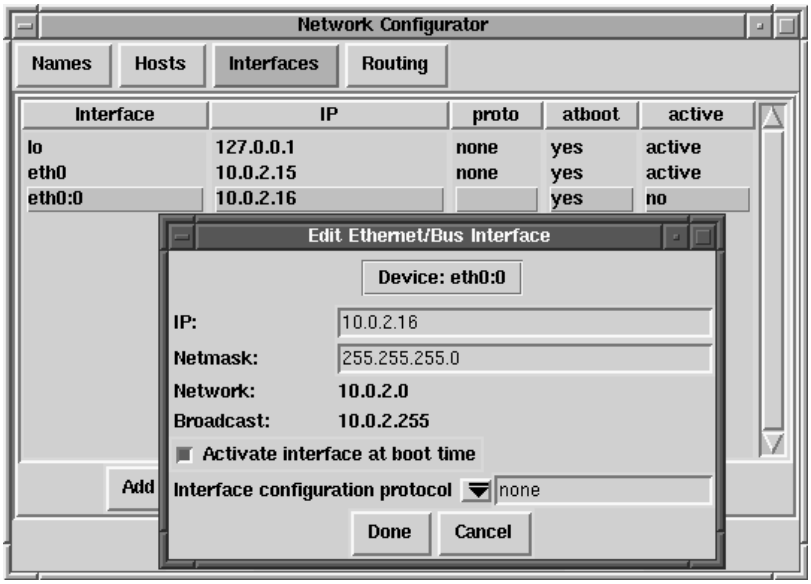
If you have added a networking interface to your machine since installing Red Hat Linux, or you didn't configure your Ethernet card at install time, you can configure it with a few clicks of a mouse.

Please Note

You may need to configure `kerneld` to load a driver for the network interface you are adding (e.g., `eth0`); see Section 3.2.2, *Loading Kernel Modules* for more information.

Begin adding an interface by clicking on `Interfaces` in the main panel. This will bring up a window of configured devices with a row of available options, see Figure 3–32, *Configured Interfaces*.

Figure 3–32 Configured Interfaces



To add a device, first click the `Add` button then select the type of interface you want to configure from the box that appears (See Figure 3–33, *Choose Interface Type*).

Figure 3–33 Choose Interface Type

Please Note

There is now a `clone` button available in `netcfg`. This button can be used to create a "clone" of an already-existing interface. By using clone interfaces, it is possible for a laptop to have one Ethernet interface defined for a work LAN, and a clone Ethernet device defined for a home LAN.

PPP Interface

Adding a PPP interface can be as simple as supplying the phone number, login name and password in the `Create PPP Interface` dialog shown in Figure 3–34, *Create PPP Interface*. If you must use PAP authentication for your PPP connection, choose `Use PAP authentication`. In many cases some degree of customization will be needed to establish a PPP connection. Choosing the `Customize` button will allow you to make changes to the hardware, communication, and networking settings for the PPP interface.

Figure 3–34 Create PPP Interface

SLIP Interface

In order to configure a SLIP interface you must first supply a phone number, login name, and password. This will supply the initial parameters for the chat script needed to establish a SLIP connection. When you choose `Done`, a dialog titled `Edit SLIP Interface` appears that enables you to further customize the hardware, communication and networking parameters for your SLIP interface.

PLIP Interface

To add a PLIP interface to your system you only have to supply the IP address, the remote IP address, and the Netmask. You can also select if you want to activate the interface at boot time.

Ethernet, Arcnet, Token Ring and Pocket Adaptor Interfaces

If you are adding an ethernet, arcnet, token ring or pocket adapter to your computer you will need to supply the following information:

- Device -- This is determined by netconfig based on the devices already configured.
- IP Address -- Enter an IP address for your network device.
- Netmask -- Enter the network mask for your network device.

The network and broadcast addresses are calculated automatically based on the IP address and netmask you enter.

- Activate interface at boot time:

  If you want the device to be configured automatically when your machine boots select this by clicking on the box.

- Allow any user to (de)activate interface:

  Check this if you want any user to be able to activate or deactivate the interface.

- Interface configuration protocol:

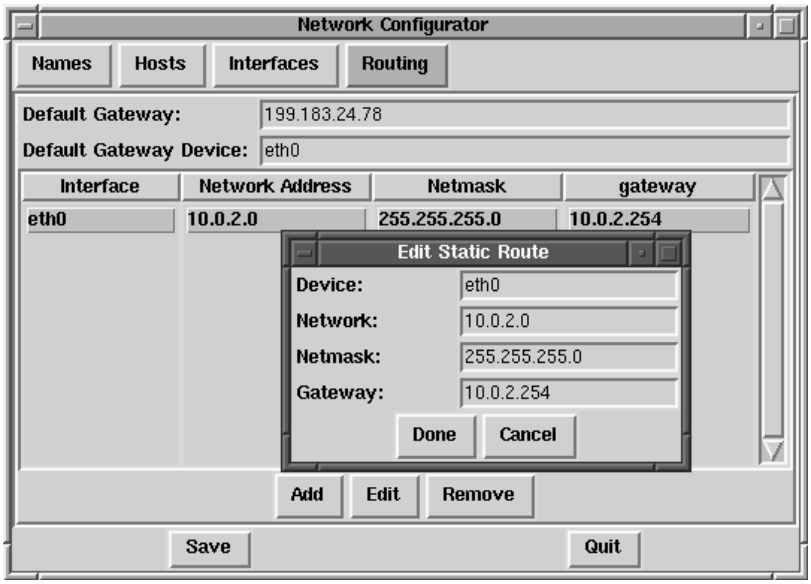
  If you have a BOOTP or DHCP server on your network and would like to use it to configure the interface, choose the appropriate option; otherwise, choose `none`.

After providing the configuration information for your new device, click `Done`. The device should appear in your `Interfaces` list as an inactive device. (The active column should have a label of `no`.) To activate the new device, first select it with a mouse click and then choose on the `Activate` button. If it does not come up properly, you may need to reconfigure it by choosing `Edit`.

Managing Routes

In the Routes management screen you have the ability to add, edit, or remove static networking routes. Adding or editing an entry involves identical actions, just like the Hosts panel. An edit dialog box will appear; simply type the new information and click `Done` when you are finished. See Figure 3–35, *Adding/Editing Routes* for an example.

Figure 3–35 Adding/Editing Routes



3.2.4 Time and Date

The "time machine" allows you to change the time and date by clicking on the appropriate part of the time and date display and clicking on the arrows to change the value.

The system clock is not changed until you click on the **Set System Clock** button.

Click on **Reset Time** to set the time machine time back to that of the system.

Please Note

Changing the time can seriously confuse programs that depend on the normal progression of time, and could possibly cause problems. Try to quit as many applications and processes as possible before changing the time or date.

---

4 PowerTools

4.1 PowerTools Packages

Red Hat PowerTools is a collection of software packages built for the Red Hat Linux 6.2 operating system. PowerTools includes the latest versions (as of this product's release date) of hundreds of programs -- so finding an interesting application should be easy.

Among the many applications are audio programs, chat clients, development tools, editors, file managers, emulators, games, graphics programs, productivity applications, math/statistics packages, systems administration and network management tools, and window managers.

Additionally, for PalmPilot personal organizer users, the `pilot` directory on the CD-ROM contains applications for use with PalmPilots.

Now that you know what you can find on PowerTools, you may want to know how to install them. See Section 4.3, *Installing PowerTools Packages* for installation information.

4.2 Reading the Contents of the CD-ROM

You can read the contents of the PowerTools CD-ROM from a shell prompt (either in a terminal window or in console mode). Then mount the CD-ROM (`mount /mnt/cdrom`). Once this is done, change directories by issuing the `cd /mnt/cdrom` command. Finally, type `less CONTENTS` to view the available applications.

4.3 Installing PowerTools Packages

4.3.1 Installing PowerTools in a GUI Environment

If you're using GNOME or KDE, place the CD-ROM in your CD-ROM drive. You'll be prompted for the root password (you must to be root in order to install packages). After you type in the root password, either the Gnome-RPM or the Kpackage package

management program will start automatically (depending on you GUI environment) and can be used to install PowerTools.

See Chapter 7, *Gnome-RPM* for specific instructions on how to use Gnome-RPM. See http://www.general.uwa.edu.au/u/toivo/kpackage/ for more information on how to use Kpackage.

If you're not using GNOME or KDE, you'll need to use the shell prompt to install PowerTools. See Section 4.3.2, *Installing PowerTools from the Shell Prompt* for more information.

4.3.2 Installing PowerTools from the Shell Prompt

First, mount the PowerTools CD-ROM on your CD-ROM drive.

Place the PowerTools CD in your CD-ROM drive. As root, type the following:

```
# mount -t iso9660 /dev/cdrom /mnt/cdrom
#
```

Please note

On your system, you or the system administrator may already allow users (instead of only root) to mount the CD-ROM drive. Users have this privilege if the `user` option is included in the `/dev/cdrom` line in the `/etc/fstab` file. However, keep in mind that you must be logged in as root to install any PowerTools RPMs.

After you've mounted the drive, `cd` to the mounted CD-ROM directory with the following command:

```
# cd /mnt/cdrom
```

When you list the contents of the CD with `ls`, you'll see the following directories: `SRPMS`, `alpha`, `i386`, `i586`, `i686`, `noarch`, `nosrc`, `sparc`, and `sparc64`. The `SRPMS` directory contains the PowerTools source RPMs. The `alpha`, `i386`,

`sparc`, and `sparc64` directories contain the RPMs for the three specified operating system architectures. The `noarch` directory contains RPMs which are not architecture-specific and should install on any of the three architectures.

The `i386` path is used as a general example. You should substitute the correct directory for `i386`, depending upon your architecture and which package you're installing.

`cd` to the `i386` directory:

```
# cd i386
#
```

List the RPM files in the directory with `ls` to see the complete list of RPM packages included for Intel-compatible systems.

You will probably want more information about a specific package before you can decide whether you want to install it. You can use RPM's querying capability to find out more information about the packages, such as the packages' functions and origination. See Section 6.3, *Impressing Your Friends with RPM* for instructions on how to query packages using RPM.

You can install your selected packages with RPM. RPM is a powerful command line-driven package management system. See Chapter 6, *Package Management with RPM* for more information on how to use RPM to install and manage PowerTools packages.

Once you have finished installing your packages, you'll need to unmount your CD-ROM. Type `umount /mnt/cdrom` to unmount the CD-ROM. Type `eject /mnt/cdrom` and the CD-ROM drive will open so that you can remove the CD.

5 Working with Update Agent

One of the more convenient ways to keep your Red Hat Linux system up to date is through Update Agent. This utility allows you to automatically locate, download and install updated packages from Red Hat for your system.

Through a secure connection to `priority.redhat.com`, Red Hat's own fast access FTP server, you're assured that new packages are specific to your Red Hat Linux system, rather than might be the case with upgrades you could find through another source. Upgrades and bug fixes are presented to you on a single "page" in your Web browser. You can select individual packages or all of the packages for download. You can also configure Update Agent to automatically install the new packages once they've been downloaded.

By using `priority.redhat.com`, you can save yourself the time and trouble of searching for and downloading individual packages from numerous FTP servers.

Please note

To use Update Agent, you must first register your copy of Red Hat Linux 6.2. To register, go to http://www.redhat.com/now. Once you register, you can select a **user name** and **password** which you should use to configure Update Agent. Please refer to Section 5.2, *Configuring* Update Agent for more details.

Update Agent can also automatically download specific packages, based on your preferences.

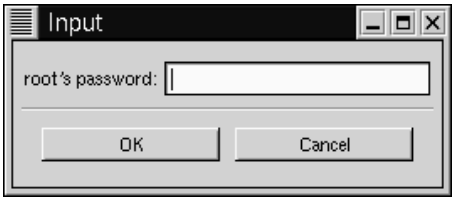
It isn't necessary to have a dedicated line in order to work with Update Agent; if you have dialup access to an ISP you can also benefit from the application's features.

5.1 Starting Update Agent

In most cases, Update Agent will be installed as a matter of course during a new installation or an upgrade.

To start the application in GNOME, select the **Main Menu Button** on the **Panel**, then go to **System** => **Update Agent.** If you're using KDE as your graphical environment, you can start Update Agent by going to the **Panel** => **Red Hat** => **System** => **Update Agent.**

Figure 5–1 Box Requesting Root Password



Please Note

Only root can use Update Agent. Each time you start the application from your user account, you will be asked to supply your root password, as Figure 5–1, *Box Requesting Root Password* shows. You can also start the Update Agent as root at a shell prompt by typing `/usr/bin/up2date`.

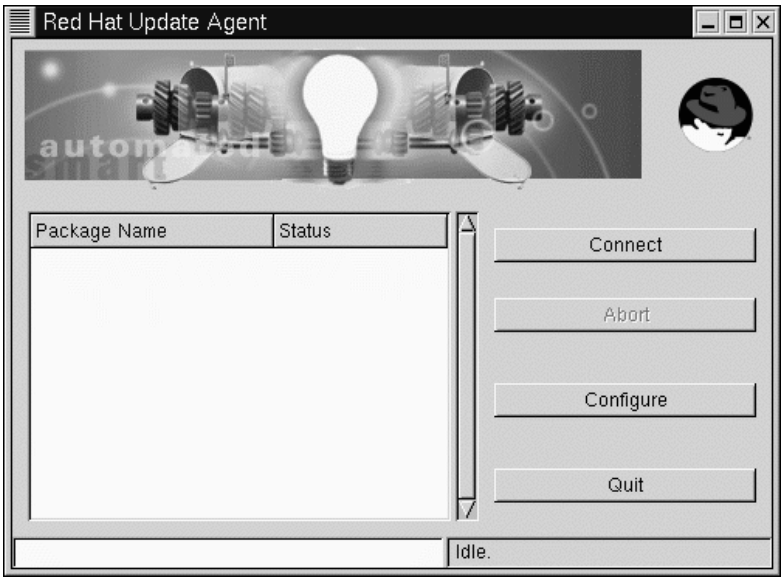
After supplying the password, the application will start. Once the application starts, you'll see a screen like Figure 5–2, *Main Update Agent Window*. From within this main dialog, you have the option of connecting to the priority update server, configuring or quitting the application.

Before continuing, however, you'll have to configure the application with information that's specific to your machine and software.

5.2 Configuring Update Agent

As shipped, Update Agent will not automatically connect, download and install updated packages. Before proceeding, you'll have to supply information about yourself, such as the user name and password which you selected when you registered Red Hat Linux 6.2. (To learn more about registration benefits, go to http://www.redhat.com/now.)

Figure 5–2 Main Update Agent Window



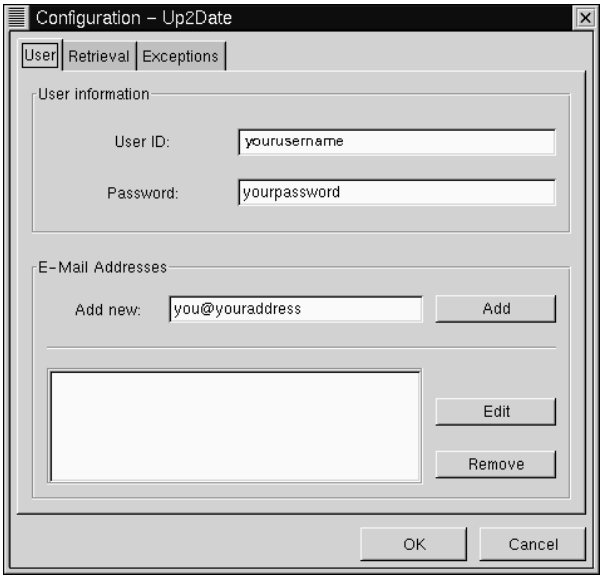
To configure the application, select **Configure** on the main dialog. (The button labeled **Abort** is only active during a download or installation process; selecting this button causes Update Agent to cease its current work.)

The **Configuration** dialog presents you with three tabbed windows in which you must enter information:

- **User** -- Here, you'll enter the user name and password which you had selected when you registered Red Hat Linux 6.2 at http://www.redhat.com/now. You should also enter an e-mail address to which correspondence regarding update transactions should be sent.
- **Retrieval** -- Specify the name of the server and certain installation instructions, such as which version of Red Hat Linux to update and whether to install as well as download packages.
- **Exceptions** -- Specify files and/or packages which you want to exclude from your update search, such as kernel-related files.

Let's begin with the **User** tab. When you're finished, your information should appear somewhat like Figure 5–3, *User Configuration Tab*.

Figure 5–3 User Configuration Tab



The **User** dialog is separated into two sections: **User Information** and **E-Mail Addresses**.

In the **User Information** section, enter the user name you specified when you registered in the box marked **User ID**. In the box for **Password**, enter the password you chose when you registered.

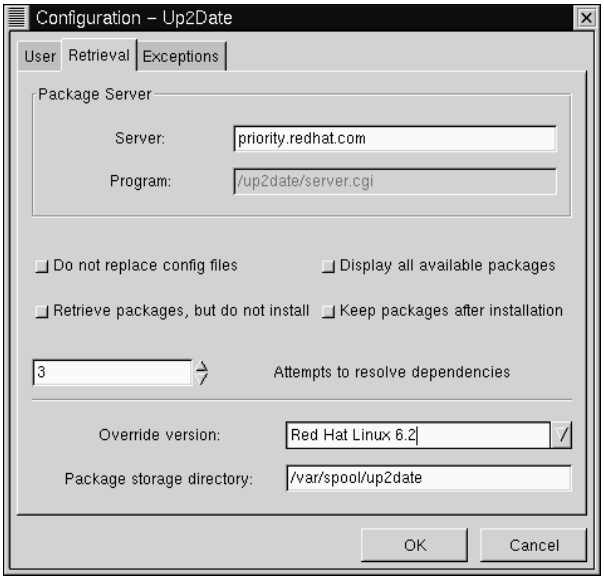
In the **E-Mail Addresses** section, you should specify the e-mail address to which feedback and summaries from Update Agent should be sent for review. You can have the feedback and summaries sent to numerous accounts, such as your user account. Just enter the addresses in the **Add new** box, then click on the **Add** button. The entry will appear in the address window.

Now it's time to specify your options in the **Retrieval** tab.

The **Retrieval** tab is divided into three sections, which detail the location of the priority updates server, specific download and installation instructions, and the location of

the directory where updates are stored on your system. (See Figure 5–4, *Retrieval Configuration Tab* for an example.)

Figure 5–4 Retrieval Configuration Tab



In the **Server** entry, you should enter `priority.redhat.com` as the priority updates server, if it isn't already present. The entry for **Program** is unavailable for you to specify; it simply indicates the location of the CGI script which executes Update Agent.

Tip

You can still use Update Agent if your registration expires to `priority.redhat.com`. Just point the application to `ftp.redhat.com`.

Similar to options you might find in Gnome-RPM, the selections which follow the server information allow you to customize how you want to retrieve and install updates. Selecting a button chooses that option. Your options are:

- **Do not replace config files** -- Packages which contain configuration files that are modified on your system will not be selected for updating. Let's say, for example, that you've modified the `/etc/syslog.conf` configuration file. If there is an update available for `sysklogd`, which would create a new `/etc/syslog.conf` file, this package will not be downloaded and installed. If you haven't modified the configuration file on your system, it will be available for downloading and installation -- and the previous file will be backed up (in this case, as `/etc/syslog.conf.rpmsave`).
- **Display all available packages** -- Select this box if you want Update Agent to show you *all* the available packages, not just the packages specific to your installed system.
- **Retrieve packages, but do not install** -- By default, this option is selected. When this option is selected, Update Agent only downloads packages into a temporary directory. Once downloaded, you will be able to install the packages yourself. If this option is unchecked, Update Agent will download and automatically install your selections. Uncheck this option, then, to both download and install updated packages; to install the packages yourself, refer to the location specified in the **Package storage directory** at the bottom of the location.
- **Keep packages after installation** -- Retains the package files after they've been installed. This could be useful if you want to keep copies of packages, however selecting this could potentially eat up quite a bit of disk space.
- **Attempts to resolve dependencies** -- The value associated with this option instructs Update Agent on the number of times to check for possible dependency problems with selected packages. The value ranges from 1 to 10, and can be adjusted by using the scroll buttons on the side of the box. The higher the value, the greater the number of dependency checks before your updated packages are installed.

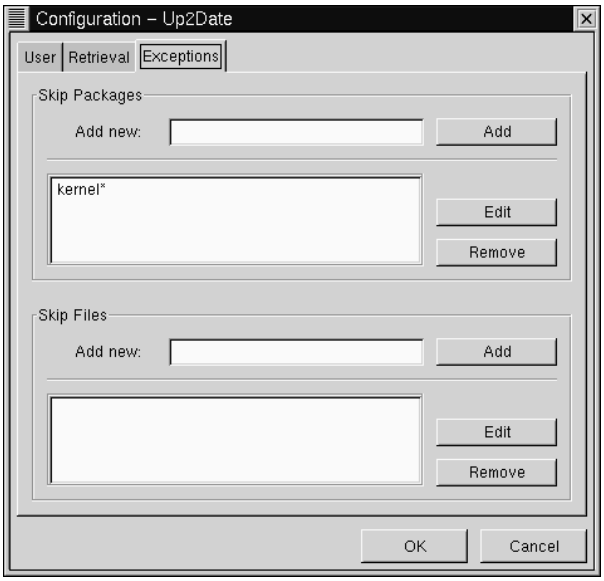
Beneath the package options, two additional boxes allow you to specify the version of Red Hat Linux files you want to update, and the location where those upgrade files will be kept.

The drop-down menu for **Override version** will show you a current or previous versions of Red Hat Linux. Most users will probably retain the default entry, which is Red Hat Linux 6.2. If no value is specified here, the proper value will automatically be determined.

The default entry for **Package storage directory** is set to `/var/spool/up2date` as the default, although you can specify a more preferable location. For example, if you've chosen to download packages without automatically installing them, you may find that these downloads take up quite a bit of space over time. Specifying a directory on your filesystem allows you to send these packages to a directory which you know can accommodate their size.

The **Exceptions** tab allows you to specify packages or individual files which you don't want retrieved from `priority.redhat.com` and offered as an update. (See Figure 5–5, *Update Exceptions Tab* for an example of this tab.)

Figure 5–5 Update Exceptions Tab



In this tab, you have the opportunity to specify which files and packages you would prefer not to be updated or modified. Both entries -- **Skip Packages** and **Skip Files** -- allow you to use **wildcards**, such as an asterisk (*) and a question mark (?).

The default entry in the **Skip Packages** section lists **kernel***, meaning that any packages which begin with `kernel` are excluded from download and installation. Most users will likely want to keep this exclusion.

To add new entries, type the name of the package you do not want selected (for example **`exmh*`** for the e-mail application exmh), then click on the **Add** button. Your entry will be added to the list.

Like the **Skip Packages** option, the **Skip Files** section allows you to specify individual files you do not want selected. For example, if you want to exclude all the files in `/home`, you would enter **`/home/*`** and the accounts in `/home` will be excluded from updates.

5.3 Using Update Agent

When you sign up for services at http://www.redhat.com/now, you can select an option to receive update notifications by e-mail. If you choose, you'll be told when updates are available, based on your preferences.

Once you receive an update notification and have correctly configured Update Agent, you can begin to use it to connect to `priority.redhat.com` for updates to download and install.

Please note

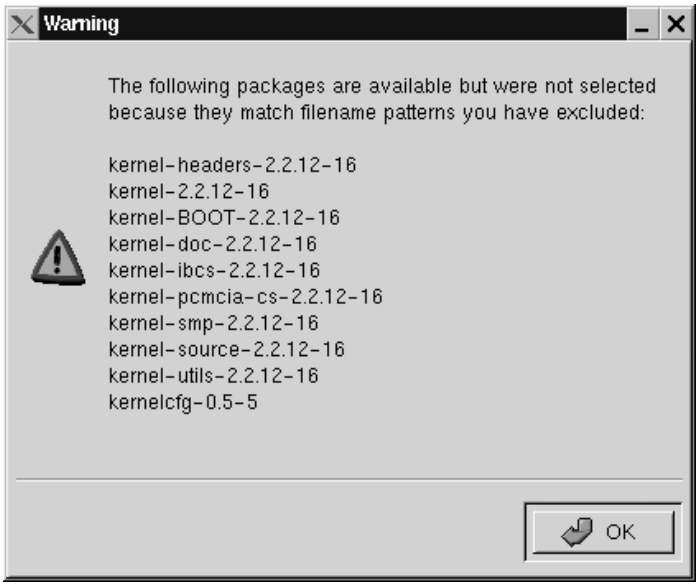
You must have either a dedicated connection, or be connected to your ISP through a dialup connection to use Update Agent.

To begin a session, start Update Agent (in **System** => **Update Agent**), and provide the root account's password.

From the main screen of the application, click on **Connect**. The status bar at the bottom of the window will indicate that the application is connecting and retrieving information from the updates FTP site.

If you have chosen to exclude packages or files (such as kernel updates), you may see a response similar to Figure 5–6, *Excluded Files*. Click on the **OK** button to continue.

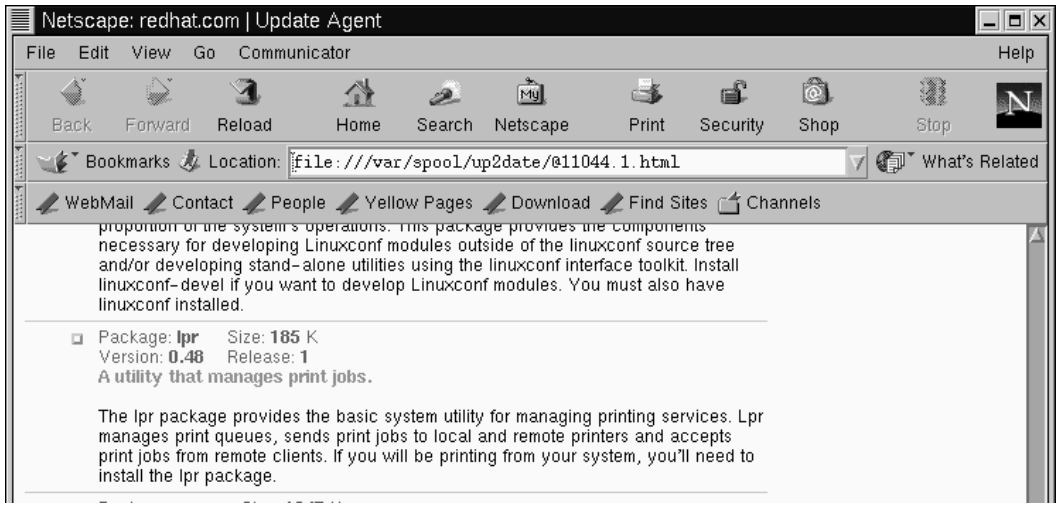
Figure 5–6 Excluded Files



The status bar on the main Update Agent window will indicate the the application is creating a page which will be displayed in your Web browser. Once it's finished sorting the packages and building the page, your browser will open and you'll see results similar to Figure 5–7, *Available Updates*.

If there are updated packages available, you'll see the name of the package, a summary of its function and related details, such as the latest version number, the release number and the size of the package.

Figure 5–7 Available Updates

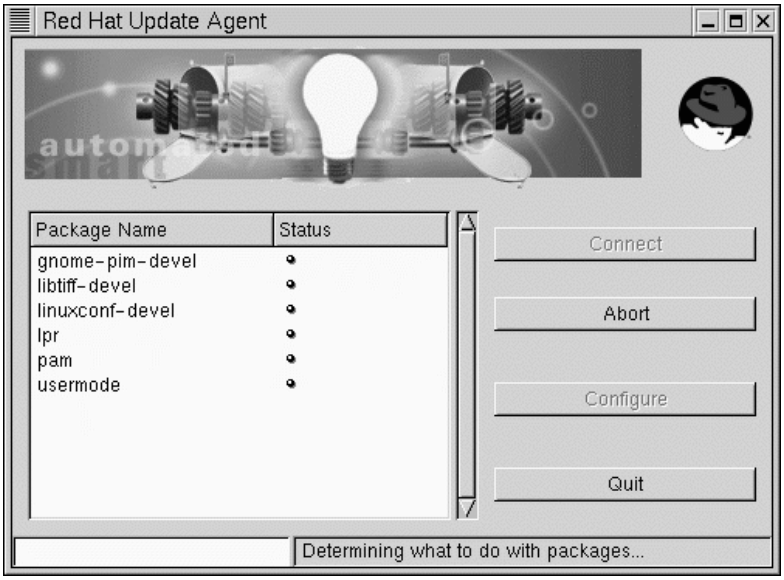


You can select individual packages by scrolling down the page and checking the box next to the package. When you're finished, click on the **Request selected packages** button at the bottom of the page.

If you'd prefer to update your system with all the available packages, click on the **Request ALL packages** button; you won't be required to make individual selections beside each package.

After you've made your request for the packages, the browser window will close. (You may receive a message in a window, asking you whether you want to close the window; confirm this decision by clicking on **OK**.)

Figure 5–8 Beginning the download process



Next, you'll see that the packages which are being downloaded will be shown in the main Update Agent screen, under **Package Name** and a small button will appear next to each package in the **Status** column, similar to Figure 5–8, *Beginning the download process*.

When packages have completed downloading, the button in the **Status** column will reflect this change by becoming a red arrow. The arrow will become a check mark when each package has been successfully installed. The progress bar on the lower right of the window will reflect the work which remains before all the updates have been downloaded and installed.

After all the packages have been retrieved and/or installed on your system, a window will open informing you of the successful procedure. You'll also receive e-mail verification that you'd performed the procedure; the verification will be sent to the address you'd specified as your **User ID**.

To quit Update Agent once all the packages have been retrieved and installed, click on the **Quit** button.

6 Package Management with RPM

The *R*PM *P*ackage *M*anager (RPM), is an open packaging system available for anyone to use, and works on Red Hat Linux as well as other Linux and UNIX systems. Red Hat, Inc. encourages other vendors to take the time to look at RPM and use it for their own products. RPM is distributable under the terms of the GPL.

For the end user, RPM provides many features that make maintaining a system far easier than it has ever been. Installing, uninstalling, and upgrading RPM packages are all one line commands, and all the messy details have been taken care of for you. RPM maintains a database of installed packages and their files, which allows you to perform powerful queries and verification of your system.

During upgrades, RPM handles configuration files specially, so that you never lose your customizations -- a feature that is impossible with straight `.tar.gz` files.

For the developer, RPM allows you to take source code for software and package it into source and binary packages for end users. This process is quite simple and is driven from a single file and optional patches that you create. This clear delineation of "pristine" sources and your patches and build instructions eases the maintenance of the package as new versions of the software are released.

Please Note

Although it can be important to understand the concepts behind RPM, for those who prefer a graphical interface to the command line, we suggest you use Gnome-RPM. Please see Chapter 7, *Gnome-RPM* for more information.

6.1 RPM Design Goals

Before trying to understand how to use RPM, it helps to have an idea of what the design goals are.

Upgradability

With RPM you can upgrade individual components of your system without completely reinstalling. When you get a new release of an operating system based on RPM (such as Red Hat Linux), you don't need to reinstall on your machine (as you do with operating systems based on other packaging systems). RPM allows intelligent, fully-automated, in-place upgrades of your system. Configuration files in packages are preserved across upgrades, so you won't lose your customizations.

Powerful Querying

RPM is also designed to have powerful querying options. You can do searches through your entire database for packages or just certain files. You can also easily find out what package a file belongs to and where it came from. The files an RPM package contains are in a compressed archive, with a custom binary header containing useful information about the package and its contents, allowing you to query individual packages quickly and easily.

System Verification

Another powerful feature is the ability to verify packages. If you are worried that you deleted an important file for some package, simply verify the package. You will be notified of any anomalies. At that point, you can reinstall the package if necessary. Any configuration files that you modified are preserved during reinstallation.

Pristine Sources

A crucial design goal was to allow the use of "pristine" software sources, as distributed by the original authors of the software. With RPM, you have the pristine sources along with any patches that were used, plus complete build instructions. This is a big advantage for several reasons. For instance, if a new version of a program comes out, you don't necessarily have to start from scratch to get it to compile. You can look at the patch to see what you *might* need to do. All the compiled-in defaults, and all of the changes that were made to get the software to build properly are easily visible this way.

This goal may only seem important for developers, but it results in higher quality software for end users too. We would like to thank the folks from the BOGUS distribution for originating the pristine source concept.

6.2 Using RPM

RPM has five basic modes of operation (not counting package building): installing, uninstalling, upgrading, querying, and verifying. This section contains an overview of each mode. For complete details and options try `rpm --help`, or turn to Section 6.4, *Other RPM Resources* for more information on RPM.

6.2.1 Installing

RPM packages typically have file names like `foo-1.0-1.i386.rpm`, which includes the package name (`foo`), version (`1.0`), release (`1`), and architecture (`i386`). Installing a package is as simple as:

```
# rpm -ivh foo-1.0-1.i386.rpm
foo                      ####################################
#
```

As you can see, RPM prints out the name of the package (which is not necessarily the same as the file name, which could have been `1.rpm`), and then prints a succession of hash marks as the package is installed, as a progress meter.

Installing packages is designed to be simple, but you can get a few errors:

Package Already Installed

If the package is already installed, you will see:

```
# rpm -ivh foo-1.0-1.i386.rpm
foo                     package foo-1.0-1 is already installed
error: foo-1.0-1.i386.rpm cannot be installed
#
```

If you really want to install the package anyway, you can use `--replacepkgs` on the command line, which tells RPM to ignore the error:

```
# rpm -ivh --replacepkgs foo-1.0-1.i386.rpm
foo                      ####################################
```

```
#
```

Conflicting Files

If you attempt to install a package that contains a file which has already been installed by another package, you'll see:

```
# rpm -ivh foo-1.0-1.i386.rpm
foo           /usr/bin/foo conflicts with file from bar-1.0-1
error: foo-1.0-1.i386.rpm cannot be installed
#
```

To cause RPM to ignore that error, use `--replacefiles` on the command line:

```
# rpm -ivh --replacefiles foo-1.0-1.i386.rpm
foo                     ####################################
#
```

Unresolved Dependency

RPM packages can "depend" on other packages, which means that they require other packages to be installed in order to run properly. If you try to install a package for which there is such an unresolved dependency, you'll see:

```
# rpm -ivh bar-1.0-1.i386.rpm
failed dependencies:
        foo is needed by bar-1.0-1
#
```

To handle this error you should install the requested packages. If you want to force the installation anyway (a bad idea since the package probably will not run correctly), use `--nodeps` on the command line.

6.2.2 Uninstalling

Uninstalling a package is just as simple as installing:

```
# rpm -e foo
#
```

Notice that we used the package *name* "`foo`," not the name of the original package *file* "`foo-1.0-1.i386.rpm`".

You can encounter a dependency error when uninstalling a package if some other installed package depends on the one you are trying to remove. For example:

```
# rpm -e foo
removing these packages would break dependencies:
        foo is needed by bar-1.0-1
#
```

To cause RPM to ignore that error and uninstall the package anyway (which is also a bad idea since the package that depends on it will probably fail to work properly), use `--nodeps` on the command line.

6.2.3 Upgrading

Upgrading a package is similar to installing.

```
# rpm -Uvh foo-2.0-1.i386.rpm
foo                      ####################################
#
```

What you don't see above is that RPM automatically uninstalled any old versions of the `foo` package. In fact you may want to always use `-U` to install packages, since it works fine even when there are no previous versions of the package installed.

Since RPM performs intelligent upgrading of packages with configuration files, you may see a message like:

```
saving /etc/foo.conf as /etc/foo.conf.rpmsave
```

This means that your changes to the configuration file may not be "forward compatible" with the new configuration file in the package, so RPM saved your original file, and installed a new one. You should investigate and resolve the differences between the two files as soon as possible to ensure that your system continues to function properly.

Since upgrading is really a combination of uninstalling and installing, you can encounter any errors from those modes, plus one more: If RPM thinks you are trying to upgrade to a package with an *older* version number, you will see:

```
# rpm -Uvh foo-1.0-1.i386.rpm
foo    package foo-2.0-1 (which is newer) is already installed
error: foo-1.0-1.i386.rpm cannot be installed
#
```

To cause RPM to "upgrade" anyway, use `--oldpackage` on the command line:

```
# rpm -Uvh --oldpackage foo-1.0-1.i386.rpm
foo                      ####################################
#
```

6.2.4 Freshening

Freshening a package is similar to upgrading:

```
# rpm -Fvh foo-1.2-1.i386.rpm
foo                      ####################################
#
```

RPM's freshen option checks the versions of the packages specified on the command line against the versions of packages that have already been installed on your system. When a newer version of an already-installed package is processed by RPM's freshen option, it will be upgraded to the newer version. However, RPM's freshen option will not install a package if no previously-installed package of the same name exists. This differs from RPM's upgrade option, as an upgrade *will* install packages, whether or not an older version of the package was already installed.

RPM's freshen option works well with single packages or with a group of packages. It's especially handy if you've just downloaded a large number of different packages, and you only want to upgrade those packages that are already installed on your system. Using the freshen option means that you won't have to pick through the downloaded packages, deleting any unwanted ones before using RPM.

In this case, you can simply issue the following command:

```
# rpm -Fvh *.rpm
```

RPM will automatically upgrade only those packages that have already been installed.

6.2.5 Querying

Querying the database of installed packages is accomplished with `rpm -q`. A simple use is `rpm -q foo` which will print the package name, version, and release number of the installed package foo:

```
# rpm -q foo
foo-2.0-1
#
```

Instead of specifying the package name, you can use the following options with `-q` to specify the package(s) you want to query. These are called *Package Specification Options*.

- `-a` queries all currently installed packages.
- `-f` *`<file>`* will query the package owning *`<file>`*.
- `-p` *`<packagefile>`* queries the package *`<packagefile>`*.

There are a number of ways to specify what information to display about queried packages. The following options are used to select the type of information for which you are searching. These are called *Information Selection Options*.

- `-i` displays package information such as name, description, release, size, build date, install date, vendor, and other miscellaneous information.
- `-l` displays the list of files that the package "owns".
- `-s` displays the state of all the files in the package.
- `-d` displays a list of files marked as documentation (man pages, info pages, README's, etc).
- `-c` displays a list of files marked as configuration files. These are the files you change after installation to adapt the package to your system (`sendmail.cf`, `passwd`, `inittab`, etc).

For those options that display file lists, you can add `-v` to your command line to get the lists in a familiar `ls -l` format.

6.2.6 Verifying

Verifying a package compares information about files installed from a package with the same information from the original package. Among other things, verifying compares the size, MD5 sum, permissions, type, owner and group of each file.

The command `rpm -V` verifies a package. You can use any of the *Package Selection Options* listed for querying to specify the packages you wish to verify. A simple use is `rpm -V foo` which verifies that all the files in the foo package are as they were when they were originally installed. For example:

- To verify a package containing particular file:

  ```
  rpm -Vf /bin/vi
  ```

- To verify ALL installed packages:

  ```
  rpm -Va
  ```

- To verify an installed package against an RPM package file:

  ```
  rpm -Vp foo-1.0-1.i386.rpm
  ```

  This can be useful if you suspect that your RPM databases are corrupt.

If everything verified properly there will be no output. If there are any discrepancies they will be displayed. The format of the output is a string of 8 characters, a possible "c" denoting a configuration file, and then the file name. Each of the 8 characters denotes the result of a comparison of one attribute of the file to the value of that attribute recorded in the RPM database. A single "." (period) means the test passed. The following characters denote failure of certain tests:

- 5 -- MD5 checksum

- S -- File size
- L -- Symbolic link
- T -- File modification time
- D -- Device
- U -- User
- G -- Group
- M -- Mode (includes permissions and file type)

If you see any output, use your best judgment to determine if you should remove or reinstall the package, or otherwise fix the problem.

6.3 Impressing Your Friends with RPM

RPM is a very useful tool for both managing your system and diagnosing and fixing problems. The best way to make sense of all the options is to look at some examples.

- Let's say you delete some files by accident, but you aren't sure what you deleted. If you want to verify your entire system and see what might be missing, you would enter:

  ```
  rpm -Va
  ```

  If some files are missing, or appear to have been corrupted, you should probably either re-install the package or uninstall, then re-install the package.

- Let's say you run across a file that you don't recognize. To find out which package owns it, you would enter:

  ```
  rpm -qf /usr/X11R6/bin/xjewel
  ```

  The output would look like:

  ```
  xjewel-1.6-1
  ```

- We can combine the above two examples in the following scenario. Say you are having problems with `/usr/bin/paste`. You would like to verify the package that owns that program but you don't know which package that is. Simply enter:

  ```
  rpm -Vf /usr/bin/paste
  ```

  and the appropriate package will be verified.

- Want to find more about a particular program? You can find out by entering the following to locate the documentation which came with the package that "owns" that program (in this case ispell):

  ```
  rpm -qdf /usr/bin/ispell
  ```

  The output would be:

  ```
  /usr/man/man4/ispell.4
  /usr/man/man4/english.4
  /usr/man/man1/unsq.1
  /usr/man/man1/tryaffix.1
  /usr/man/man1/sq.1
  /usr/man/man1/munchlist.1
  /usr/man/man1/ispell.1
  /usr/man/man1/findaffix.1
  /usr/man/man1/buildhash.1
  /usr/info/ispell.info.gz
  /usr/doc/ispell-3.1.18-1/README
  ```

- Let's say you find a new koules RPM, but you don't know what it is. To find out some information on it, enter:

  ```
  rpm -qip koules-1.2-2.i386.rpm
  ```

  The output would be:

  ```
  Name        : koules Distribution: Red Hat Linux Colgate
  Version     : 1.2          Vendor: Red Hat Software
  Release     : 2        Build Date: Mon Sep 02 11:59:12 1996
  Install date: (none)   Build Host: porky.redhat.com
  Group       : Games    Source RPM: koules-1.2-2.src.rpm
  Size        : 614939
  Summary     : SVGAlib action game; multiplayer, network
  ```

```
Description :
This arcade-style game is novel in conception and
excellent in execution.  No shooting, no blood, no guts,
no gore.  The play is simple, but you still must develop
skill to play.  This version uses SVGAlib to run on a
graphics console.
```

- Now you want to see what files the koules RPM installs. You would enter:

```
rpm -qlp koules-1.2-2.i386.rpm
```

The output is:

```
/usr/man/man6/koules.6
/usr/lib/games/kouleslib/start.raw
/usr/lib/games/kouleslib/end.raw
/usr/lib/games/kouleslib/destroy2.raw
/usr/lib/games/kouleslib/destroy1.raw
/usr/lib/games/kouleslib/creator2.raw
/usr/lib/games/kouleslib/creator1.raw
/usr/lib/games/kouleslib/colize.raw
/usr/lib/games/kouleslib
/usr/games/koules
```

These are just several examples. As you use the system you will find many more uses for RPM.

6.4 Other RPM Resources

For more information on RPM, check out the man page, the help screen (`rpm --help`), and the RPM documents available at

http://www.rpm.org/

There is also an RPM book available. It's called *Maximum RPM*, and it is available from Red Hat, at your local bookstore and through online booksellers. It contains a wealth of information about RPM for both the end-user and the package builder.

An on-line version of the book is available at http://www.rpm.org/.

There is also a mailing list for discussion of RPM-related issues, called `rpm-list@redhat.com`.

The list is archived on http://www.redhat.com/support/mailing-lists/. To subscribe, send mail to `rpm-list-request@redhat.com` with the word `subscribe` in the subject line.

7 Gnome-RPM

One of the most convenient package manipulation tools available is Gnome-RPM, a graphical tool which runs under the X Window System. Gnome-RPM was written by James Henstridge (`james@daa.com.au`); RPM 3.0 support was written by Red Hat and additional rpmfind code was written by Daniel Veillard.

Gnome-RPM (which is also referred to as gnorpm) allows the end-user to easily work with RPM technology; it is fast, powerful and features a friendly interface.

To learn more about RPM technology, turn to Chapter 6, *Package Management with RPM*.

Gnome-RPM is "GNOME-compliant," meaning that it seamlessly integrates into GNOME, the X Window System desktop environment.

With Gnome-RPM, you can easily

- install RPM packages
- uninstall RPM packages
- upgrade RPM packages
- find new RPM packages
- query RPM packages
- verify RPM packages

The interface features a menu, a toolbar, a tree and a display window of currently installed packages.

Operations are often performed in Gnome-RPM by finding and selecting packages, then choosing the type of operation to perform via push-button on the toolbar, through the menu or by right-clicking with the mouse.

- Installing a package places all of the components of that package on your system in the correct locations.
- Uninstalling a package removes all traces of the package except for configuration files you have modified.

- Upgrading a package installs the newly available version and uninstalls all other versions that were previously installed. This allows quick upgrading to the latest releases of packages. Refer to Section 7.4, *Configuration* for information about how to alter the default settings for installing and uninstalling packages.

You can also use the **Web find** option to search the Internet for newly released packages. You can direct Gnome-RPM to search for particular distributions when you want to look for new packages. (If you have a slow connection, this option can take some time to fully execute.) See Section 7.4, *Configuration* for more information about this feature.

Please Note

Exercise caution if you choose to use the **Web find** option, since there is no way to verify the integrity of the many packages which are available at numerous repositories. Before installing packages, you should perform a query on that package to help you determine whether it can be trusted. Packages not produced by Red Hat are not supported in any way by Red Hat.

Using Gnome-RPM to perform all of these and many other operations is the same as using `rpm` from the shell prompt. However, the graphical nature of Gnome-RPM often makes these operations easier to perform.

The usual way to work with Gnome-RPM is to display the available packages, select the package(s) you want to operate on, and then select an option from the toolbar or menu which performs the operation. However, Gnome-RPM is flexible enough to display packages in a variety of views, thanks to the use of **filters.** Refer to Section 7.3, *Installing New Packages* for more information on using filters to identify packages.

You can install, upgrade or uninstall several packages with a few button clicks. Similarly, you can query and verify more than one package at a time. Because of Gnome-RPM's integration with GNOME, you can also perform installation, query and verification on packages from within the GNOME File Manager.

7.1 Starting Gnome-RPM

You can start Gnome-RPM from either an Xterm window or from the GNOME desktop Panel (**Main Menu Button** => **System** => **GnoRPM**).

To start Gnome-RPM from an Xterm window, at the shell prompt, simply type

```
gnorpm &
```

That will bring up the main Gnome-RPM window (as shown in Figure 7–1, *Main Gnome-RPM Window*).

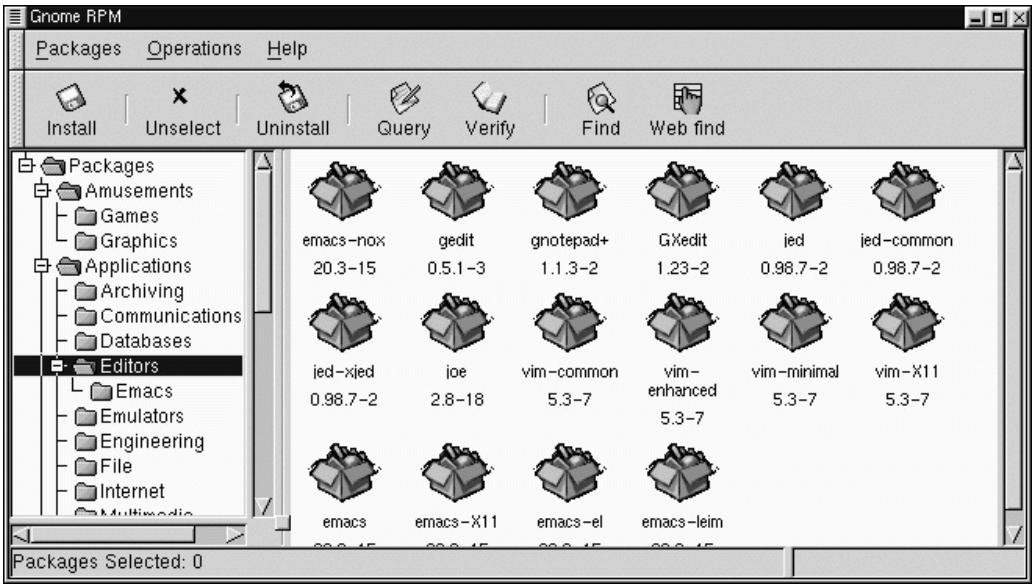
Please Note

If you would like to install, upgrade or uninstall packages, you must be in root. The easiest way to do this is to type `su` to become root, and then type the root password at a shell prompt. However, it isn't necessary to be root in order to query and verify packages.

There are several parts to the Gnome-RPM interface.

- Package Panel - On the left; allows you to browse and select packages on your system.
- Display window - To the right of the package panel; shows you contents from folders in the panel.
- Toolbar - Above the display and panel; a graphical display of package tools.
- Menu - Above the toolbar; contains text-based commands, as well as help info, preferences and other settings.
- Status bar - Beneath the panel and display windows; shows the total number of selected packages.

Figure 7–1 Main Gnome-RPM Window



7.2 The Package Display

Each folder icon in the tree view at left represents a group of packages. Each group can contain subgroups. Groups are used to place packages that perform similar functions in similar locations. For example, the folder **Editors** contains text editors such as ed, vim and GXedit. From the tree view on the left, you might find another folder beneath **Editors** called **Emacs**, which would contain both emacs and emacs-X11.

The tree view is also arranged in an expandable and collapsible manner, which helps you to easily navigate through the packages. A folder which appears with a **+** next to it indicates that there are subfolders within that category.

To view the packages and subgroups within a group, click once on a folder or a **+** with your left mouse button. The display window will then show you the contents of that folder. By default, you will be presented with icons to represent the packages. You can change that view to a list view by selecting **View as list** from the **Interface** tab

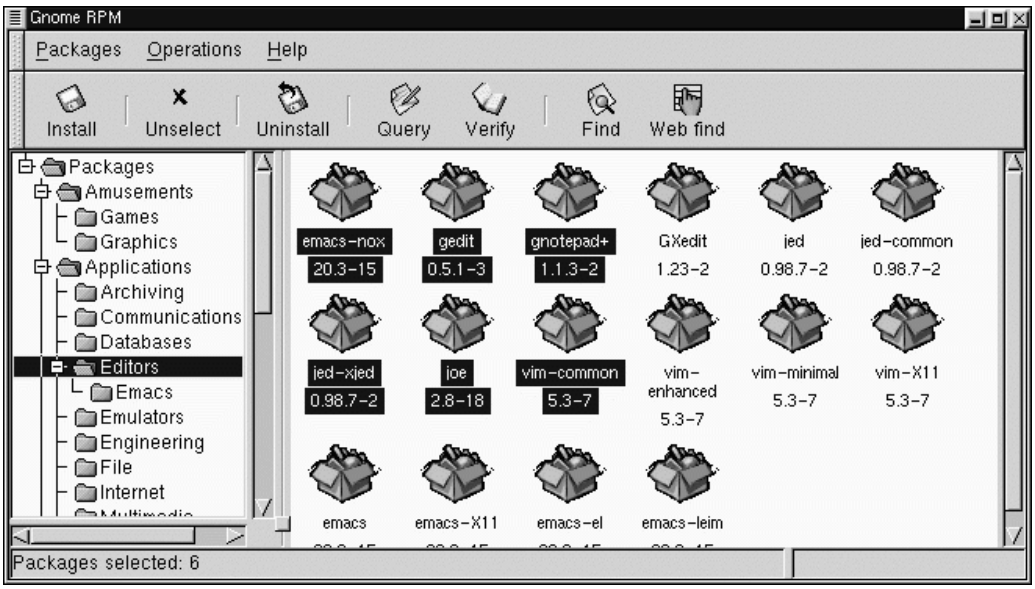
you'll find under **Operations** => **Preferences**. Refer to Section 7.4, *Configuration* for more information about customizing the settings.

In this manner, you can move about the tree view, opening and expanding folders containing applications, games, tools and more. The contents of each folder will be displayed at the right.

7.2.1 Selecting Packages

To select a single package, click on it with the left mouse button. You'll notice that highlighting will appear around the package title (as shown in Figure 7–2, *Selecting Packages in Gnome-RPM*) which reflects the selection. To unselect it, either click on an empty space in the display panel with the left mouse button, or click on the **Unselect** button on the toolbar. When you unselect a package, the highlighting will disappear.

Figure 7–2 Selecting Packages in Gnome-RPM



You can select and unselect multiple packages, in more than one folder in the tree panel. To select more than one package incrementally, left-click with your mouse

button, while holding down the [Ctrl] key; you'll see highlighting around each additional selection.

To select more than one package **globally**, that is, make larger selections within a folder, left-click one package, then, while holding down the [Shift] key, left-click on the final package you wish to select. By doing so, you'll notice that individual packages between your starting and ending selections will also be highlighted for selection. Using this option makes selecting groups of packages quicker than selecting each package individually.

The status bar at the bottom of Gnome-RPM will display the total number of packages you have selected.

7.3 Installing New Packages

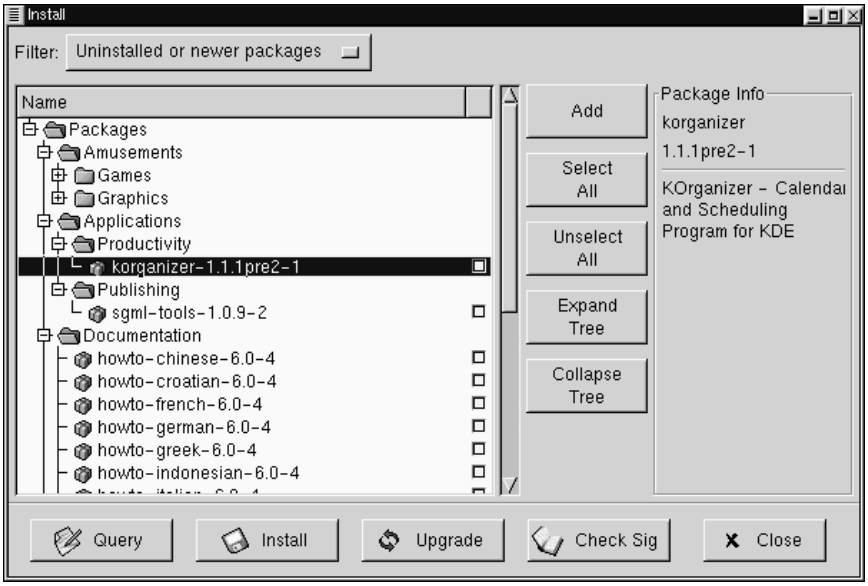
To install new packages, choose **Install** from the toolbar. The **Install** window will open, revealing packages which are either already installed, newer packages or other options, which can be selected from the **Filter** drop-down bar at the top of the window. See Figure 7–4, *The Add Packages Window* for an example of the **Install** window, using the filter for "**All but installed packages.**"

Using the **Filter** feature allows you to winnow your choices for viewing packages. Available filters for viewing include:

- All packages
- All but installed packages
- Only uninstalled packages
- Only newer packages
- Uninstalled or newer packages

You can switch the display of packages by using the drop-down bar at the top of the window.

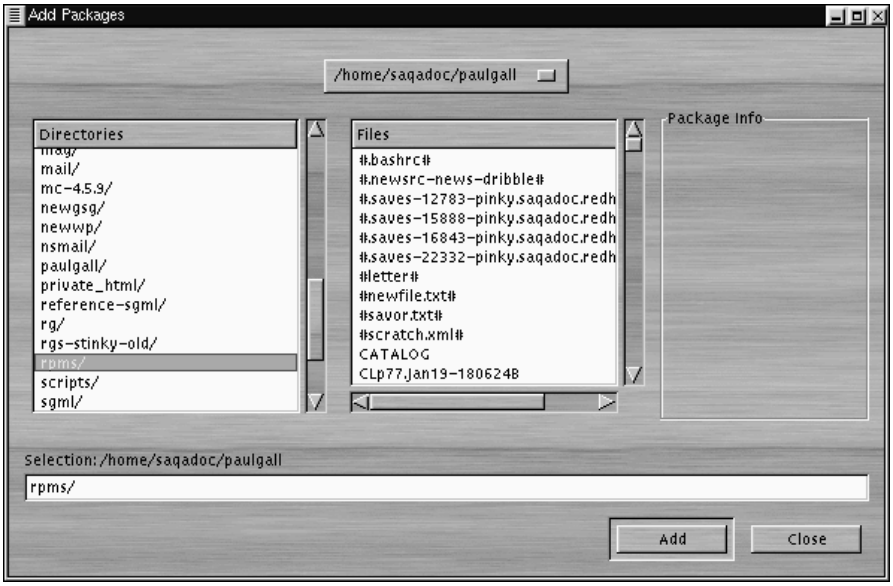
Figure 7–3 The Install Window



Choose the **Add** button. By default, if your CD-ROM is mounted with a Red Hat Linux CD-ROM, Gnome-RPM will search in `/mnt/cdrom/RedHat/RPMS` for new packages. (You can find this default option in the **Install Window** tab of the **Preferences** dialog. See Section 7.4, *Configuration* for more information on this feature.)

If no packages are available in the default path, you'll be presented with an **Add Packages** window from which you can select the appropriate location of your new package. In this view, you can select the correct path by using the drop-down bar at the top of the window to quickly navigate to pre-set locations; or you can double-click in the left panel of the **Add Packages** window to navigate to the correct path (as in Figure 7–4, *The Add Packages Window*). You can also type the path in the text window at the bottom of the **Add Packages** window.

Figure 7–4 The Add Packages Window



Please Note

If you download RPMs, you'll likely find those downloaded packages in a directory called `rpms` within your home directory. For example, if your home directory is `/home/bill`, your downloaded RPMs will be in `/home/bill/rpms`.

By clicking on the item, you'll find a brief description of the package in the **Package Info** panel of the **Install** window. To perform an installation or a query on the package, click inside the **checkbox** next to the package, then select the **Install** button. You can also query the item once it's checked; from within the **Package Info** window, you can also perform the installation (see Figure 7–10, *Query Window* in Section 7.5.1, *Querying Packages* for an example).

To choose an item, double-click on it with your left mouse button, or click on the **Add** button. The selected package(s) will be added to the **Install** window. You can also install more than one package in the same manner; each selection will be added to the **Install** window.

In addition to choosing to install the packages from within the **Install** window, you can install after performing a query on the selected package. Click on **Query**, which will open the **Package Info** window. Here, you can find a variety of details about the file(s) you've selected to install. Information will include the origination of the package, the date it was built, its size and more.

Within this **Package Info** window, you have the option of installing or upgrading packages.

If the package already exists on your system and you're querying a newer release, the **Package Info** window provides an **Upgrade** button, which will perform an upgrade to newer releases.

You can also "drag and drop" packages from GNOME File Manager. Within the File Manager, left-click on your selected RPM file then, while still holding down the mouse button, "drag" the file to the **Install** window and place it within the **Name** panel.

When dragging files to the **Install** window from the File Manager, you'll notice that the file appears as an icon while it's being dragged toward Gnome-RPM. Once inside the **Name** panel, you'll see that the package is checked for installation by default, and its information appears in the **Package Info** panel to the right.

To install the package now, just select the **Install** button.

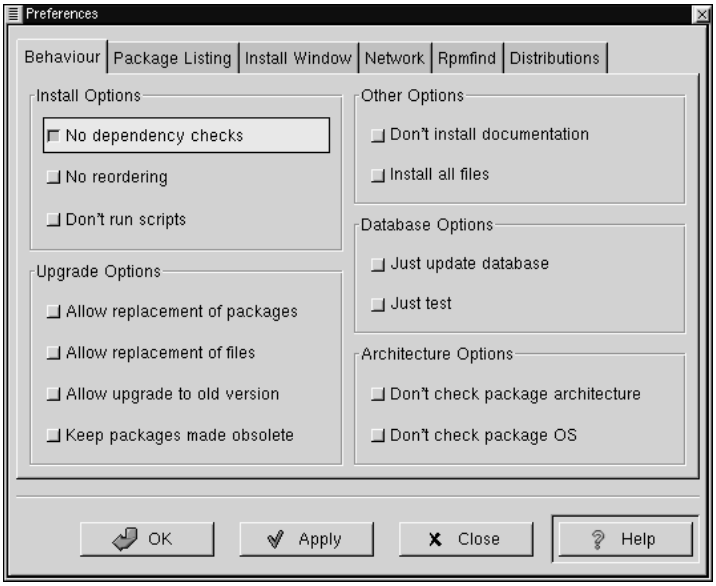
You'll see a progress indicator when your package is being installed.

7.4 Configuration

Gnome-RPM offers a wide selection of choices for installing and uninstalling packages, documentation and other features. You can customize Gnome-RPM through the **Preferences** dialog, which you can access from **Operations** => **Preferences** on the menu. To make selections in the many of the **Preferences** dialogs, select the check boxes next to the options.

Under the **Behavior** tab, you'll find a number of options for configuring the way Gnome-RPM installs, uninstalls and upgrades packages. The Behavior tab is split into five sections: Install, Upgrade, Other, Database and Architecture. Note that by default these boxes are not checked (see Figure 7–5, *Behavior Tab in Preferences*).

Figure 7–5 Behavior Tab in Preferences



Under **Install Options**, you have the following choices:

- **No dependency checks** - When selected, this will install or upgrade a package without checking for other types of files on which the program may be dependent in order to work. However, unless you know what you're doing we strongly suggest you not use this option as some packages may depend on other packages for files, libraries or programs to function correctly.
- **No reordering** - This option is useful if RPM is unable to change the installation order of some packages to satisfy dependencies.

- **Don't run scripts** - Pre- and post-install scripts are sequences of commands that are sometimes included in packages to assist with installation. This check box is similar to the `--noscripts` option when installing from the shell prompt.

Under **Upgrade Options**, you can select the following:

- **Allow replacement of packages** - Replaces a package with a new copy of itself. Similar to the `--replacepkgs` option from the shell prompt. This option can be useful if an already-installed package has become damaged or may require other repair to function correctly.
- **Allow replacement of files** - Allows the replacement of files which are owned by another package. The shell prompt equivalent for this RPM option is `--replacefiles`. This option can sometimes be useful when there are two packages with the same file name but different contents.
- **Allow upgrade to old version** - Like the shell prompt RPM option equivalent `--oldpackage`, this option allows you to "upgrade" to an earlier package. It can sometimes be useful if the latest version of a package doesn't function correctly for your system.
- **Keep packages made obsolete** - Prevents packages listed in an Obsoletes header from being removed.

In **Other Options**, you can select:

- **Don't install documentation** - Like `--excludedocs`, this option can save on disk space by excluding documentation such as man pages or other information related to the package.
- **Install all files** - Installs all files in the package.

The choices available in **Database Options** and **Architecture Options** allow you to decide, among other things, whether you want to perform a "test" installation (which will check for file conflicts without actually performing an install), or whether you want to exclude packages for other operating systems or system architectures.

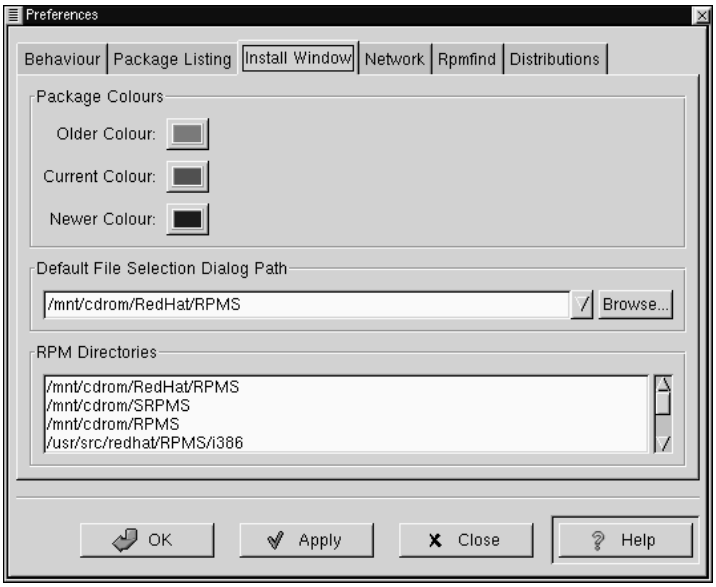
In the **Package Listing** tab, you'll find a choice of displays for your packages: either **View as icons**, which will be graphically-based, or **View as list**, which is not graphical but can provide more information about the packages.

In **Install Window,**, you can specify the path through which Gnome-RPM can find new RPMs on your system. Refer to Figure 7–6, *Install Window* for an example of this dialog. If you're using your Red Hat Linux CD-ROM, this path will probably be

```
/mnt/cdrom/RedHat/RPMS
```

or a similar path which had been set as the default path for Gnome-RPM. If you download new RPMs from the Internet or want to install RPMs via a NFS-mounted CD-ROM this path will be different for you.

Figure 7–6 Install Window



To change this path, type the full path to the RPMs you'd like to work with. Choosing the **Apply** or **OK** buttons will save this path, making it the default path for future sessions. You can also determine the default path by selecting the **Browse...** button, and visually navigating through the **RPMPath** window.

After changing install path and closing the dialog box, you can use the **Install** button to view the packages available in your new location.

(If the path for your RPMs doesn't match the default path in your preferences, you'll be presented with a browser window, which will allow you to select the correct path for your new RPMs.)

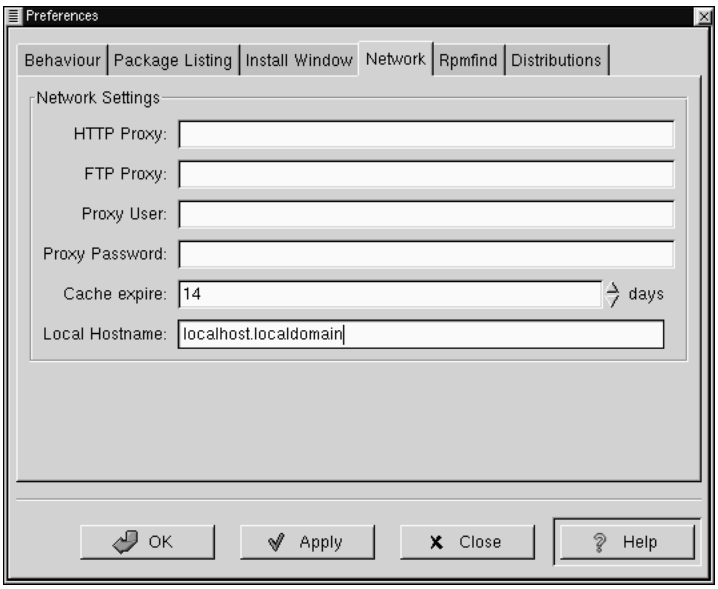
Under **Package Colors**, you'll find color coding for packages. The default setting for older packages is gray; for current packages, the color is green; for newer packages than those installed, the color is blue. These color values can be customized to suit your needs.

The **RPM Directories**, contains a listing of default locations where Gnome-RPM will search for packages.

In **Network**, you have the ability to specify proxies for use with HTTP and FTP transfers, as well as user and password names (see Figure 7–7, *Network Settings*). Note, however, that the password will not be stored securely.

In the **Cache expire** window, you can set the length of time before data from the rpmfind database is considered out of date.

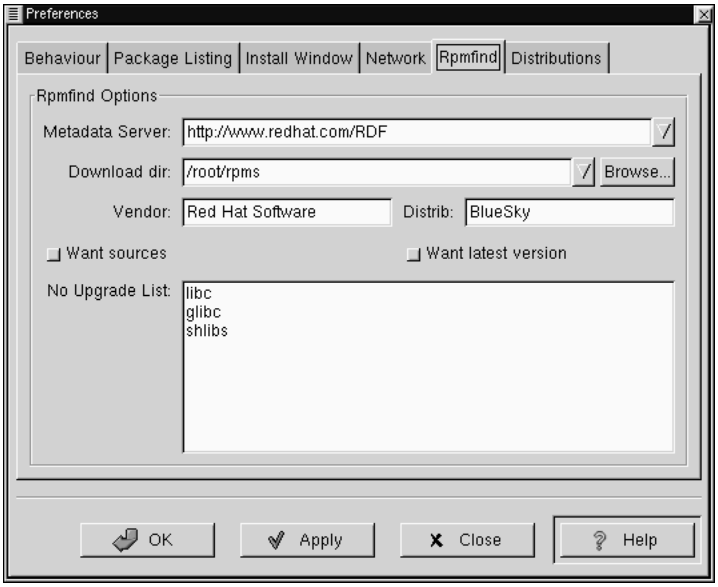
Figure 7–7 Network Settings



In **Rpmfind** and **Distributions**, you'll find settings and options which correspond to the **Web find** feature.

The Rpmfind system was devised by Daniel Veillard, and allows the user to search the Internet for packages by name, summary, architecture and more (see Figure 7–8, *The Rpmfind Window*). The user is then given the option of downloading and installing the most appropriate packages for their system. To learn more about Rpmfind, go to http://rpmfind.net/.

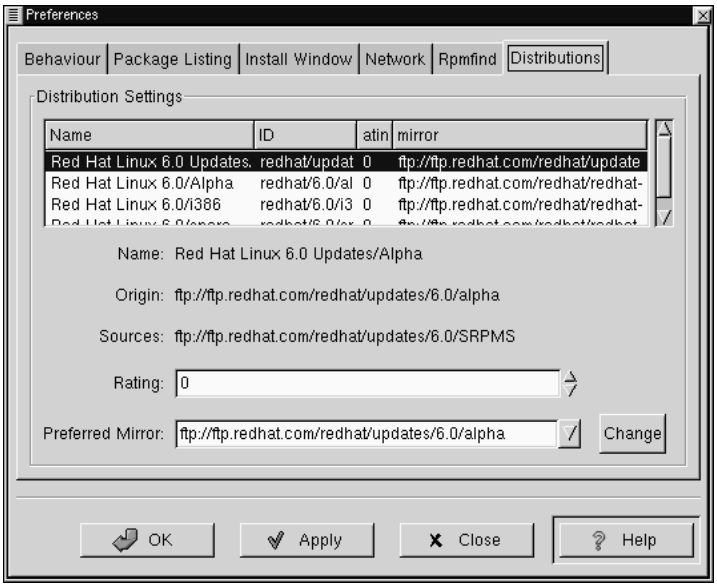
Figure 7–8 The Rpmfind Window



The **Metadata server** sets the server to be used for searches. The **Download dir:** entry allows you to specify where you want the files to be placed.

You can also specify the vendor, distribution name and whether to find sources and/or the latest files.

Figure 7–9 Distribution Settings in Preferences



In **Distribution Settings**, you can set the options for choosing the most appropriate package out of the selections Rpmfind returns, as well as which mirror you would like to use. The higher the rating you indicate for your selection (as shown in Figure 7–9, *Distribution Settings in Preferences*) the higher the priority it will receive; the lower rating, such as "-1," will specify that packages not be recommended.

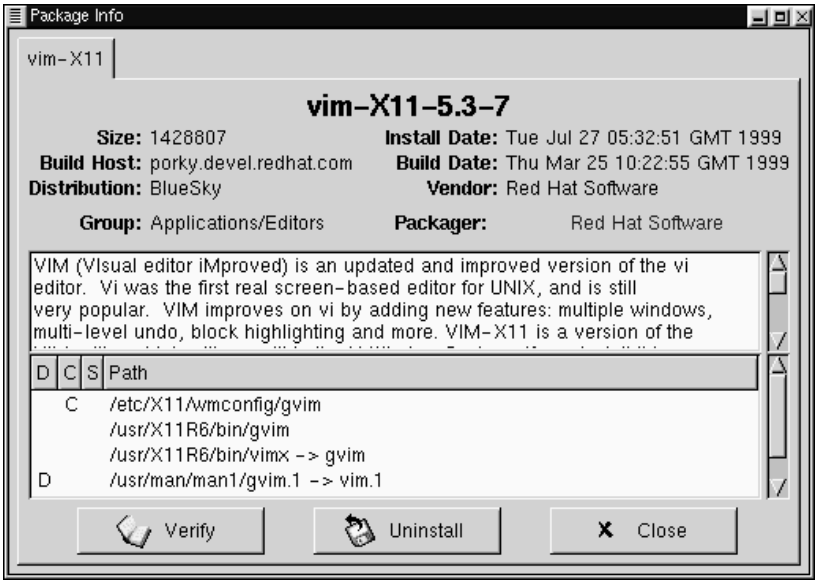
7.5 Package Manipulation

7.5.1 Querying Packages

The easiest way to query packages is to use the **Query** option from the menu at the top. If you want to query more than one package, make all your selections then press the **Query** button on the menu.

You'll be presented with a window like the one shown in Figure 7–10, *Query Window*. The more packages you've queried, the more "tabs" you'll find within the **Query** box, each tab representing a **Query** window for a package.

Figure 7–10 Query Window



The name of the package is centered at the top of the box. Below, the box is divided into two columns of listed information; below this information, you'll see a display area showing package files.

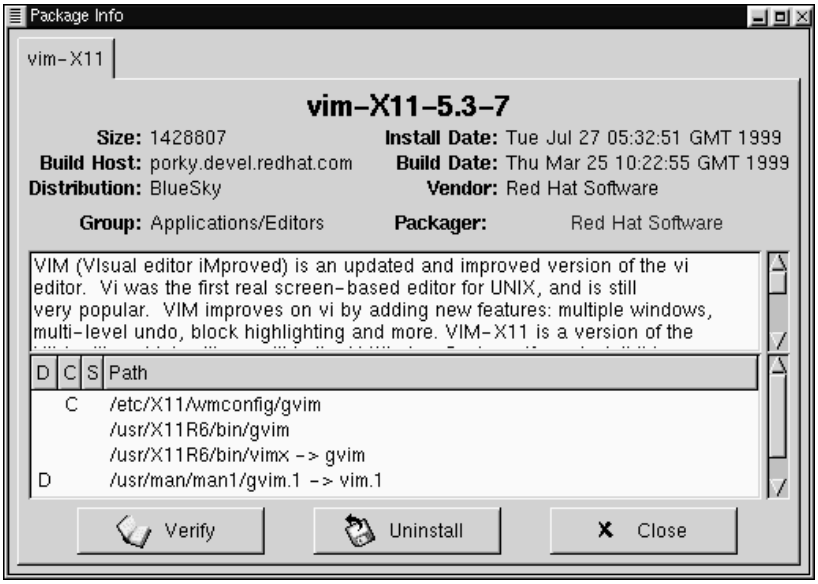
In the left column in the information list, you'll find the size of the file, the machine on which the file is found, the name of the package distribution and the group to which its function belongs.

In the right column, you'll find the date of the package's installation on your machine, the date the package was built, the name of the vendor and the name of the group who packaged the software. If the package has not been installed on your machine, that space will simply read, "not installed." Clicking on the name following **Packager** will cause your browser's e-mail application to open in compse, so that you can write to the packager.

Centered at the bottom of the information list is the URL of the application's developer (see Figure 7–11, *URL in the Query Window*). Similar to the e-mail function of

the **Packager** entry, clicking on the URL will cause your browser to open to the corresponding website.

Figure 7–11 URL in the Query Window



Below the description is a list of the files contained in the package. If a D appears in its related column to the left of the path, that file is a documentation file and would be a good thing to read for help on using the application. If a C appears in its respective column, the file is a configuration file. Under the S column, you can view the "state" of the package; here, you'll receive information if any files are reported as "missing" from the package (and therefore probably mean there's a problem with the package).

If you're querying a package that's already installed, you'll also find two additional buttons beneath at the bottom of this window: **Verify** and **Uninstall**. If you're performing a query on a package that hasn't been installed yet, the buttons on the bottom will be labeled **Install**, **Upgrade** and **Check Sig**.

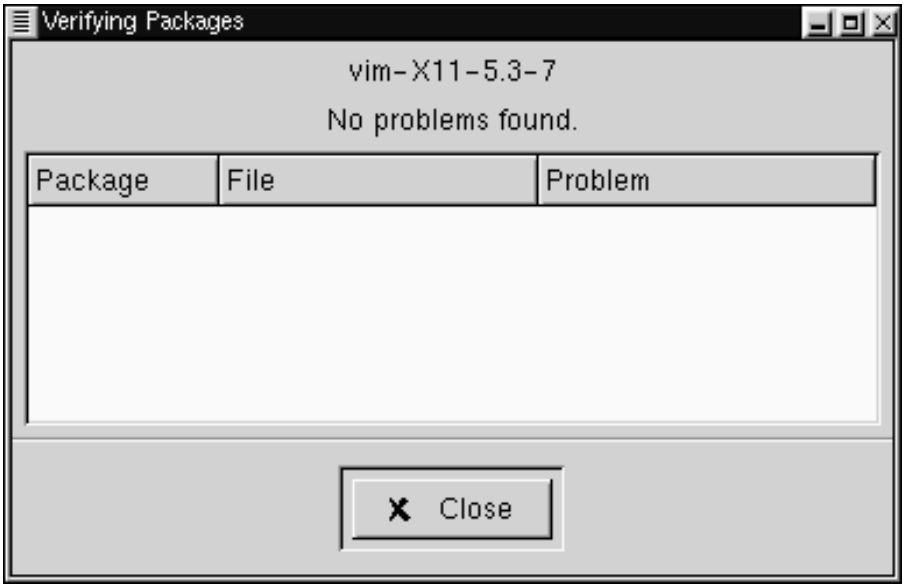
To close the query window without performing any action, left-click on the **x** at the top right of the window bar.

7.5.2 Verifying Packages

Verifying a package checks all of the files in the package to ensure they match the ones present on your system. The checksum, file size, permissions, and owner attributes are all checked against the database. This check can be used when you suspect that one of the program's files has become corrupted for some reason.

Choosing the packages to verify is like choosing the packages to query. Select the packages in the display window and use the **Verify** button on the toolbar or from **Packages** => **Verify** on the menu. A window opens like the one in Figure 7–12, *Verify Window*.

Figure 7–12 Verify Window



As the package is being checked, you'll see the progress in the **Verify** window. If there are any problems discovered during the verify process, they'll be described in the main display area.

7.5.3 Uninstalling Packages

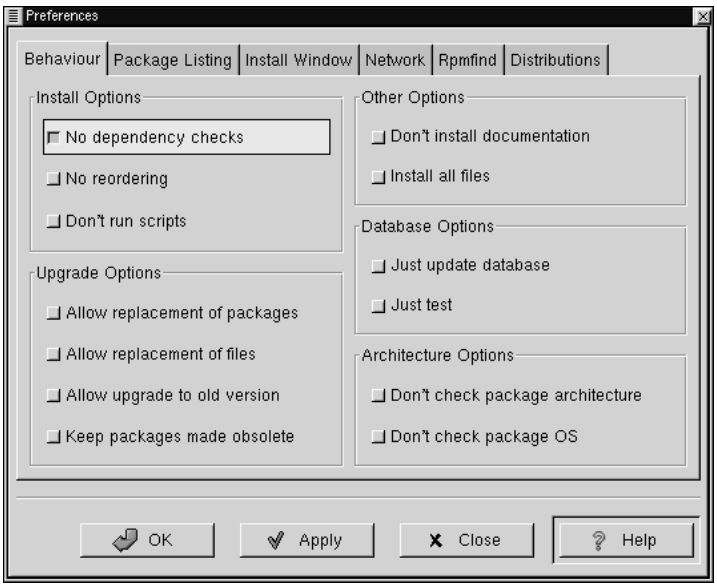
Uninstalling a package removes the application and associated files from your machine. When a package is uninstalled, any files it uses that are not needed by other packages on your system are also removed. Changed configuration files are copied to `<filename>.rpmsave` so you can reuse them later.

Please Note

Remember that you must be root to uninstall packages.

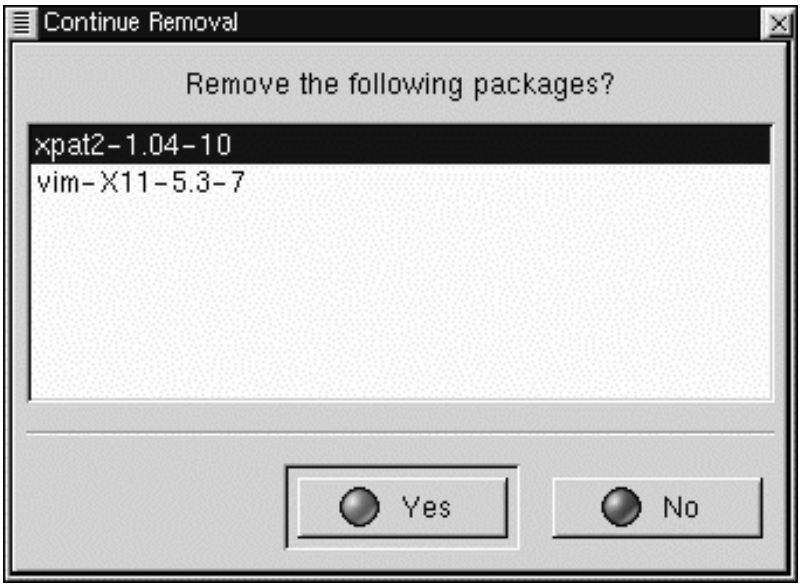
If uninstalling a package would break "dependencies" (which could hobble other applications that require one or more of the removed files in the package), a dialog will pop up, asking you to confirm the deletion. This will occur if you haven't selected the **"No dependency checks"** box from the **Preferences** menu (as shown in Figure 7–13, *The Behavior Tab in Preferences*).

Figure 7–13 The Behavior Tab in Preferences



There are a variety of methods through which you can remove a selected package: from the menu, under **Packages**; from the toolbar and from the **Query** function. If you decide to remove more than one package at a time, you can choose either an incremental or global selection in the same way as you would when installing, querying or verifying. The total of your selections will be reflected in the status bar on the bottom of the main window. Because you can remove more than one package at a time, use caution to select only those which you wish to remove.

Figure 7–14 Uninstall Window



Once you've begun the uninstall, Gnome-RPM asks for confirmation, showing a window like the one in Figure 7–14, *Uninstall Window*. All of the packages that are about to be uninstalled are listed. You should look at them all to ensure you're not about to remove something you want to keep. Clicking the **Yes** button will start the uninstallation process. After it completes, the packages and groups that have been removed will disappear from any windows they were in.

Upgrading Packages

When a new version of a package has been released, it is easy to install it on your system. Select the packages from the window of available packages in the same way you select packages for installation. Both the **Upgrade** button on the toolbar and, from the menu, under **Operations** => **Upgrade** will begin the process. You simply **Add** packages in the same manner as you would a new package installation.

During the upgrade, you'll see a progress indicator like the one for installing packages. When it's finished, any old versions of the packages will be removed, unless you specify otherwise (refer to Section 7.4, *Configuration* for more information).

It is much better to use the upgrade option than to uninstall the old versions of a package and then install the new one. Using upgrade ensures that any changes you made to package configuration files are preserved properly, while uninstalling and then reinstalling a new package could cause those changes to be lost.

If you run out of disk space during an installation, the install will fail. However, the package which was being installed when the error occurred may leave some files around. To clean up after this error, reinstall the package after you've made more disk space available.

8 Lightweight Directory Access Protocol (LDAP)

8.1 What is LDAP?

LDAP (Lightweight Directory Access Protocol) is a proposed open standard for global or local directory services over a network and/or the Internet. A directory, in this sense, is very much like a phone book. LDAP can handle other information, but at present it is typically used to associate names with phone numbers and e-mail addresses. Directories are designed to support a high volume of queries, but the data in the directory doesn't change all that often.

LDAP is much more useful than a paper phone book, because LDAP's design is intended to support propagation over LDAP servers throughout the Internet, much like the Domain Name Service (DNS). The DNS system acts as the address book of the Internet by keeping track of domain name/IP address pairs. DNS servers tell networked machines where packets need to go. In the future, LDAP could provide the same type of global access to many types of directory information: at present, LDAP is more commonly used within a single large organization, like a college or a company, for directory services.

LDAP is a client-server system. An LDAP client connects to an LDAP server and either queries for information or provides information that needs to be entered into the directory. The server either answers the query, refers the query to another LDAP server, or accepts the information for incorporation into the directory.

LDAP is sometimes known as **X.500 Lite**. X.500 is an international standard for directories. X.500 is full-featured, but it is complex and requires lots of computing resources and the full OSI stack. LDAP, in contrast, can run easily on a PC and over TCP/IP. LDAP can access X.500 directories, but it does not support every capability of X.500.

This chapter will refer to the configuration and use of OpenLDAP, an open source implementation of LDAP. OpenLDAP includes `slapd`, a stand-alone LDAP server;

`slurpd`, a stand-alone LDAP replication server; libraries implementing the LDAP protocol; utilities; tools; and sample clients.

8.2 Pros and Cons of LDAP

The main benefit of using LDAP is the consolidation of certain types of information within your organization. For example, all of the different lists of users within your organization can be merged into one LDAP directory. This directory can be queried by any LDAP-enabled applications that need this information. The directory can also be used by users who need directory information.

Other LDAP benefits include its ease of implementation (compared to X.500), and its well-defined Application Programming Interface (API), which means that the number of LDAP-enabled applications and LDAP gateways should increase in the future.

On the negative side, if you want to use LDAP, you'll need LDAP-enabled applications or you'll need to use LDAP gateways. As mentioned previously, LDAP will only increase in usage, but at present, there aren't a plethora of LDAP-enabled applications available for Linux. Also, while LDAP does support some access control, it does not support as many security features as X.500.

8.3 Uses for LDAP

Several Netscape applications, including Netscape Roaming Access are LDAP-enabled. Sendmail 8.[89].x can use LDAP to look up addresses. Your organization can use LDAP as an organization-wide directory and/or name service (in place of NIS or flat files). You can even use a personal LDAP server to keep track of your own e-mail address book (see Section 8.10, *LDAP Resources on the Web*).

LDAP can be used as an authentication service via the `pam_ldap` module. LDAP is commonly used as a central authentication server so that users have a unified login that covers console logins, POP servers, IMAP servers, machines connected to the network using Samba, and even Windows NT machines. All of these login situations can rely on the same user ID and password combination, using LDAP.

8.4 LDAP Terminology

An **entry** is one unit in an LDAP directory. An entry is identified or referenced by its unique **Distinguished Name** (DN).

An entry has attributes; attributes are pieces of information which are directly associated with the entry. For example, an organization could be an LDAP entry. Attributes associated with the organization might be its fax number, its address, and so on. People can also be entries in the LDAP directory. Common attributes for people include their telephone numbers and their e-mail addresses.

Certain attributes are required, while other attributes are optional. An **objectclass** sets which attributes are required and which are optional. Objectclass definitions are found in the `slapd.oc.conf` file.

The **LDAP Data Interchange Format** (LDIF) is an ASCII text format for LDAP entries. Files that import or export data to and from LDAP servers must be in LDIF format. An LDIF entry looks like this:

```
[<id>]
dn: <distinguished name>
<attrtype>: <attrvalue>
<attrtype>: <attrvalue>
<attrtype>: <attrvalue>
```

An entry can contain as many <attrtype>: <attrvalue> pairs as needed. A blank line indicates that the entry is finished and that another entry is about to begin.

Everything enclosed within < > is variable, and can be set by you when you add an LDAP entry, with the exception of the <id>. The <id> is a number normally set by the LDAP tools when you add an entry, and you'll probably never need to manually set one.

8.5 OpenLDAP Files

OpenLDAP configuration files are installed into the `/etc/openldap` directory. If you do an `ls` on `/etc/openldap`, you'll see something like:

```
$ ls
ldap.conf        ldapsearchprefs.conf  slapd.at.conf  slapd.oc.conf
ldapfilter.conf  ldaptemplates.conf    slapd.conf
```

```
$
```

The files you should know about are `slapd.conf`, `slapd.at.conf` and `slapd.oc.conf`. The `slapd.conf` file contains configuration information for the `slapd` daemon and for all of the database back-ends. You will need to make some changes to `slapd.conf` before you start the `slapd` daemon.

The `slapd.at.conf` file holds **attribute syntax definitions** for the LDAP directory. Attribute syntax definitions describe the type of information that the attribute provides. You have five choices:

- bin -- binary information
- ces -- case exact string (a string whose case matters)
- cis -- case ignore string (a string whose case doesn't matter)
- tel -- a telephone number (blank spaces and hyphens are ignored)
- dn -- the distinguished name

For example, here are attribute syntax definitions from a typical `slapd.at.conf`:

```
attribute       photo                                   bin
attribute       personalsignature                       bin
attribute       jpegphoto                               bin
attribute       audio                                   bin
attribute       labeledurl                              ces
attribute       ref                                     ces
attribute       userpassword                            ces
attribute       telephonenumber                         tel
```

Certain LDAP-enabled applications may require that you edit the `slapd.at.conf` file, usually to add particular attribute syntax definitions.

The `slapd.oc.conf` file includes the **objectclass definitions** for an LDAP directory. The objectclass definitions state which attributes are required and which attributes are optional for particular objectclasses.

The following excerpt from a typical `slapd.oc.conf` file contains the objectclass definitions for the top, alias, and referral objectclasses:

```
objectclass top
        requires
```

```
                objectClass

objectclass alias
        requires
                aliasedObjectName,
                objectClass

objectclass referral
        requires
                ref,
                objectClass
```

You may need to edit the objectclass definitions in your `slapd.oc.conf`, depending upon how you're going to use your LDAP directory. For example, if you're creating an LDAP directory of employees for use in your organization, you'll probably have specific required attributes for certain objectclasses that might not be used outside the organization (e.g., an employee ID number internal to your organization might be a required attribute for an objectclass of "person").

8.6 OpenLDAP Daemons and Utilities

The OpenLDAP package includes two daemons: `slapd` and `slurpd`. The `slapd` daemon is the stand-alone LDAP daemon, which you'll need to run to support LDAP.

The `slurpd` daemon controls the replication of LDAP directories over a network. `Slurpd` sends changes from the master LDAP directory to slave LDAP directories. You won't need to run `slurpd` unless you have more than one LDAP server on your network. If you have two or more LDAP servers, you'll need to run `slurpd` to keep the LDAP directories in sync.

OpenLDAP also includes some utilities for adding, modifying and deleting entries in an LDAP directory. The `ldapmodify` tool is used to modify entries in an LDAP database. The `ldapadd` utility is used to add entries to your directory (`ldapadd` is actually a hard link to `ldapmodify -a`). `Ldapsearch` is used to search for entries and `ldapdelete` is used to delete entries. The `ldif2ldbm` tool converts an LDIF file into an LDBM back-end database.

See their man pages for more information on all of these utilities.

8.7 Modules for Adding Extra Functionality to LDAP

Red Hat Linux includes the following packages which add functionality to LDAP:

The `nss_ldap` module is an LDAP module for the **Solaris Nameservice Switch** (NSS). NSS is a set of C library extensions necessary for accessing LDAP directory information, instead of or in addition to the **Network Information Service** (NIS) name service and/or flat files. The `nss_ldap` module is needed to use LDAP as a native name service.

The `pam_ldap` module is needed to integrate LDAP authentication into the Pluggable Authentication Modules (PAM) API. If you use `pam_ldap`, users can authenticate and change their password using LDAP directories. The `nss_ldap` and `pam_ldap` modules are provided in the `nss_ldap` package.

Red Hat Linux also includes LDAP modules for the Apache Web server. The `auth_ldap` module is for authenticating HTTP clients against the user entries in an LDAP directory. The `php-ldap` module adds LDAP support to the PHP3 HTML-embedded scripting language. The `auth_ldap` and `php-ldap` modules will need to be compiled into Apache as **Dynamic Shared Objects** (DSOs).

8.8 LDAP How To: A Quick Overview

This section provides a quick overview of the steps you'll need to take to get an LDAP directory working.

1. Make sure the `openldap` RPM, and any other LDAP-related RPMS that you need, are installed.
2. See either the Quick Start Guide at the OpenLDAP site (http://www.openldap.org/faq/data/cache/172.html; start at "Create configuration file for slapd," since the LDAP files are already installed), or see the Linux-LDAP HOWTO (http://www.linuxdoc.org/HOWTO/LDAP-HOWTO.html) for instructions on using LDAP on your system. Both cover the rest of these steps.
3. Edit the `slapd.conf` file to get it right for your system.
4. Start `slapd`.

5. Create your LDAP directory (examples of LDAP entries are provided at the PADL Software website at http://www.padl.com/ldap_examples.html).
6. Add entries to your LDAP directory with `ldapadd` or with a script.
7. Use `ldapsearch` to see if `slapd` is working.
8. At this point, your LDAP directory should exist. The next step is to configure your LDAP-enabled applications so that they can use the LDAP directory.

8.9 Configuring Your System to Authenticate Using OpenLDAP

This section provides a brief overview of how to configure your Red Hat Linux system to authenticate using OpenLDAP. Unless you're an OpenLDAP expert, you will probably need more documentation than is provided here. Please refer to the references provided in Section 8.10, *LDAP Resources on the Web* for more information.

8.9.1 Install the Necessary LDAP Packages

First, you'll need to make sure that the appropriate packages are installed on both the LDAP server and the LDAP client machines. The LDAP server needs the `openldap` package.

The LDAP client machines need the following packages installed: `openldap`, `auth_ldap`, `nss_ldap` and `pam_ldap`.

8.9.2 Edit Configuration Files

Edit `/etc/openldap/slapd.conf`

The `slapd.conf` file, located in `/etc/openldap`, contains the configuration information needed by your `slapd` LDAP server. You'll need to edit this file to make it specific to your domain and your server.

The suffix line names the domain for which the LDAP server will provide information. The suffix line should be changed from:

```
suffix   "dc=your-domain, dc=com"
```

so that it reflects your domain name. For example:

```
suffix  "dc=acmewidgets, dc=com"
```

or

```
suffix   "dc=acmeuniversity, dc=org"
```

The rootdn entry is the DN for a user who is unrestricted by the access control or administrative limit parameters set for operations on the LDAP directory. The rootdn user can be thought of as the root user for the LDAP directory. The rootdn line needs to be changed from:

```
rootdn  "cn=root, dc=your-domain, dc=com"
```

to something like:

```
rootdn  "cn=root, dc=redhat, dc=com"
```

or

```
rootdn  "cn=ldapmanager, dc=my_organization, dc=org"
```

Change the rootpw line from:

```
rootpw   secret
```

to something like

```
rootpw  {crypt}s4L9sOIJo4kBM
```

In the above example, you're using an encrypted root password, which is a much better idea than leaving a plain text root password in the `slapd.conf` file. To make this crypt string, you should either copy it out of a `passwd` file, or use Perl:

```
perl -e "print crypt('passwd','a_salt_string');"
```

In the previous Perl line, `salt_string` is a two character salt, and `passwd` is the plain text version of the password.

You could also copy a `passwd` entry out of `/etc/passwd`, but this won't work if the `passwd` entry is an MD5 password (the default in Red Hat Linux 6.2).

Edit `ldap.conf`

Edit the `ldap.conf` files in `/etc` and in `/etc/openldap` on the LDAP server and clients.

Edit `/etc/ldap.conf`, the configuration file for `nss_ldap` and `pam_ldap`, to reflect your organization and search base. The file `/etc/openldap/ldap.conf` is the configuration file for the command line tools like `ldapsearch`, `ldapadd`, etc., and it will also need to be edited for your LDAP setup. Client machines will need to have both of these files modified for your system.

Edit `/etc/nsswitch.conf`

To use `nss_ldap`, you'll need to add `ldap` to the appropriate fields in `/etc/nsswitch.conf`. (Be very careful when editing this file; be sure that you know what you're doing.) For example:

```
passwd: files ldap
shadow: files ldap
group: files ldap
```

Copy the PAM Configuration Files

To use `pam_ldap`, you'll need to copy the PAM configuration files from `/usr/doc/nss_ldap<version>/pam.d/` to your `/etc/pam.d/` directory. These are a set of PAM configuration files that allow all of the standard PAM-enabled applications to use LDAP for authentication. (PAM is beyond the scope of this LDAP overview, so if you need help, consult Section 2.6, *User Authentication with PAM* and/or PAM man pages.)

8.9.3 Migrate Your Old Authentication Information to LDAP Format

The `/usr/share/openldap/migration` directory contains a set of shell and Perl scripts for migrating your old authentication information into LDAP format. (Yes, you'll need to have Perl on your system to use these scripts.)

First, you'll need to modify the `migrate_common.ph` file so that it reflects your domain. The default DNS domain should be changed from:

```
$DEFAULT_MAIL_DOMAIN = "padl.com";
```

to something like:

```
$DEFAULT_MAIL_DOMAIN = "your_company.com";
```

The default base should also be changed, from:

```
$DEFAULT_BASE = "dc=padl,dc=com";
```

to something like:

```
$DEFAULT_BASE = "dc=your_company,dc=com";
```

Next, you'll need to decide which script to use. The following table should tell you:

Table 8–1 LDAP Migration Scripts

Existing name service	Is LDAP running?	Use this script:
`/etc` flat files	yes	`migrate_all_online.sh`
`/etc` flat files	no	`migrate_all_offline.sh`
NetInfo	yes	`migrate_all_netinfo_online.sh`
NetInfo	no	`migrate_all_netinfo_offline.sh`
NIS (YP)	yes	`migrate_all_nis_online.sh`
NIS (YP)	No	`migrate_all_nis_offline.sh`

Run the appropriate script based on your existing name service.

The `README` and the `migration-tools.txt` files in `/usr/share/openldap/migration` provide more details.

8.10 LDAP Resources on the Web

Lots of useful information about LDAP can be found on the Web. Please review these sources, especially the OpenLDAP website and the LDAP HOWTO, before you start to set up LDAP on your system.

OpenLDAP

http://www.openldap.org

University of Michigan

http://www.umich.edu/~dirsvcs/ldap/

The SLAPD and SLURPD Administrator's Guide

http://www.umich.edu/~dirsvcs/ldap/doc/guides/slapd

Innosoft/Critical Angle

http://www.innosoft.com/ldapworld

Jeff Hodges' LDAP Road Map and FAQ

http://www.kingsmountain.com/ldapRoadmap.shtml

PADL (nss_ldap, pam_ldap and ypldapd)

http://www.padl.com/

auth_ldap

http://www.rudedog.org/auth_ldap/1.4/auth_ldap.html

The LDAP HOWTO

http://www.linuxdoc.org/HOWTO/LDAP-HOWTO.html

Sendmail 8.[89].x using LDAP to do address lookups

http://www.stanford.edu/~bbense/Inst.html

Using a personal LDAP server, the Apache Web server and PHP to manage your personal e-mail addressbook

http://www.webtechniques.com/archives/1999/05/junk/junk.shtml

Also, remember that man pages exist for the various LDAP daemons and utilities. Please check the man pages if you need more information.

Part II Installation-Related Reference

9 Preparing for a Text Mode Installation

9.1 Things You Should Know

Before attempting to install Red Hat Linux, you should collect some system information in order to prevent any surprises during the installation. You can find most of this information in the documentation that came with your system, or from the system's vendor or manufacturer.

The most recent list of hardware supported by Red Hat Linux can be found at http://www.redhat.com/hardware. It's a good idea to check your hardware against this list before proceeding.

Please Note

You can perform a text mode installation of Red Hat Linux 6.2 by following instructions in this chapter and in Chapter 10, *Installing Red Hat Linux via Text Mode*. However, if you're installing from a CD-ROM, you might prefer to use the graphical installation mode, which offers both ease of use and a flexible, custom-class installation mode. For more information on graphical installations, turn to the *Official Red Hat Linux Installation Guide*.

Tip

At the end of the *Before You Begin* chapter in the *Official Red Hat Linux Installation Guide* is a table for you to fill out with your specific system requirements, which will help you keep up with any information you will need during your installation.

9.1.1 Basic Hardware Configuration

You should have a basic understanding of the hardware installed in your computer, including:

- hard drive(s) -- Specifically, the number, size, and type. If you have more than one, it's helpful to know which one is first, second, and so on. It is also good to know if your drives are IDE or SCSI. If you have IDE drives, you should check your computer's BIOS to see if you are accessing them in **linear mode**. Please refer to your computer's documentation for the proper key sequence to access the BIOS. Note that your computer's BIOS may refer to linear mode by other names, such as "large disk mode." Again, your computer's documentation should be consulted for clarification.
- memory -- The amount of RAM installed in your computer.
- CD-ROM -- Most importantly, the unit's interface type (IDE, SCSI, or other interface) and, for non-IDE, non-SCSI CD-ROMs, the make and model number. IDE CD-ROMs (also known as ATAPI) are the most common type in recently manufactured, PC-compatible computers.
- SCSI adapter (if one is present) -- The adapter's make and model number.
- network card (if one is present) -- The card's make and model number.
- mouse -- The mouse's type (serial, PS/2, or bus mouse), protocol (Microsoft, Logitech, MouseMan, etc.), and number of buttons; also, for serial mice, the serial port it is connected to.

On many newer systems, the installation program is able to automatically identify most hardware. However, it's a good idea to collect this information anyway, just to be sure.

Learning About Your Hardware with Windows

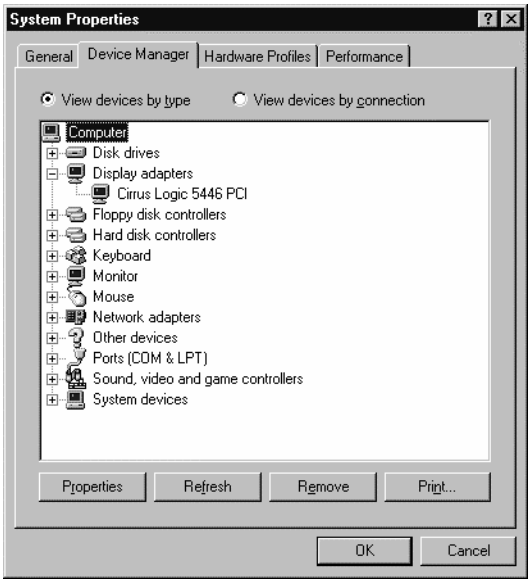
If your computer is already running Windows 9*x*, you can use the following procedure to get additional configuration information:

Figure 9–1 Windows System Properties

- In Windows, click on the **My Computer** icon using the secondary (normally the right) mouse button. A pop-up menu should appear.
- Select **Properties**. The **System Properties** window should appear. Note the information listed under **Computer** -- in particular the amount of RAM listed.
- Click on the **Device Manager** tab. You will then see a graphical representation of your computer's hardware configuration. Make sure the **View devices by type** button is selected.

At this point, you can either double-click on the icons (or single-click on the plus sign **+**) to look at each entry in more detail. Look under the following icons for more information:

Figure 9–2 Device Manager Under Windows 95



- **Disk drives** -- You will find the type (IDE or SCSI) of hard drive here. (IDE drives will normally include the word "IDE," while SCSI drives won't.)
- **Hard disk controllers** -- You can get more information about your hard drive controller here.
- **CDROM** -- Here is where you'll find out about any CD-ROM drives connected to your computer.

Please Note

In some cases, there may be no CD-ROM icon, yet your computer has a functioning CD-ROM drive. This is normal, depending on how Windows was originally installed. In this case, you may be able to learn additional information by looking at the CD-ROM driver loaded in your computer's `config.sys` file.

- **Mouse** -- The type of mouse present on your computer can be found here.
- **Display adapters** -- If you're interested in running the X Window System, you should write down the information you find here.
- **Sound, video and game controllers** -- If your computer has sound capabilities, you'll find more information about that here.
- **Network adapters** -- Here you'll find additional information on your computer's network card (if you have one).
- **SCSI controllers** -- If your computer uses SCSI peripherals, you'll find additional info on the SCSI controller here.

While this method is not a complete substitute for opening your computer's case and physically examining each component, in many cases it can provide sufficient information to continue with the installation.

Please Note

This information can also be printed by clicking on the **Print** button. A second window will appear, allowing you to choose the printer, as well as the type of report (the **All Devices and System Summary** report type is the most complete).

9.1.2 Video Configuration

If you will be installing the X Window System, you should also be familiar with the following:

- your video card -- The card's make and model number (or the video chipset it uses), and the amount of video RAM it has. (Most PCI-based cards are auto-detected by the installation program.)
- your monitor -- The unit's make and model number, along with allowable ranges for horizontal and vertical refresh rates. (Newer models may be auto-detected by the installation program.)

9.1.3 Network-related Information

If you're connected to a network, be sure you know the following:

- IP address -- Usually represented as a set of four numbers separated by dots, such as `10.0.2.15`.
- netmask -- Another set of four numbers separated by dots. An example netmask would be `255.255.248.0`.
- gateway IP address -- Yet another set of four dot-separated numbers. For instance, `10.0.2.254`.
- one or more name server IP addresses -- One or more sets of dot-separated numbers. For example, `10.0.2.1` might be the address of a name server.
- domain name -- The name your organization uses. For instance, Red Hat has a domain name of `redhat.com`.
- hostname -- The name assigned to your individual system. A computer might be named `pooh`, for instance.

Please Note

The information given above is an example only! Do *not* use it when you install Red Hat Linux! If you don't know the proper values for your network, ask your network administrator.

10 Installing Red Hat Linux via Text Mode

This release of Red Hat Linux features a graphical, mouse-based installation program, documented in the *Official Red Hat Linux Installation Guide*. But you can also install Red Hat Linux with a text mode, keyboard-based installation program. This chapter explains how to use it. Here are some recommendations:

- If you're new to Linux installations, read the *Official Red Hat Linux Installation Guide* first. Although it focuses on the graphical installation process, most of the concepts apply to the text mode installation as well. After that, you'll find that Chapter 9, *Preparing for a Text Mode Installation*, *Preparing for a Text Mode Installation*, will give you more in-depth information regarding those aspects of installing Red Hat Linux that do not apply to the graphical installation process.

  Additionally, you'll find Appendix B, *An Introduction to Disk Partitions*, *An Introduction to Disk Partitions*, helpful, as it discusses disk partition resizing (crucial if you plan to install Linux on a disk where another operating system is currently installed).
- If you'll need PCMCIA support to perform the installation (for example, you're installing on a laptop equipped with a PCMCIA card), you must make a PCMCIA boot disk. The *Official Red Hat Linux Installation Guide* explains how to do this.
- If you plan to install over a network (via NFS, FTP, or HTTP), you must make a network boot disk. The *Official Red Hat Linux Installation Guide* explains how to do this.
- If you've never used the text mode installation program, or need a refresher on its user interface, read the next section.
- To begin installation without further delay, turn to Section 10.2, *Starting the Installation Program*.

10.1 The Installation Program User Interface

The Red Hat Linux text mode installation program uses a screen-based interface that includes most of the on-screen "widgets" commonly found on graphical user interfaces. Figure 10–1, *Installation Program Widgets as seen in* **Configure TCP/IP** and

Figure 10–2, *Installation Program Widgets as seen in* Disk Druid illustrate the screens you'll see.

Figure 10–1 Installation Program Widgets as seen in Configure TCP/IP

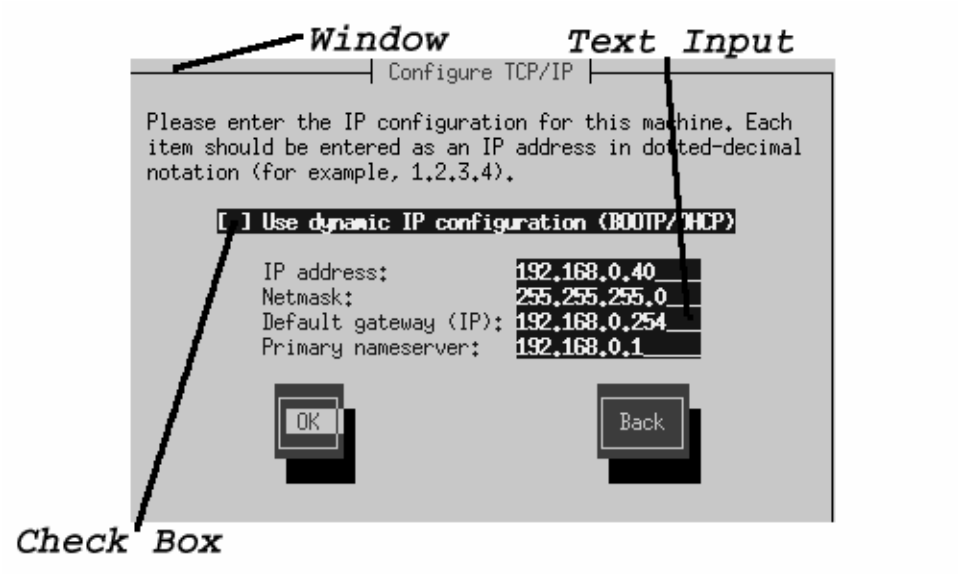
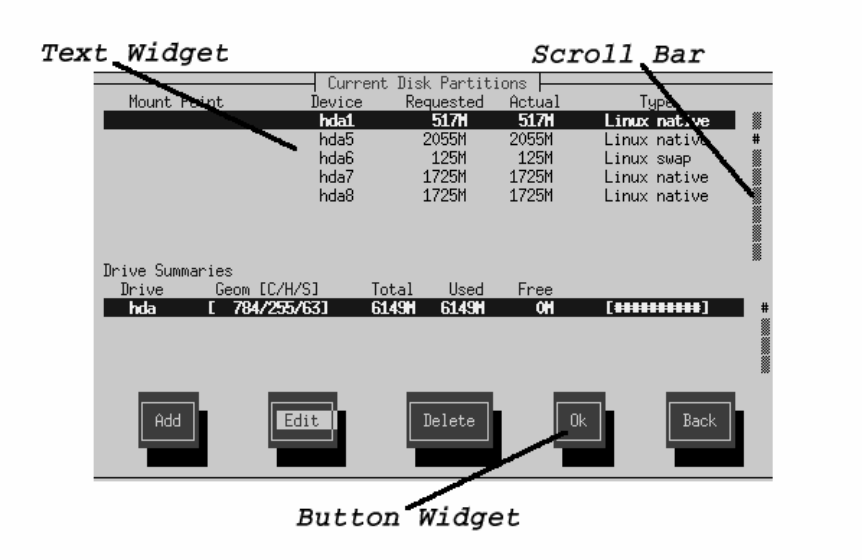


Figure 10–2 Installation Program Widgets as seen in Disk Druid



Here's a list of the most important widgets shown in Figure 10–1, *Installation Program Widgets as seen in* **Configure TCP/IP** and Figure 10–2, *Installation Program Widgets as seen in* Disk Druid :

- Window -- Windows (usually referred to as **dialogs** in this manual) will appear on your screen throughout the installation process. At times, one window may overlay another; in these cases, you can only interact with the window on top. When you are finished in that window, it will disappear, allowing you to continue working in the window underneath.
- Text Input -- Text input lines are regions where you can enter information required by the installation program. When the cursor rests on a text input line, you may enter and/or edit information on that line.
- Check Box -- Check boxes allow you to select or deselect a feature. The box displays either an asterisk (selected) or a space (unselected). When the cursor is

within a check box, press [Space] to select an unselected feature or to deselect a selected feature.

- Text Widget -- Text widgets are regions of the screen for the display of text. At times, text widgets may also contain other widgets, such as check boxes. If a text widget contains more information than can be displayed in the space reserved for it, a scroll bar appears; if you position the cursor within the text widget, you can then use the [Up] and [Down] arrow keys to scroll through all the information available. Your current position is shown on the scroll bar by a **#** character, which moves up and down the scroll bar as you scroll.
- Button Widget -- Button widgets are the primary method of interacting with the installation program. You progress through the windows of the installation program by navigating these buttons, using the [Tab] and [Enter] keys. Buttons can be selected when they are highlighted.
- Cursor -- Although not a widget, the cursor is used to select (and interact) with a particular widget. As the cursor is moved from widget to widget, it may cause the widget to change color, or you may only see the cursor itself positioned in or next to the widget. In Figure 10–1, *Installation Program Widgets as seen in* **Configure TCP/IP**, the cursor is positioned on the **OK** button. Figure 10–2, *Installation Program Widgets as seen in* Disk Druid shows the cursor on the **Edit** button.

10.1.1 Using the Keyboard to Navigate

Navigation through the installation dialogs is performed through a simple set of keystrokes. To move the cursor, use [Left], [Right], [Up], and [Down] arrow keys. Use [Tab], and [Alt]-[Tab] to cycle forward or backward through each widget on the screen. Most screens display along the bottom a summary of available cursor positioning keys.

To "press" a button, position the cursor over the button (using [Tab], for instance) and press [Space] or [Enter]. To select an item from a list of items, move the cursor to the item you wish to select and press [Enter]. To select an item with a **check box**, move the cursor to the check box and press [Space] to select an item. To deselect, press [Space] a second time.

Pressing [F12] accepts the current values and proceeds to the next dialog; it is equivalent to pressing the **OK** button.

CAUTION

Unless a dialog box is waiting for your input, do not press any keys during the installation process (doing so may result in unpredictable behavior).

10.2 Starting the Installation Program

There are several ways to start the installation. You can:

- insert the diskette included in your boxed set (or a PCMCIA boot or network boot disk that you've created) into the primary diskette drive and reboot your computer
- insert the Red Hat Linux CD into the drive and reboot, if your computer can boot from the CD-ROM drive
- boot MS-DOS, and start a program in the `dosutils` directory of the Red Hat Linux CD-ROM named `autoboot.bat` (this will work from DOS only; it will not work from a DOS window started from Windows)

While the installation program loads, messages will scroll on your screen. When the installation program has loaded, this prompt appears:

```
boot:
```

10.2.1 Displaying Online Help

Once the installation program is loaded into memory, you can obtain information about the installation process and options by pressing [F1] through [F6]. For example, press [F2] to see general information about the online help screens.

10.2.2 Text Mode Boot Options

If you press [Enter] at the `boot` prompt, or if you take no action within the first minute after the `boot` prompt appears, the graphical installation program, as explained in the *Official Red Hat Linux Installation Guide*, starts. Pressing one of the help screen function keys as described above disables this autostart feature.

To start the text mode installation program explained here, before pressing [Enter], type:

```
boot: text
```

If the installation program does not properly detect your hardware, you may need to restart the installation in "expert" mode. To start an expert mode installation, type:

```
boot: text expert
```

Expert mode disables most hardware probing, and gives you the option of entering options for the drivers loaded during the installation.

Please Note

The initial boot messages will not contain any references to SCSI or network cards. This is normal; these devices are supported by modules that are loaded during the installation process.

Note that the command to start a serial installation has changed. If you must perform the installation in serial mode, type:

```
boot: linux console=<device>
```

Where <*device*> should be the device you are using (such as ttyS0 or ttyS1).

Other options that may be entered with the boot command include passing options to the kernel. For example, to instruct the kernel to use all the RAM in a 128 MB system, enter:

```
boot: linux mem=128M
```

10.3 Choosing a Language

Using the [Up] and [Down] arrow keys, select the appropriate language for both the installation program and the system default, and press [Enter] (Figure 10–3, *Selecting a Language*).

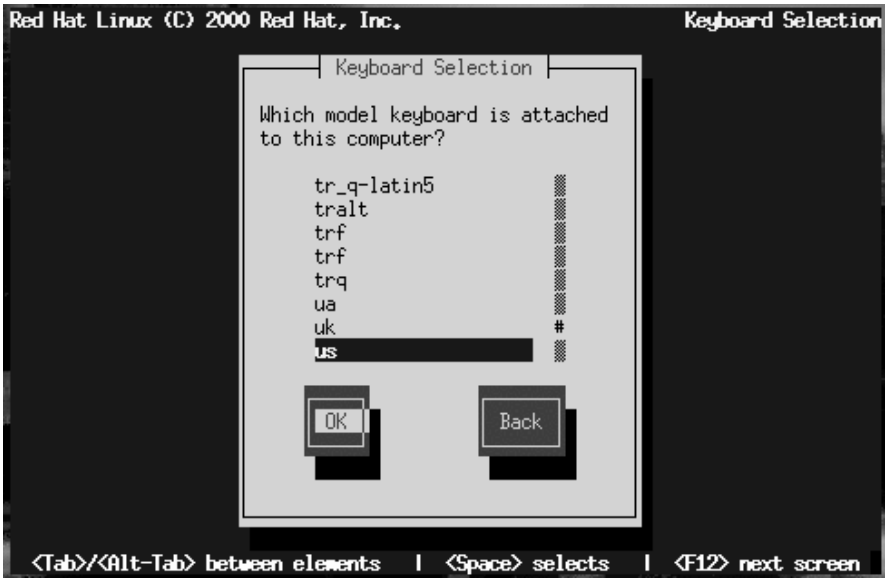
Figure 10–3 Selecting a Language

A scroll bar appears to the right of the list. This indicates that there are more entries than can be displayed in the available space at once. You'll see other scroll bars like this throughout the installation program.

10.4 Selecting a Keyboard Type

Next, choose a keyboard type (Figure 10–4, *Selecting a Keyboard Type*).

Figure 10–4 Selecting a Keyboard Type



After selecting the appropriate keyboard type, press [Enter]; the keyboard type you select will be loaded automatically both for the remainder of the installation process and each time you boot your Red Hat Linux system.

Tip

If you wish to change your keyboard type after you have installed your Red Hat Linux system, become root and type either `/usr/sbin/kbdconfig` or `setup` at the root prompt.

10.5 Selecting an Installation Method

If you booted directly from the Red Hat Linux CD-ROM, you'll see the **Welcome** dialog: turn to Section 10.8, *Welcome*.

Otherwise, an **Installation Method** dialog appears. The choices presented in the dialog vary depending on the type of diskette you booted from (the one in your boxed set, or a network or PCMCIA boot disk that you created).

Figure 10–5, *Installation Method Dialog* shows all available choices.

Figure 10–5 Installation Method Dialog

Please Note

If you are performing a network installation and are copying the files from the Red Hat Linux CD-ROM or an FTP site, be sure to check the file permissions to make sure they are set correctly for your installation. If you do not, the files that you copy will not be executable and you will have to change the permissions before you are able to install.

Red Hat Linux can be installed via any of the following:

Local CDROM

If you booted from the diskette in your boxed set and are installing with the Red Hat Linux CD-ROM in a local drive. If you choose this method, you'll next see the **Welcome** dialog: turn to Section 10.8, *Welcome*.

NFS Image

If you booted from a network or PCMCIA boot disk and are installing from an NFS Image server which is exporting the Red Hat Linux CD-ROM (or a mirror image of one.) If you choose this method, you'll next see the **NFS Setup** dialogs: turn to Section 10.7, *Installing over a Network*.

FTP

If you booted from a network or PCMCIA boot disk and are installing directly from an FTP server. If you select this method, you'll next see the **FTP Setup** dialogs: turn to Section 10.7, *Installing over a Network*.

HTTP

If you booted from a network or PCMCIA boot disk and are installing directly from an HTTP Web server. If you select this method, you'll next see the **HTTP Setup** dialogs: turn to Section 10.7, *Installing over a Network*.

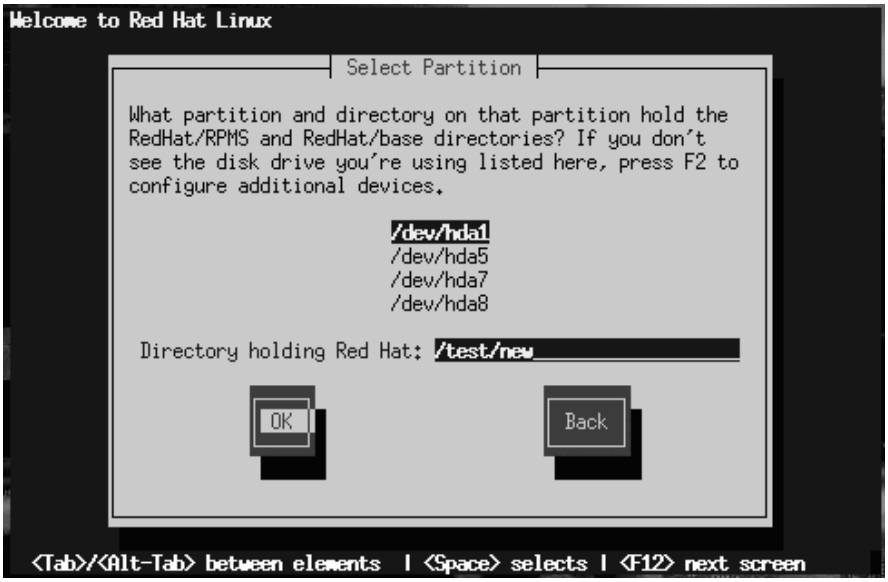
Hard Drive

If you booted from the diskette in your boxed set and are installing from the Red Hat Linux files that you have previously copied to a local hard drive. If you select this method, you'll next see the **Select Partition** dialog: turn to Section 10.6, *Identify Disk Partition to Install From*.

10.6 Identify Disk Partition to Install From

The **Select Partition** screen (Figure 10–6, *Selecting Partition Dialog for Hard Drive Installation*) applies only if you are installing from a disk partition (that is, if you selected **Hard Drive** in the **Installation Method** dialog). This dialog allows you to name the disk partition you are installing from.

Figure 10–6 Selecting Partition Dialog for Hard Drive Installation



Enter the device name of the partition containing the `RedHat` directory tree. There is also a field labelled **Directory holding Red Hat**. If the `RedHat` directory is not in the root directory of that partition, enter the path to the `RedHat` directory (for example, if the `RedHat` directory is at `/test/new/RedHat`, you would enter `/test/new`).

After you've identified the disk partition, you'll next see the **Welcome** dialog: Turn now to Section 10.8, *Welcome*.

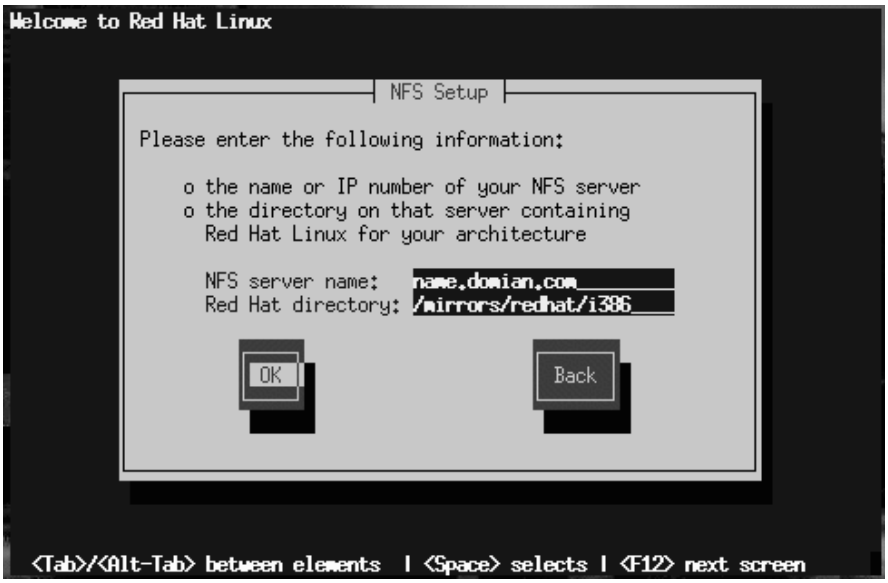
10.7 Installing over a Network

If you are performing a network installation, the **Configure TCP/IP** dialog appears; for an explanation of this dialog, go to Section 10.13, *Configuring a Network Connection* and then return here.

10.7.1 NFS Setup

The NFS setup dialog (Figure 10–7, *NFS Setup Dialog*) applies only if you are installing from an NFS server (that is, if you booted from a network or PCMCIA boot disk and selected **NFS Image** in the **Installation Method** dialog).

Figure 10–7 NFS Setup Dialog



Enter the fully-qualified domain name or IP address of your NFS server, and the name of the exported directory that contains the Red Hat Linux installation files or CD. For example, if you're installing from a host named `eastcoast` in the domain `redhat.com`, enter `eastcoast.redhat.com` in the **NFS Server** field.

If the NFS server has the Red Hat Linux CD mounted on `/mnt/cdrom`, enter `/mnt/cdrom` in the **Red Hat directory** field. If the NFS server is exporting a mirror of the Red Hat Linux installation tree instead of a CD, enter the directory which contains the `RedHat` directory. For example, if your NFS server contains the directory `/mirrors/redhat/i386/RedHat`, enter `/mirrors/redhat/i386`.

Please Note

If you are performing an NFS installation and are copying the files from the Red Hat Linux CD-ROM, be sure to check the file permissions to make sure they are set correctly for your installation. If you do not, the files that you copy will not be executable and you will have to change the permissions before you are able to install.

Next you'll see the **Welcome** dialog: Turn now to Section 10.8, *Welcome*.

10.7.2 FTP Setup

The **FTP Setup** screen (Figure 10–8, *FTP Setup Diallog*) applies only if you are installing from an FTP server (that is, if you selected **FTP** in the **Installation Method** dialog). This dialog allows you to identify the FTP server you are installing from.

Figure 10–8 FTP Setup Diallog

Enter the fully-qualified domain name or IP address of the FTP site you are installing from, and the name of the directory there which contains the `RedHat` installation files for your architecture. For example, if the FTP site contains the directory `/pub/mirrors/redhat/i386/RedHat`, enter **`/pub/mirrors/redhat/i386`**.

If everything has been specified properly, a message box appears indicating that `base/hdlist` is being retrieved.

Next you'll see the **Welcome** dialog: Turn now to Section 10.8, *Welcome*.

10.7.3 HTTP Setup

The **HTTP Setup** screen (Figure 10–9, *HTTP Setup Dialog*) applies only if you are installing from an HTTP server (that is, if you selected **HTTP** in the **Installation Method** dialog). This dialog prompts you for information about the HTTP server you are installing from.

Figure 10–9 HTTP Setup Dialog

Enter the name or IP address of the HTTP site you are installing from, and the name of the directory there which contains the `RedHat` installation files for your architecture. For example, if the HTTP site contains the directory `/pub/mirrors/redhat/i386/RedHat`, enter **`/pub/mirrors/redhat/i386`**.

If everything has been specified properly, a message box appears indicating that `base/hdlist` is being retrieved.

Next you'll see the **Welcome** dialog: Turn now to Section 10.8, *Welcome*.

10.8 Welcome

Once you've selected your language and keyboard, and indicated the installation method, Figure 10–10, *Welcome dialog* appears. Press **OK** to continue.

Figure 10–10 Welcome dialog

10.9 Upgrading or Installing

The **Installation Type** dialog (Figure 10–11, *Installation Type Dialog*) presents you with five choices, described below.

Figure 10–11 Installation Type Dialog

- **Install GNOME Workstation** -- Install on a system that will be used primarily as a workstation. Load the GNOME GUI and configure the system to start GNOME when the system boots. The installation program deletes all data in any existing Linux partitions, decides how to partition the disk for the new version, and chooses which software packages to load.

WARNING

Do not choose this method if you're sharing a disk with Windows NT; if you do, you will be unable to boot Windows NT. LILO will write over NT's boot loader and you will be unable to boot NT. You must perform a custom-class installation and configure LILO so that it is not installed on the Master Boot Record (MBR).

To create a dual-boot environment on a system that currently has NT, you must install LILO on the first sector of the root partition, not the MBR. Please be sure to create a boot disk. In a case such as this, you will either need to use the boot disk, or configure the NT system loader to boot LILO from the first sector of the root partition. Be sure to check out http://www.linux-doc.org/HOWTO/mini/Linux+NT-Loader.html for more information on setting up LILO and NT.

- **Install KDE Workstation** -- Install on a system that will be used primarily as a workstation. Load the KDE GUI and configure the system to start KDE when the system boots. The installation program deletes all data in any existing Linux partitions, decides how to partition the disk for the new version, and chooses which software packages to load.

WARNING

Do not choose this method if you're sharing a disk with Windows NT; if you do, you will be unable to boot Windows NT. LILO will write over NT's boot loader and you will be unable to boot NT. You must perform a custom-class installation and configure LILO so that it is not installed on the Master Boot Record (MBR).

To create a dual-boot environment on a system that currently has NT, you must install LILO on the first sector of the root partition, not the MBR. Please be sure to create a boot disk. In a case such as this, you will either need to use the boot disk, or configure the NT system loader to boot LILO from the first sector of the root partition. Be sure to check out http://www.linux-doc.org/HOWTO/mini/Linux+NT-Loader.html for more information on setting up LILO and NT.

- **Install Server System** -- Install on a system that will be used primarily as a server. The X Window System is not configured and no GUI starts when the system boots. The Installation program deletes *all* data in *all* existing partitions of any kind, decides how to partition the disk for the new version, and chooses which software packages to load.
- **Install Custom System** -- Perform a custom installation. You make all decisions regarding disk partitioning and initialization, which software packages to install, and how to configure the X Window System and the user interface.

- **Upgrade Existing System** -- Upgrade an earlier version of Red Hat Linux (2.0 or later) without deleting any existing data. The installation program updates the modular 2.2.x kernel and all currently installed software packages.

10.9.1 Upgrading

If you choose to upgrade and the installation program detects more than one installed Linux version on the system, you'll be asked which version to upgrade. After you indicate this, or if there's only one installed Linux version on the system, the installation program probes your existing system to determine which software packages require updating and presents the **Customize Packages to Upgrade** dialog (Figure 10–12, *Customize Packages to Upgrade Dialog*).

Figure 10–12 Customize Packages to Upgrade Dialog

If you answer **No**, the installation program starts upgrading existing packages.

Answer **Yes** if you want to add to or remove items from the list of individual packages to be upgraded. The package selection dialog is seen in Section 10.19.1, *Selecting Individual Packages*. The upgrade starts when you finish making your changes.

Please Note

Some upgraded packages may require that other packages are also installed for proper operation. The upgrade procedure takes care of these **dependencies**, but in doing so it may need to install additional packages which are not on your existing system.

The upgrade process preserves existing configuration files by renaming them using a `.rpmsave` extension (e.g., `sendmail.cf.rpmsave`) and leaves a log telling what actions it took in `/tmp/upgrade.log`. As software evolves, configuration file formats can change, so you should carefully compare your original configuration files to the new files before integrating your changes.

The next dialog you'll see is Figure 10–40, *Package Installation Status Dialog*. This dialog remains on the screen until the upgrade is complete.

10.9.2 Installing

If you choose a workstation- or server-class installation, Figure 10–13, *Automatic Partitioning Dialog* appears.

Figure 10–13 Automatic Partitioning Dialog

If you select **Continue** and press **OK**, the installation program partitions your disk and decides which software packages to install. Next, you'll see the **Hostname** dialog explained in Section 10.12, *Naming Your Computer*.

If you select **Manually partition** or choose to perform a custom-class installation, Disk Druid will begin. The disk partitioning dialogs described in the next section will appear, showing you any current partitions on your system. It is up to you to indicate the partitions and mount points to be used for installation of this version of Red Hat Linux.

CAUTION

Installing Red Hat Linux over another installation of Linux (including Red Hat Linux) does *not* preserve any information (files or data) from the prior installation. Make sure you save any important files! If you are worried about saving the current data on your existing Red Hat Linux system (without making a backup on your own), you should consider performing an upgrade instead.

10.10 Partitioning Your Disk for Red Hat Linux

If you have not yet planned how you will set up your partitions, turn to Appendix B, *An Introduction to Disk Partitions*. There you'll find an introduction to basic disk partitioning concepts. As a bare minimum, you'll need an appropriately-sized root partition, and a swap partition of at least 16 MB.

Figure 10–14, *Disk Setup Dialog* shows the two disk partitioning applications that are available for you to use.

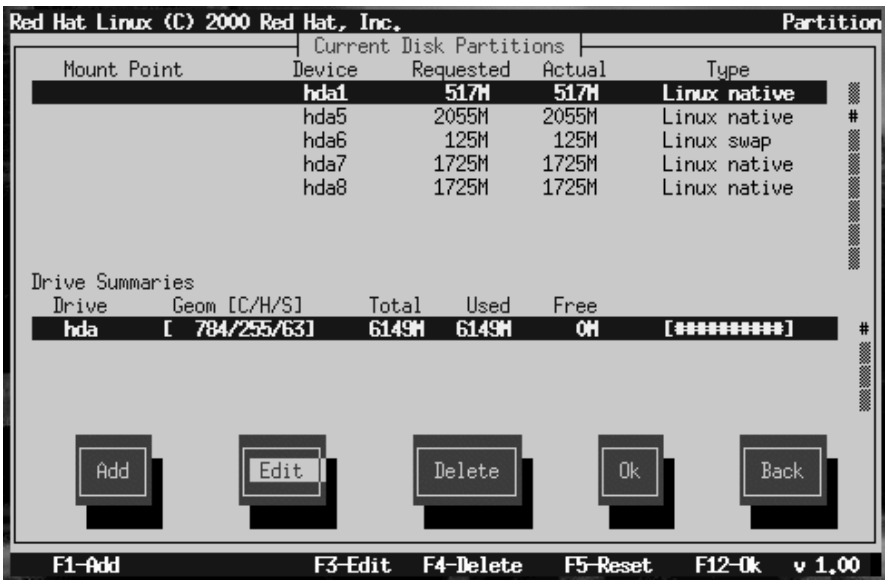
If you will be using fdisk to partition your drive, please see Section 10.10.11, *Using* fdisk for those instructions. If you select Disk Druid, continue reading below.

Figure 10–14 Disk Setup Dialog

The following sections describe the layout of Figure 10–15, *Disk Druid Main Screen* and how to use its buttons to set up partitions. If you're already familiar with Disk Druid, you can partition your disk and skip to Section 10.10.10, *Choose Partitions to Format Dialog*.

You use the disk partitioning dialogs to tell the installation program where to install Red Hat Linux (Figure 10–15, *Disk Druid Main Screen*).

Figure 10–15 Disk Druid Main Screen



10.10.1 The Current Disk Partitions Section

Each line in the **Current Disk Partitions** section represents a disk partition. In this example (Figure 10–15, *Disk Druid Main Screen*), there are Linux existing partitions.

Note the scroll bar to the right, which indicates that there may be more partitions than can be displayed at one time. If you use the [Up] and [Down] arrow keys, you can see if there are any additional partitions. Each line in this section has five different fields:

- **Mount Point** -- Indicates where the partition will be mounted when Red Hat Linux is installed (such as `/`, `/boot`, or `swap`).
- **Device** -- Displays specific hard drive and partition information.
- **Requested** -- Shows the partition's initial size.
- **Actual** -- Shows the partition's current size.
- **Type** -- Shows the partition's type.

As you scroll through the **Current Disk Partitions** section, you might see an `Unallocated Requested Partitions` title bar, followed by one or more partitions. These are partitions that have been requested but, for one reason or another, have not been allocated. A common reason for having an unallocated partition is a lack of sufficient free space for the partition. In any case, the reason the partition remains unallocated will be displayed after the partition's mount point.

10.10.2 The Drive Summaries Section

Each line in the **Drive Summaries** section represents a hard disk on your system. Each line has the following fields:

- **Drive** -- Shows the hard disk's device name.
- **Geom [C/H/S]** -- Shows the hard disk's **geometry**. The geometry consists of three numbers representing the number of cylinders, heads and sectors as reported by the hard disk.
- **Total** -- Shows the total available space on the hard disk.
- **Used** -- Shows how much of the hard disk's space is currently allocated to partitions.
- **Free** -- Shows how much of the hard disk's space is still unallocated.
- **Bar Graph** -- Presents a visual representation of the space currently used on the hard disk. The more pound signs there are between the square braces, the less free space there is. In Figure 10–15, *Disk Druid Main Screen*, the bar graph shows no free space.

Please Note

The **Drive Summaries** section is displayed only to indicate your computer's disk configuration. It is not meant to be used as a means of specifying the target hard drive for a given partition. This is described more completely in Section 10.10.5, *Adding a Partition*.

10.10.3 Disk Druid's Buttons

These buttons control Disk Druid's actions. They are used to add and delete partitions, and to change partition attributes. In addition, there are buttons that are used to accept the changes you've made, or to exit Disk Druid entirely. Let's take a look at each button in order.

- **Add** -- Request a new partition. Selecting this button causes a dialog box to appear containing fields that must be filled in.
- **Edit** -- Modify the mount point of the partition currently highlighted in the **Current Disk Partitions** section. Selecting this button will cause a dialog box to appear allowing you to change the name of the mount point.
- **Delete** -- Delete the partition currently highlighted in the **Current Disk Partitions** section. Selecting this button will cause a dialog box to appear asking you to confirm the deletion.
- **OK** -- Confirm that changes made to your system's partitions to be written to disk. You will be asked to confirm the changes before Disk Druid rewrites your hard disk partition table(s). In addition, any mount points you've defined are passed to the installation program, and will eventually be used by your Red Hat Linux system to define the filesystem layout.
- **Back** -- Abort without saving any changes you've made. When this button is selected, the installation program will take you back to the previous screen, so you can start over.

10.10.4 Handy Function Keys

Use the [F5] (Reset) function key to discard all changes you may have made while in Disk Druid, and return the list of partitions to those read from the partition table(s) on your hard disk(s). When selected, you'll be asked to confirm whether you want to discard the changes. Note that any mount points you've specified will be lost, and will need to be reentered.

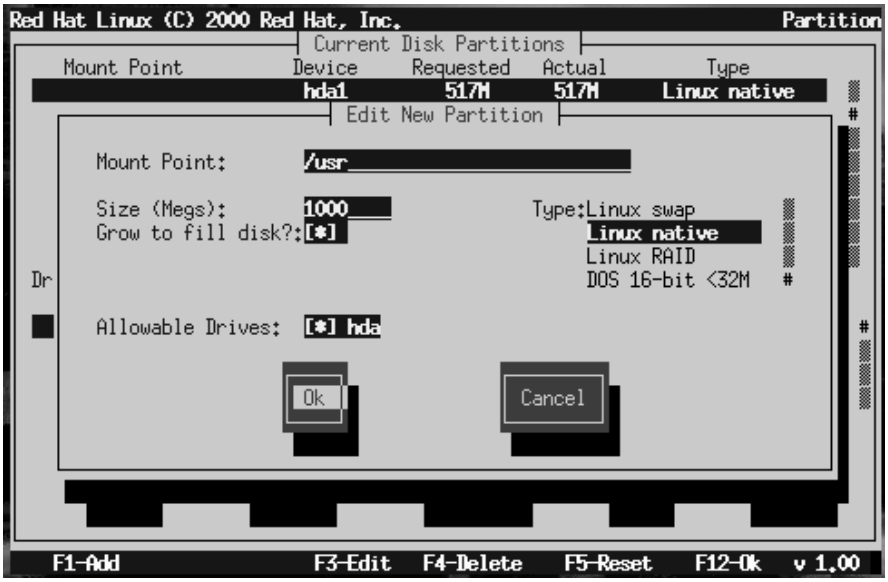
Please Note

You will need to dedicate at least one partition to Red Hat Linux, and optionally more. This is discussed more completely in Section B.1.8, *How Many Partitions?*.

10.10.5 Adding a Partition

To add a new partition, select the **Add** button and press [Space] or [Enter]. The **Edit New Partition** dialog (Figure 10–16, *Edit New Partition Dialog*) appears.

Figure 10–16 Edit New Partition Dialog



The screen contains the following fields:

- **Mount Point** -- Highlight this field and enter the partition's mount point. For example, if this partition should be the root partition, enter `/`; enter `/usr` for the `/usr` partition, and so on.

- **Size (Megs)** -- In this field, enter the size (in megabytes) of the partition. Note that this field starts with a "1" in it, meaning that unless you change it, you'll end up with a 1 MB partition. Delete it using the [Backspace] key, and enter the desired partition size.
- **Grow to fill disk?** -- This check box indicates whether the size you entered in the previous field is to be considered the partition's exact size, or its minimum size. Press [Space] to select this option. When selected, the partition will grow to fill all available space on the hard disk. In this case, the partition's size will expand and contract as other partitions are modified. If you make more than one partition growable, the partitions will compete for the available free space on the disk.
- **Type** -- This field contains a list of different partition types. Select the appropriate partition type by using the [Up] and [Down] arrow keys.
- **Allowable Drives** -- This field contains a list of the hard disks installed on your system, with a check box for each. If a hard disk's box is checked, then this partition may be created on that hard disk. By using different check box settings, you can direct Disk Druid to place partitions as you see fit, or let Disk Druid decide where partitions should go.
- **OK** -- Select this button and press [Space] when you are satisfied with the partition's settings, and wish to create it.
- **Cancel** -- Select this button and press [Space] when you don't want to create the partition.

10.10.6 Problems When Adding a Partition

Please Note

If you are having problems adding a partition, turn to Appendix B, *An Introduction to Disk Partitions*, *An Introduction to Disk Partitions*, to find a solution.

If you attempt to add a partition and Disk Druid can't carry out your request, you'll see a dialog box listing any partitions that are currently unallocated, along with the reason

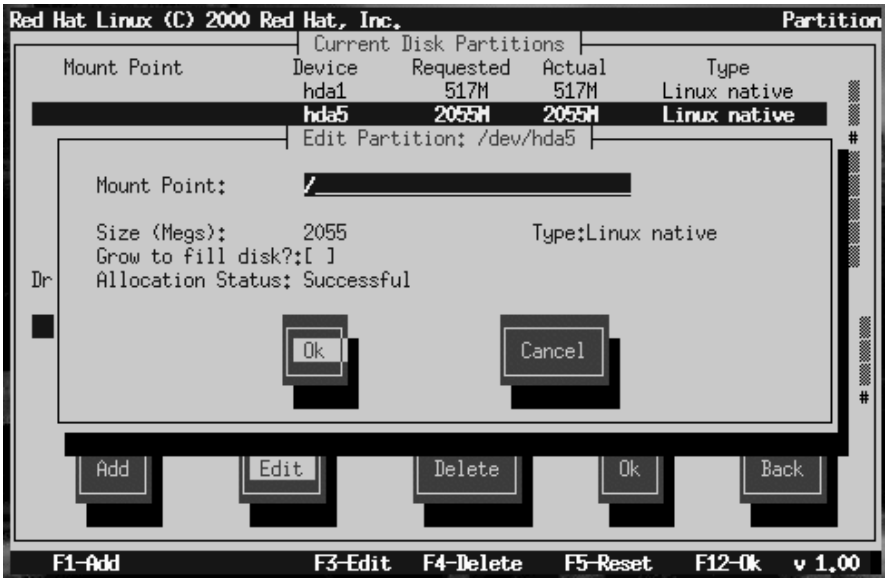
they could not be allocated, as in Figure 10–17, *Unallocated Partitions Dialog*. Select the **OK** button, and press [Space] to continue. Note that the unallocated partition(s) are also displayed on Disk Druid's main screen (though you may have to scroll the **Current Disk Partitions** section to see them).

Figure 10–17 Unallocated Partitions Dialog

10.10.7 Editing a Partition

To change a partition's mount point, highlight the partition in the **Current Disk Partitions** section, select the **Edit** button, and press [Space]. The **Edit Partition** dialog is similar to the one shown in Figure 10–16, *Edit New Partition Dialog*. The difference here is that all fields except the mount point are read-only. To modify any other value, delete the partition and add it again with the new values.

Figure 10–18 Edit Partition Dialog



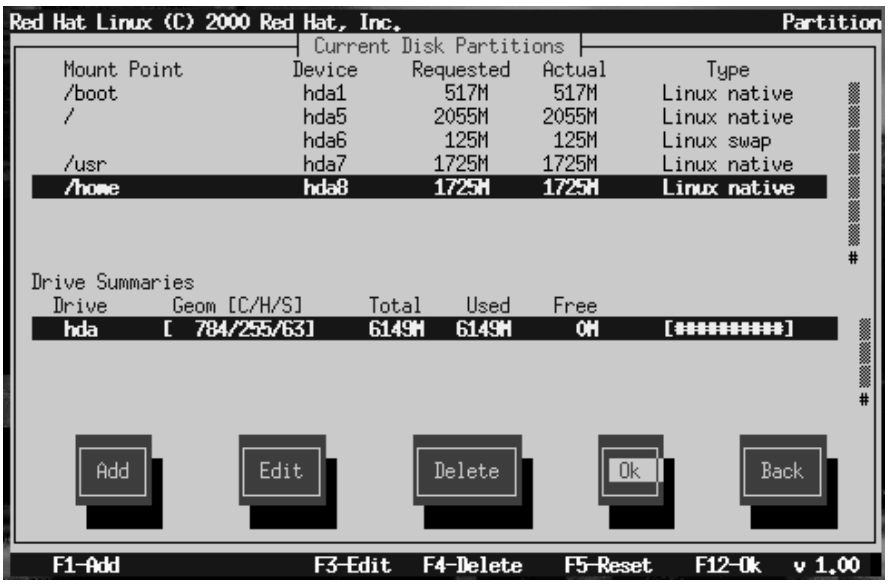
10.10.8 Deleting a Partition

To delete a partition, highlight the partition in the **Current Disk Partitions** section, select the **Delete** button, and press [Space]. You will be asked to confirm the deletion.

10.10.9 When You're Finished

Once you've configured your partitions and entered your mount points, you screen should look something like Figure 10–19, *Current Disk Partitions Dialog*.

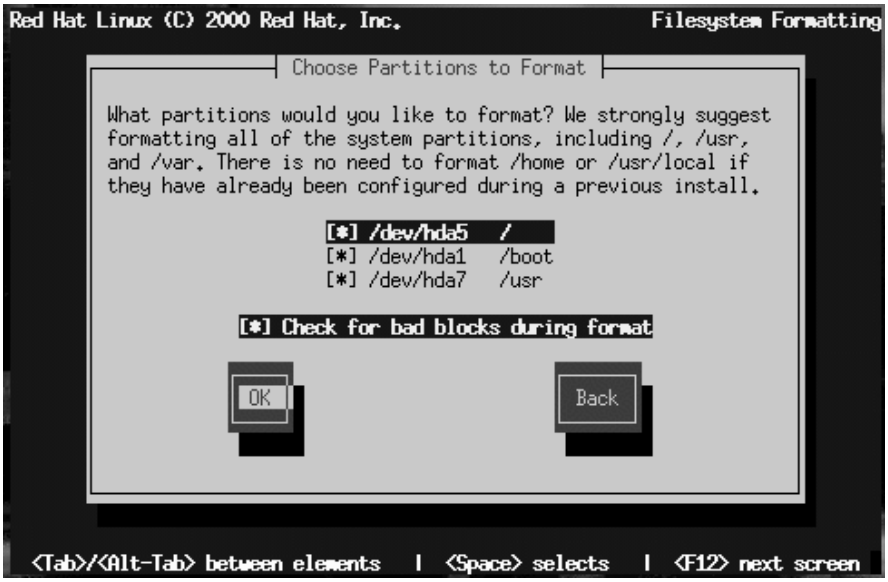
Figure 10–19 Current Disk Partitions Dialog



10.10.10 Choose Partitions to Format Dialog

Next, select which partitions you want to format (Figure 10–20, *Choose Partitions to Format*). You must format all newly created partitions and other partitions that contain old data (assuming they don't contain data you wish to keep).

Figure 10–20 Choose Partitions to Format



CAUTION

If partitions such as `/home` or `/usr/local` already exist and contain data you wish to keep, do not select these for formatting.

When you have selected the partitions to format, press [Space]. If you wish to check for bad blocks while formatting each filesystem (recommended for those with older disk drives), select **Check for bad blocks during format**. Select **OK**, and press [Space].

10.10.11 Using fdisk

If you chose a custom-class installation, you also chose which disk partitioning application to use. This section only applies if you opted to use fdisk.

Once you've selected fdisk, you'll be presented with the **Partition Disks** dialog box (Figure 10–21, *Disk Setup Dialog*). In this box is a list of every disk on your computer.

Using [Tab], and the [Up] and [Down] arrow keys, highlight the disk you'd like to partition, select **Edit**, and press [Space].

You will then enter fdisk and can partition the disk you selected. Repeat this process for each disk you want to partition. When you're finished, select **Done**.

Figure 10–21 Disk Setup Dialog

An Overview of fdisk

fdisk includes online help which is terse but useful. Here are a few tips:

- The command for help is m.
- To list the current partition table, use the p command (see Figure 10–22, *Sample Output from* fdisk).
- To add a new partition, use n.

- Linux fdisk creates partitions of type **Linux native** by default. When you create a swap partition, don't forget to change it to type **Linux swap** using the `t` command. The value for the **Linux swap** type is `82`. For other partition types, use the `l` command to see a list of partition types and values.
- Linux allows up to four (4) partitions on one disk. If you wish to create more than that, one (and only one) of the four may be an **extended** partition, which acts as a container for one or more **logical** partitions. Since it acts as a container, the extended partition must be at least as large as the total size of all the logical partitions it is to contain.
- It's a good idea to write down which partitions (e.g., `/dev/hda2`) are meant for which filesystems (e.g., `/usr`) as you create each one.

Please Note

None of the changes you make take effect until you save them and exit fdisk using the `w` command. You can quit fdisk at any time without saving changes by using the `q` command.

Figure 10–22 Sample Output from fdisk

```
This is the fdisk program for partitioning your drive. It is running
on /dev/hda.

Command (m for help): p

Disk /tmp/hda: 128 heads, 63 sectors, 620 cylinders
Units = cylinders of 8064 * 512 bytes

   Device Boot   Begin    Start      End   Blocks   Id  System
/tmp/hda1            1        1       21    84640+  83  Linux native
/tmp/hda2           22       22      148   512064   83  Linux native
/tmp/hda3          149      149      620  1903104    5  Extended
/tmp/hda5          149      149      275   512032+  83  Linux native
/tmp/hda6          276      276      402   512032+  83  Linux native
/tmp/hda7          403      403      419    68512+  82  Linux swap
/tmp/hda8          420      420      620   810400+  83  Linux native

Command (m for help):
```

Changing the Partition Table

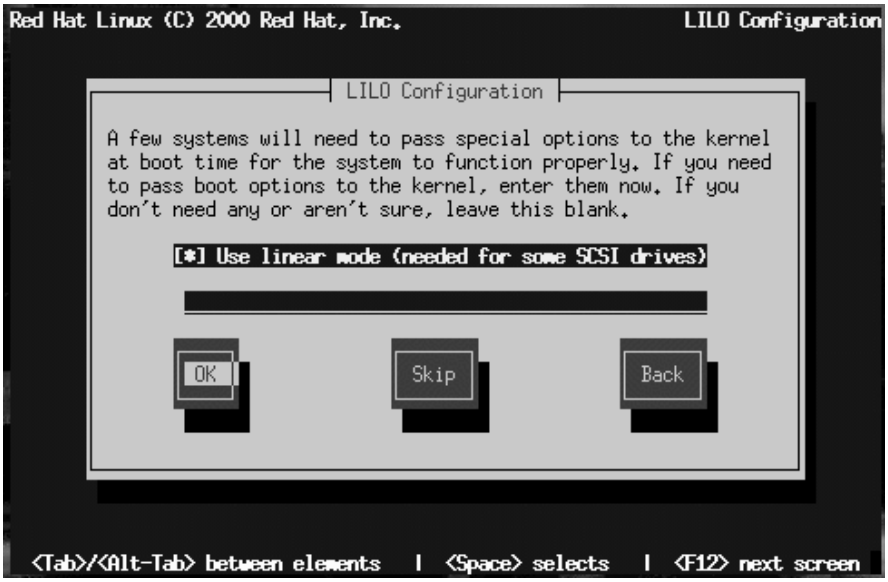
When you are finished partitioning your disks, press **Done**; you may see a message indicating that the installation program needs to reboot. This is a normal occurrence after changing a disk's partition data; it usually happens if you created, changed or deleted any extended partitions. After you press **OK**, your machine will reboot and the installation will begin again. Repeat the same installation steps you performed earlier until you reach the **Partition Disks** dialog; then simply choose **Done**.

10.11 Installing LILO

The `LI`nux `LO`ader (LILO) lets you specify at boot time whether to start Linux or another operating system. (If you are performing a workstation- or server-class installation, LILO is configured automatically in the Master Boot Record [MBR].) If you are performing a custom-class installation, the **LILO Installation** dialogs let you indicate how or whether to install LILO.

The **Choosing LILO in LILO Configuration Dialog** dialog (Figure 10–23, *Choosing LILO in LILO Configuration Dialog*) lets you add default options to the LILO boot command or choose to not install LILO at all. Any options you enter will be passed to the Linux kernel at boot time.

Figure 10–23 Choosing LILO in LILO Configuration Dialog



In Section 9.1.1, *Basic Hardware Configuration*, you were asked to review your computer's BIOS settings. In reviewing the BIOS settings, if you determined your system does not access a hard drive in linear mode, deselect this option. **Use linear mode** is selected by default. Select **OK** and press [Space] to continue.

If you do not wish to install LILO, press **Skip**.

WARNING

If you choose Skip, you will not be able to boot your Red Hat Linux system directly, and will need to use another boot method (such as a boot disk). Use this option only if you are sure you have another way of booting your Red Hat Linux system!

10.11.1 Alternatives to LILO

If you do not wish to use LILO to boot your Red Hat Linux system, there are a few alternatives:

Boot Disk

You can use the boot disk created by the installation program (if you elected to create one).

LOADLIN

LOADLIN can load Linux from MS-DOS; unfortunately, it requires a copy of the Linux kernel (and an initial RAM disk, if you have a SCSI adapter) to be available on an MS-DOS partition. The only way to accomplish this is to boot your Red Hat Linux system using some other method (e.g., from LILO on a diskette) and then copy the kernel to an MS-DOS partition. LOADLIN is available from ftp://metalab.unc.edu/pub/Linux/system/boot/dualboot/ and associated mirror sites.

SYSLINUX

SYSLINUX is an MS-DOS program very similar to LOADLIN; it is also available from ftp://metalab.unc.edu/pub/Linux/system/boot/loaders/ and associated mirror sites.

Commercial Bootloaders

Some commercial bootloaders are able to boot Linux. However, these products still require LILO to be installed in your Linux boot partition).

10.11.2 LILO Configuration

Choose where you want to install LILO (Figure 10–24, *Installing LILO in LILO Configuration Dialog*).

Figure 10–24 Installing LILO in LILO Configuration Dialog

You may install LILO in one of two places:

WARNING

To create a dual-boot environment on a system that currently has NT, you must install LILO on the first sector of the root partition, not the MBR. Please be sure to create a boot disk. In a case such as this, you will either need to use the boot disk, or configure the NT system loader to boot LILO from the first sector of the root partition. Be sure to check out http://www.linuxdoc.org/HOWTO/mini/Linux+NT-Loader.html for more information on setting up LILO and NT.

The Master Boot Record (MBR)

The recommended place to install LILO, unless the MBR already starts another OS loader, such System Commander or OS/2's Boot Manager. The MBR is a special area on your hard drive that is automatically loaded by your computer's BIOS, and is the earliest point at which LILO can take control of the boot process. If you install LILO in the MBR, when your machine boots, LILO will present a `boot:` prompt. You can then boot Red Hat Linux or any other operating system you configure LILO to boot.

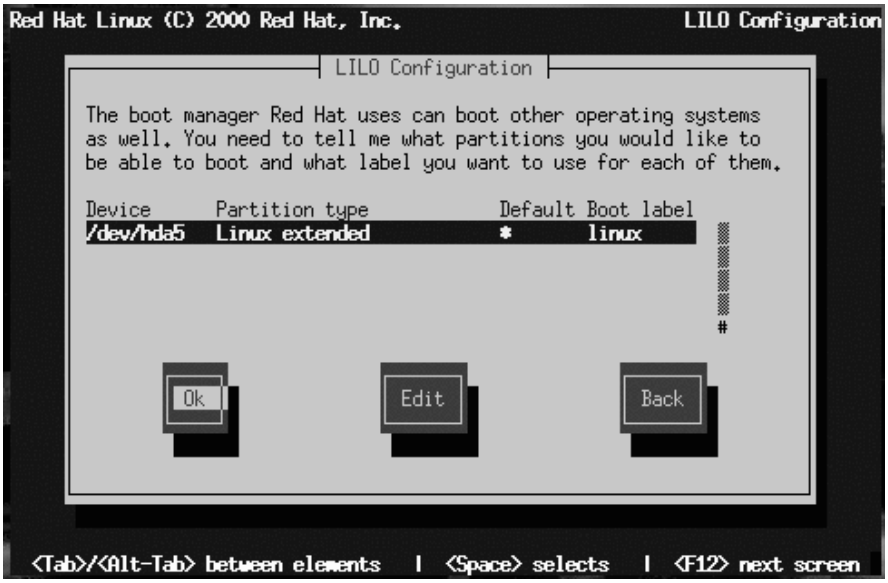
The first sector of your boot partition

Recommended if you are already using another boot loader on your system (such as OS/2's Boot Manager). In this case, your other boot loader will take control first. You can then configure that boot loader to start LILO (which will then boot Red Hat Linux).

Select the location where you wish to install LILO and press **OK**.

Finally the installation program lets you set the default operating system and specify boot labels, as in Figure 10–25, *Selecting Bootable Partitions in LILO Configuration Dialog*.

Figure 10–25 Selecting Bootable Partitions in LILO Configuration Dialog



Every partition that is bootable is listed, including partitions used by other operating systems. The **Boot label** column will be filled in with the word `linux` on the partition holding your Red Hat Linux system's root filesystem. Other partitions may also have boot labels (such as a `dos` boot label for Windows 95/98 partitions).

To add a boot label for a partition (or change an existing boot label), use the arrow keys to highlight the desired partition. Then use the [Tab] key to select the **Edit** button, and press [Space]. You'll then see a small dialog box permitting you to enter/modify the partition's boot label. Press **OK** when done.

Tip

The contents of the **Boot label** column show what you will need to enter at LILO's `boot:` prompt in order to boot the desired operating system. Should you forget the boot labels defined on your system, press [Tab] at LILO's `Boot:` prompt to display a list of defined boot labels.

There is also a column labeled **Default**. Only one partition will contain an asterisk under that column. The partition marked as the default will be the partition LILO will boot if there is no user input during the boot process. Initially the root partition for your Red Hat Linux installation will be selected as the default. If you'd like to change this, use the arrow keys to highlight the partition you'd like to make the default, and press [F2]. The asterisk will move to the selected partition. When you've finished, select **OK**, and press [Space].

10.11.3 SMP Motherboards and LILO

If the installer detects a symmetric multi-processor motherboard on your system, it will automatically create two `lilo.conf` entries: `linux`, the default, boots the system in SMP mode and `linux-up` boots the system in uni-processor mode.

10.12 Naming Your Computer

The **Hostname Configuration** dialog prompts you to enter a name for your computer (called a **hostname**).

Figure 10–26 Hostname Configuration Dialog

If you have a network card in your computer, enter a **fully-qualified domain name** in this format:

hostname.domain.name

In this example, *hostname* is the name you've chosen for your computer and *domain.name* is the TCP/IP domain. (A domain name may contain more nodes: for example, *eastcoast.mainserver.redhat.com* and *westcoast.mainserver.redhat.com*.)

Even if your computer is not part of a network, you can enter a hostname for your system. Take this opportunity to enter in a name; if you do not, your system will be known as `localhost`.

Tip

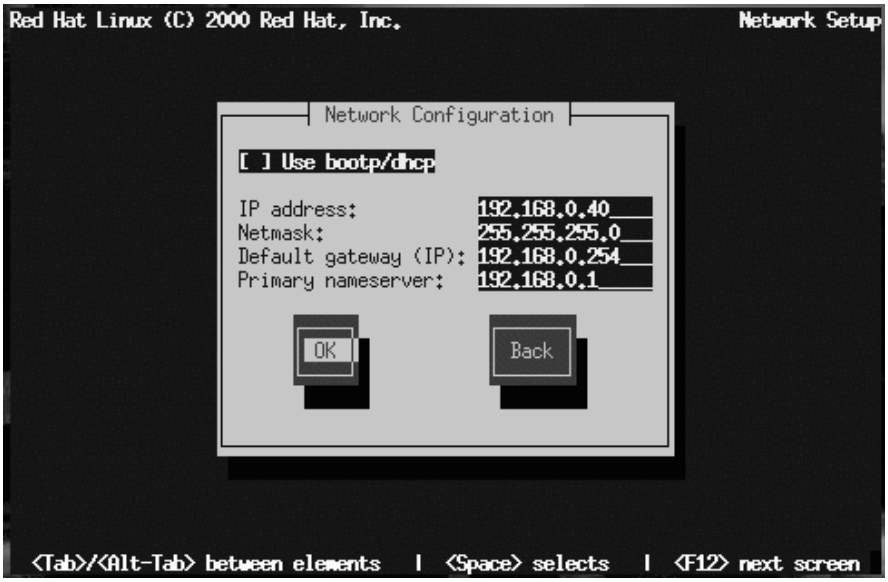
To change your hostname once you have rebooted your system, first become root. In a terminal at the root prompt, type `hostname` *`newname`*, where *newname* is what you want the hostname to be. If you just want to have the hostname echoed onto the display, type `hostname` and it will display the system's hostname.

10.13 Configuring a Network Connection

If you are installing over the network, you have performed your network configuration at the beginning of the installation process and do not need to complete this information again. If you are installing via local media and have a network card in place, please continue with this section.

Figure 10–27, *Network Configuration Dialog* appears only if your computer has a network card. (If there is more than one network card, this dialog configures the primary card.)

Figure 10–27 Network Configuration Dialog



You have two choices in this dialog:

- Select **Use bootp/dhcp**. In this case, an existing server on your LAN dynamically supplies network-related information needed to add this system to the network at connect-time. (Note that if you do this, the remaining fields in this dialog will be disabled, as DHCP and BOOTP essentially "fill in the blanks" for you.)
- Enter static network information in the fields provided. In this case, the network information you supply is assigned permanently to this computer.

10.14 Configuring Your Mouse

Next, the installation program probes for a mouse (Figure 10–28, *Mouse Selection Dialog*). Use the [Up] and [Down] arrow keys to confirm or change the selection as required.

Figure 10–28 Mouse Selection Dialog

If no mouse is detected, you'll have to select one manually.

To determine your mouse's interface, follow the mouse cable back to where it plugs into your system. If the connector at the end of the mouse cable plugs into a rectangular connector, you have a serial mouse. On the other hand, if the connector is round, you have a PS/2 mouse. If you are installing Red Hat Linux on a laptop computer, in most cases the pointing device will be PS/2 compatible.

If you cannot find an exact match, select one of the **Generic** entries, based on your mouse's number of buttons, and its interface.

The **Emulate 3 Buttons** check box allows you to emulate a three-button mouse if your mouse only has two buttons. If you select this check box, you can simulate the third, "middle" button by pressing both mouse buttons simultaneously. If you have a two-button mouse, checking this box is a good idea, since the X Window System is easiest to use with a three-button mouse.

If you've selected a mouse with a serial interface, highlight the appropriate serial port, select **OK**, and press [Space].

Tip

To change your mouse configuration after you have booted your Red Hat Linux system, become root and use the `/usr/sbin/mouseconfig` command.

To configure your mouse as a left-handed mouse after you have booted your Red Hat Linux system, open a terminal and type `gpm -B 321`.

10.15 Configuring the Time Zone

Next, enter your system's time zone (see Figure 10–29, *Time Zone Selection Dialog*).

Figure 10–29 Time Zone Selection Dialog

Notice that the screen shows the current time for your system. When you select different time zones, the current time also changes. Make sure that when you select your proper time zone, that the current time seen is also correct.

If you wish to set the hardware (CMOS) clock to GMT (Greenwich Mean Time, also known as UTC, or Universal Coordinated Time), select **Hardware clock set to GMT**. Setting it to GMT means your system will properly handle daylight-saving time, if your time zone uses it.

WARNING

If your computer uses another operating system, setting the clock to GMT may cause the other operating system to display the incorrect time. Also keep in mind that if more than one operating system is allowed to automatically change the time to compensate for daylight saving time, it is likely that the time will be improperly set.

Select your time zone from the list and press [Enter].

Tip

To change your time zone configuration after you have booted your Red Hat Linux system, use the `/usr/sbin/timeconfig` command.

10.16 Setting a Root Password

The **Root Password** dialog prompts you to set a **root password** for your system. You'll use the root password to log into your Red Hat Linux system to perform system administration functions.

Figure 10–30 Root Password Dialog

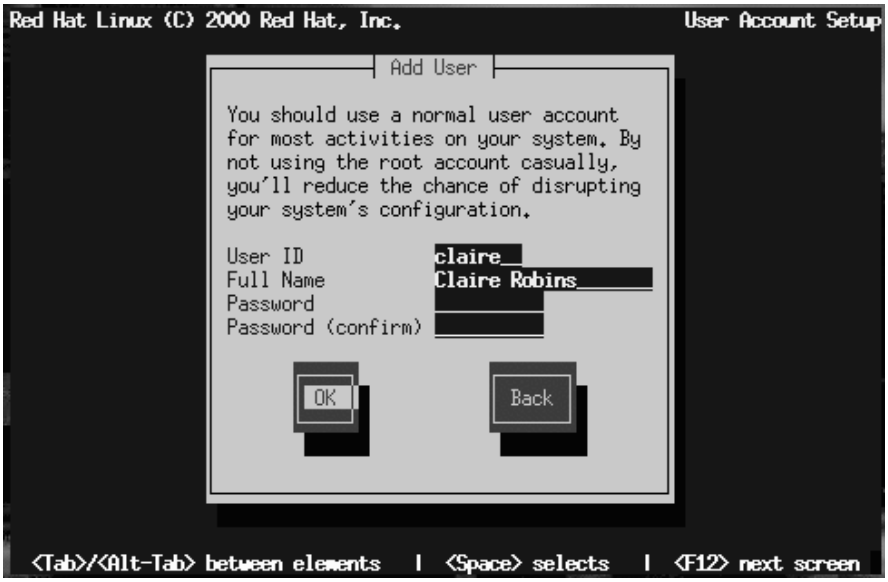
The root password must be at least six characters long; the password you type is not echoed to the screen. You must enter the password twice; if the two passwords do not match, the installation program will ask you to enter them again.

You should make the root password something you can remember, but not something that is easy for someone else to guess. Your name, your phone number, **qwerty**, **password**, **root**, **123456**, and **anteater** are all examples of poor passwords. Good passwords mix numerals with upper and lower case letters and do not contain dictionary words: **Aard387vark** or **420BMttNT**, for example. Remember that the password is case-sensitive. Write down this password and keep it in a secure place.

10.17 Creating a User Account

Next, you can create a user account for yourself which is for your everyday use (as in Figure 10–31, *Add User Dialog*). If you do not create a user account, after installation you'll have to log in as the **root** user (also known as the **superuser**). Root has complete access to the entire system. Logging in as the root user is best done *only* to perform system maintenance or administration. For instructions on how to create or modify user accounts after installation, refer to Chapter 3, *System Configuration* or the *Getting Started* chapter in the *Official Red Hat Linux Getting Started Guide*.

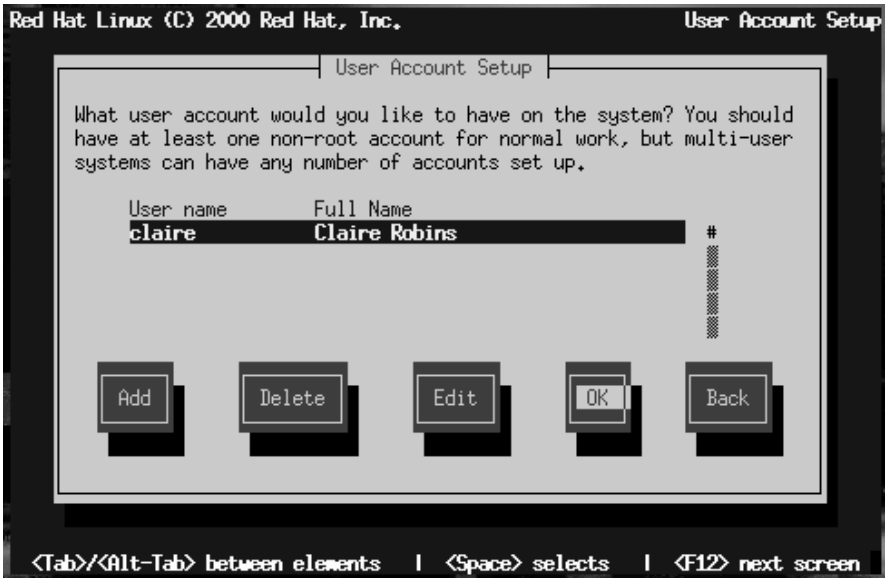
Figure 10–31 Add User Dialog



If you choose to create a user account, the account directory will be created under the path `/home` (for example, `/home/claire`). Passwords are case-sensitive and must contain at least six characters.

After you add a user account for yourself, the **User Account Setup** dialog (Figure 10–32, *User Account Setup Dialog*) appears, giving you the opportunity to create additional accounts. Select **Add** to do so or **OK** to continue.

Figure 10–32 User Account Setup Dialog



If you are performing a workstation- or server-class installation, your next task is to confirm your video hardware: turn to Section 10.20, *Configuring Your Video Adapter*. Otherwise, continue reading.

10.18 Authentication Configuration

If you are performing a custom installation, your next step is to configure the type of password authentication your Red Hat Linux system will use (see Figure 10–33, *Authentication Configuration Dialog*). You will also have the opportunity to configure NIS support; If you are unsure as to whether or not you should do this, ask your network administrator.

Figure 10–33 Authentication Configuration Dialog

Here's a brief explanation of the authentication password prompts:

- **Use Shadow Passwords** -- provides a very secure method of retaining passwords for you. The password filed in the `/etc/passwd` file is replaced by `/etc/shadow` which is readable only by root.
- **Enable MD5 Passwords** -- allows a long password to be used (up to 256 characters) instead of the standard eight letters or less.
- **Enable NIS** -- allows you to run a group of computers in the same Network Information Service domain with a common password and group file. There are two options here to choose from:
  - **NIS Domain** -- this option allows you to specify which domain or group of computers your system will belong to.

– **NIS Server** -- this option causes your computer to use a specific NIS server, rather than "broadcasting" a message to the local area network asking for any available server to host your system.

Please Note

To configure the NIS option, you must be connected to an NIS network. If you are unsure whether you are connected to an NIS network, please ask your network administrator.

Unless you are setting up NIS, you will notice that both shadow passwords and MD5 passwords are selected. We recommend you use both to make your machine as secure as possible.

10.19 Select Packages to Install

Figure 10–34, *Package Group Selection Dialog* appears only if you're performing a custom-class installation. Use this dialog to select the types of software packages you wish to install.

Figure 10–34 Package Group Selection Dialog

Selecting **Everything** (which can be found at the end of the component list) installs all packages included with Red Hat Linux. Selecting every package will require close to 1.7GB of free disk space.

10.19.1 Selecting Individual Packages

You can also select or deselect individual packages in an application category. To do this, check the **Select individual packages** check box. When you select **OK**, a screen like Figure 10–35, *Selecting Individual Packages in Package Group Selection Dialog* appears.

Figure 10–35 Selecting Individual Packages in Package Group Selection Dialog

To see the list of packages in the **Applications/Editors** category, for example, use the arrow keys to select this group and press [Enter] or [Space]. When you do, the **+** sign (closed) changes to a **-** sign (open) and a list of packages in this category appears. An ***** appears beside currently selected packages. Press the [Enter] or [Space] keys to select or deselect packages.

To see a description of a particular package, make sure that package is highlighted and press [F1].

Note that some packages that are required for every Red Hat Linux system (such as the kernel and certain libraries) do not appear in the package selection dialogs.

10.19.2 Unresolved Package Dependencies

Many software packages, in order to work correctly, require that other software packages are also installed on your system. For example, many of the graphical system administration tools require the `python` and `pythonlib` packages. To make sure

your system has all the packages it needs in order to be fully functional, Red Hat Linux checks these package **dependencies** each time you install or remove software packages.

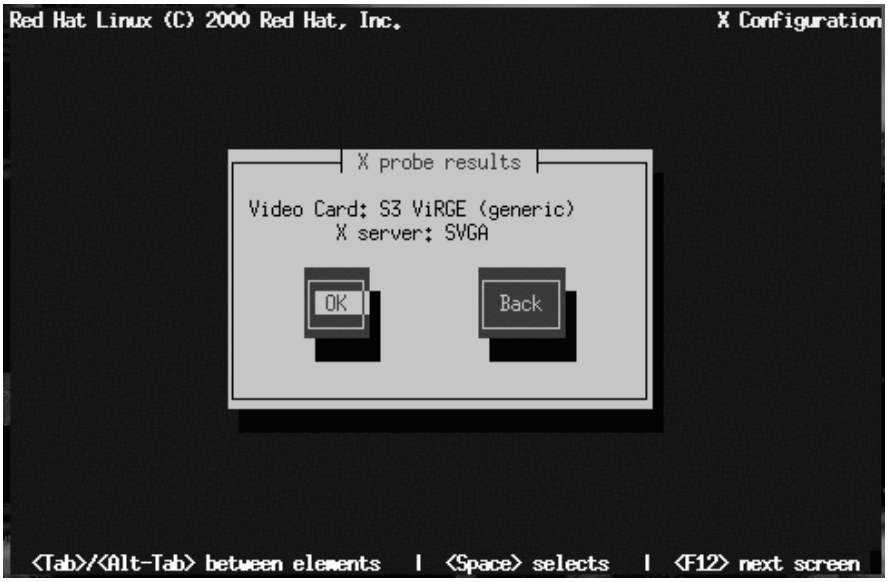
Figure 10–36 Package Dependencies Dialog

After you have finished selecting packages to install, the installation program checks the list of selected packages for dependencies. If any package requires another package which you have not selected to install, the program presents a list of these **unresolved dependencies** and gives you the opportunity to resolve them (see Figure 10–36, *Package Dependencies Dialog*). If you simply press **OK**, the program will resolve them automatically by adding all required packages to the list of selected packages.

10.20 Configuring Your Video Adapter

The installation program then probes for your video card. If your card is detected, click **OK** to continue.

Figure 10–37 Video Card Confirmation



If the installation program cannot detect your video adapter, you'll see Figure 10–38, *Video Card Selection Dialog*. In this case, select your video card from the list using the arrow keys and [Space].

Figure 10–38 Video Card Selection Dialog

10.21 Package Installation

Figure 10–39, *Installation to Begin Dialog* appears when the installation program is ready to format partitions and load software packages. The installation program names the log file (`/tmp/install.log`) for this Red Hat Linux installation. Select **OK** and press [Space] to continue.

Figure 10–39 Installation to Begin Dialog

While software packages are being installed, a screen like Figure 10–40, *Package Installation Status Dialog* appears.

Figure 10–40 Package Installation Status Dialog

As each software package is installed, the top part of the **Package Installation** screen displays its name, size, and a summary description of its function.

In the bottom part of the **Package Installation** screen, the **Total** row shows the total number of packages to be installed, the total size of all packages, and the amount of time required to install all packages. As each package begins and completes installation, the completed and remaining rows are updated.

If you are performing a server-class installation, you are almost finished; turn to Section 10.24, *Finishing Up*.

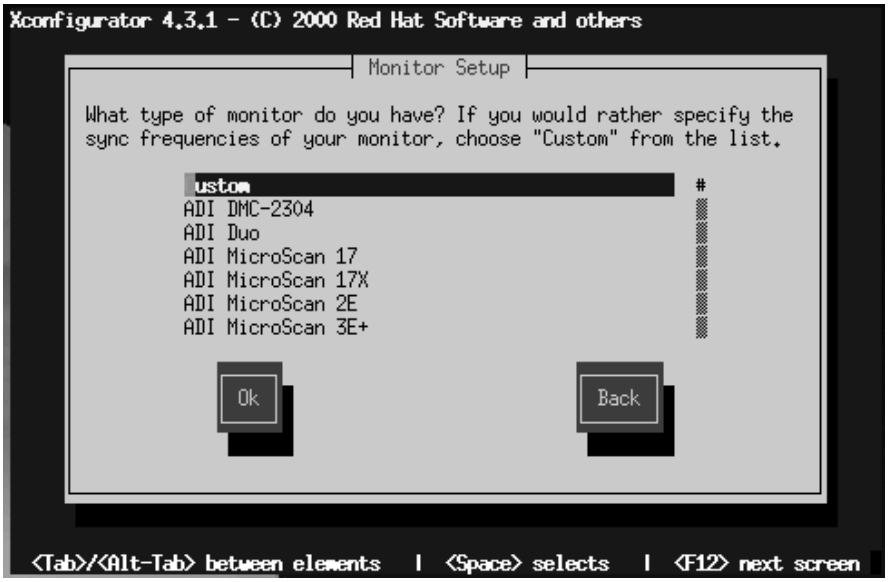
10.22 Configuring the X Window System

If you are performing a workstation- or a custom-class installation and you have chosen to install the X Window System, the `Xconfigurator` utility starts and gives you the opportunity to configure the X server for your system. If you are performing

a server-class installation or you did not choose to install the X Window System, skip ahead to Section 10.24, *Finishing Up*.

First, `Xconfigurator` presents a list of monitors (see Figure 10–41, *Monitor Setup Dialog*). If your monitor is listed, select it and press [Enter]. Otherwise, select **Custom**.

Figure 10–41 Monitor Setup Dialog



If you select **Custom**, `Xconfigurator` prompts you to select the horizontal sync range and vertical sync range of your monitor (these values are usually available in the documentation which accompanies your monitor, or from your monitor's vendor or manufacturer).

CAUTION

Do not select any range that exceeds the capacity of your monitor. If you do this, it is possible you may overclock your monitor and damage or destroy it.

After you've selected your monitor, the program starts to probe your video card. To bypass the probe and enter scan ranges and display modes manually, select **Do Not Probe**. This option is provided because an unsuccessful probe of your video card could cause your system not to boot properly or a program to hang.

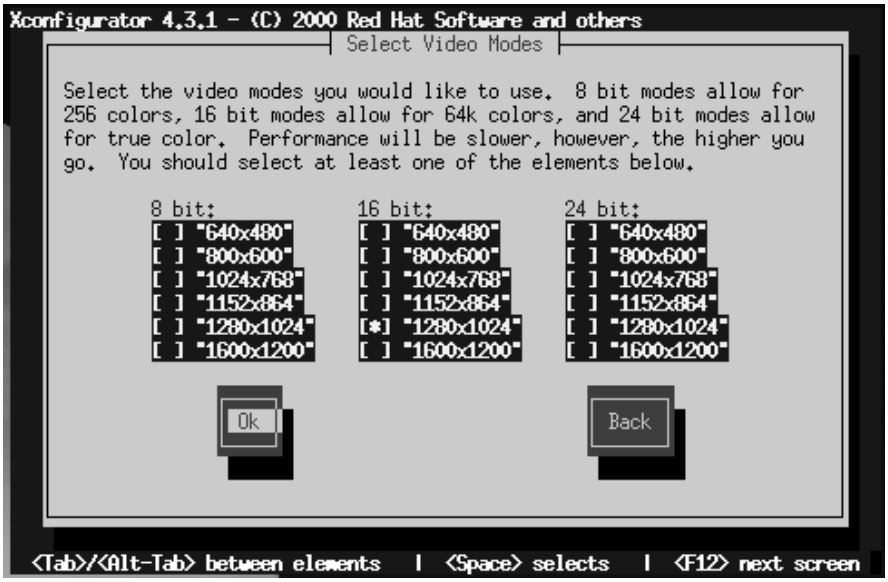
While the probe is in progress, it is normal for your display to blink.

When the probe is finished, Figure 10–42, *Probing Finished* appears, displaying the results of the probe.

Figure 10–42 Probing Finished

Select **Use Default** to accept the displayed color depth and resolution. If you select **Let Me Choose**, the **Select Video Modes** dialog (Figure 10–43, *Select Video Modes Dialog*) displays the video modes supported by your monitor.

Figure 10–43 Select Video Modes Dialog



Do not select a mode that exceeds the capacity of your monitor.

Once you've either selected the default video mode or specified a different mode, the **Starting X** dialog lets you test your X configuration.

Figure 10–44 Test Your X Configuration

If you select **OK**, you'll have the opportunity to go back enter different configuration values should there be an error. If there is no error, X will start, and a small display window will ask you if you can read its contents clearly: using the mouse, click **Yes** within ten seconds. Then you are asked whether you want X to start when the system boots.

In most cases, X configuration is complete at this point, and you'll see the completion screen.

To modify your X configuration after installation, as root issue the `Xconfigurator` command.

10.22.1 If Your Video Card Was Not Detected…

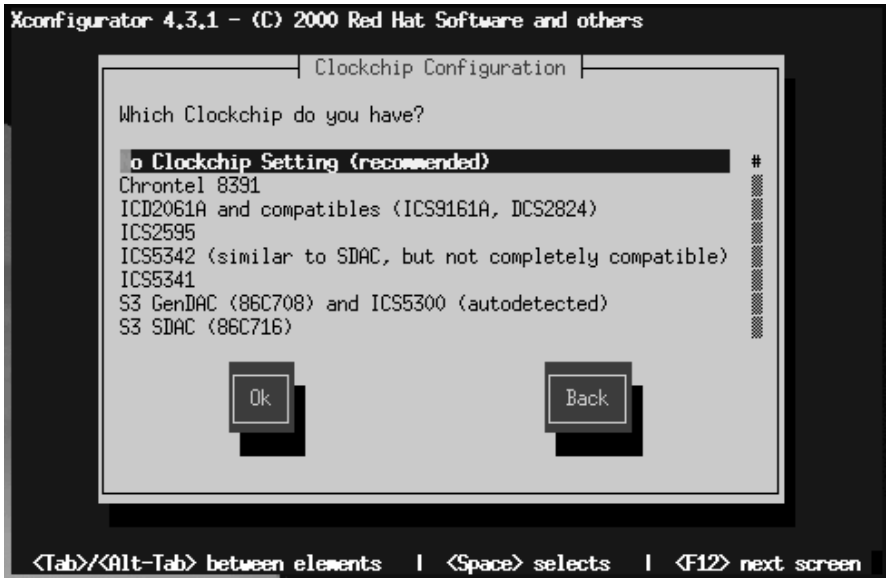
Earlier, in Section 10.20, *Configuring Your Video Adapter*, your system was probed for a video card. If none was detected, you were asked to select your card. If your card was not listed and you selected **Custom**, you are now asked for the following information about your video card:

- The amount of video memory installed on your video card. If you are not sure, consult the documentation accompanying your video card. Choosing more memory than is present in your card will not damage it, but may cause the XFree86 server to start incorrectly.

Figure 10–45 Select Video Memeory Dialog

- If your video card supports a video clockchip (if it has one). The recommended choice is **No Clockchip Setting**, since XFree86 can automatically detect the proper clockchip in most cases.

Figure 10–46 Clockship Configuration Dialog



- The preferred display mode (see Figure 10–43, *Select Video Modes Dialog*). Select one or more modes by pressing [Space].

`Xconfigurator` then saves all of your choices to the configuration file `/etc/X11/XF86Config`. To modify your X configuration after installation, as root issue the `Xconfigurator` command.

10.23 Creating a Boot Disk

If you're performing a custom-class installation, the **Bootdisk** dialog (see Figure 10–47, *Creating a Boot Disk*) gives you the opportunity to create a customized boot disk for your Red Hat Linux system.

Figure 10–47 Creating a Boot Disk

A boot disk can be handy in these situations:

- Use It Instead of LILO -- You can use a boot disk instead of LILO. This is handy if you're trying Red Hat Linux for the first time, and you'd feel more comfortable if the boot process for your other operating system is left unchanged. With a boot disk, going back to your other operating system is as easy as removing the boot disk and rebooting.
- Use It When Another Operating System Overwrites LILO -- Other operating systems may not be as flexible as Red Hat Linux when it comes to supported boot methods. Quite often, installing or updating another operating system can cause the master boot record (originally containing LILO) to be overwritten, making it impossible to boot your Red Hat Linux installation. The boot disk can then be used to boot Red Hat Linux so you can reinstall LILO.

Select **Yes** and press [Space] to create a boot disk. Next, you'll be prompted to insert a blank, formatted diskette.

Figure 10–48 Boot Disk Dialog

For instructions on how to create a boot disk after the installation, read the `mkbootdisk` man page. Be sure you create a new boot disk if you make any changes to your kernel.

10.24 Finishing Up

When finished, the installation program displays the **Complete** dialog (as in Figure 10–49, *Compete Dialog*), telling you to reboot your computer. Remove any diskette from the diskette drive (unless you skipped the LILO installation, in which case you must use the boot disk created during the installation), or the Red Hat Linux CD if your system booted from the CD-ROM.

Figure 10–49 Compete Dialog

Select **OK** to reboot your newly installed system. After your computer's normal power-up sequence has completed, you should see LILO's standard `boot:` prompt, at which you can do any of the following:

- Press [Enter] -- Causes LILO's default boot entry (as seen in Figure 10–25, *Selecting Bootable Partitions in LILO Configuration Dialog*) to be booted.
- Enter a Boot Label, followed by [Enter] -- Causes LILO to boot the operating system corresponding to the boot label.
- Wait -- After LILO's timeout period, (which, by default, is five seconds) LILO will automatically boot the default boot entry.

Do whatever is appropriate to boot Red Hat Linux. You should see one or more screens of messages scroll by. Eventually, you should see either a graphical login screen, or a `login:` prompt.

Congratulations! Your Red Hat Linux installation is complete!

Part III Appendixes

A General Parameters and Modules

This appendix is provided to illustrate *some* of the possible parameters that may be needed by certain drivers. It should be noted that, in most cases, these additional parameters are unnecessary. Also included is a list of network hardware and the associated modules required by that hardware.

Please keep in mind that if a device you are attempting to use requires one of these parameters, and support for that device is *not* compiled into the kernel, the traditional method of adding the parameter to the LILO boot command will not work. Drivers loaded as modules require that these parameters are specified when the module is loaded. The Red Hat Linux installation program gives you the option to specify module parameters when a driver is loaded.

A.1 A Note About Kernel Drivers

During installation of Red Hat Linux, there are some limits placed on the filesystems and other drivers supported by the kernel. However, after installation there is support for all filesystems available under Linux. At install time the modularized kernel has support for (E)IDE devices, (including ATAPI CD-ROM drives), SCSI adapters, and network cards. Additionally, all mice, SLIP, CSLIP, PPP, PLIP, FPU emulation, console selection, ELF, SysV IPC, IP forwarding, firewalling and accounting, reverse ARP, QIC tape and parallel printers, are supported.

Because Red Hat Linux supports installation on many different types of hardware, many drivers (including those for SCSI adapters, network cards, and many CD-ROMs) are not built into the Linux kernel used during installation; rather, they are available as **modules** and loaded as you need them during the installation process. If necessary, you will have the chance to specify options for these modules at the time they are loaded, and in fact these drivers will ignore any options you specify for them at the `boot:` prompt.

After the installation is complete you may want to rebuild a kernel that includes support for your specific hardware configuration. See Section 2.8, *Building a Custom Kernel* for information on building a customized kernel. Note that, in most cases, a custom-built kernel is not necessary.

A.2 CD-ROM Module Parameters

Please Note

Not all of the cards that are listed are supported. Please check the hardware compatibility list on Red Hat's World Wide Web site at http://www.redhat.com/support/docs/hardware.html to make sure your card is supported.

One of the more commonly used parameters, the `hdX=cdrom` parameter, *can* be entered at the boot prompt, as it deals with support for IDE/ATAPI CD-ROMs, which is part of the kernel.

In the tables below, most modules without any parameters listed are either able to auto-probe to find the hardware, or require you to manually change settings in the module source code, and recompile.

Table A–1 Hardware Parameters

Hardware	Module	Parameters
ATAPI/IDE CD-ROM Drives		hdX=cdrom
Aztech CD268-01A, Orchid CD-3110, Okano/Wearnes CDD110, Conrad TXC, CyCDROM CR520, CyCDROM CR540 (non-IDE)	`aztcd.o`	aztcd=io_port

Hardware	Module	Parameters
Sony CDU 31A or 33A CD-ROM	`cdu31a.o`	cdu31a=io_port,IRQ[,PAS] cdu31a_port=base_addr cdu31a_irq=irq
Philips/LMS CDROM drive 206 with cm260 host adapter card	`cm206.o`	cm206=io_port,IRQ
Goldstar R420 CD-ROM	`gscd.o`	gscd=io_port
ISP16, MAD16, or Mozart sound card CD-ROM interface (OPTi 82C928 and OPTi 82C929) with Sanyo/Panasonic, Sony, or Mitsumi drives	`isp16.o`	isp16=io_port, IRQ, dma,drive_type isp16_cdrom_base=io_port isp16_cdrom_irq=IRQ isp16_cdrom_dma=dma isp16_cdrom_type=drive_type
Mitsumi CD-ROM, Standard	`mcd.o`	mcd=io_port,IRQ
Mitsumi CD-ROM, Experimental	`mcdx.o`	mcdx=io_port_1, IRQ_1, io_port_n, IRQ_n
Optics storage 8000 AT CD-ROM "Dolphin" drive; Lasermate CR328A	`optcd.o`	optcd=io_port
SB Pro 16 compatible	`sbpcd.o`	sbpcd=io_port,sb_pro_Setting
Sanyo CDR-H94A	`sjcd.o`	sjcd=io_port sjcd_base=io_port
Sony CDU-535 & 531 (some Procomm drives)	`sonycd535.o`	sonycd535=io_port

Here are some examples of these modules in use:

Table A–2 Hardware Parameters Configuration Examples

Configuration	Example
ATAPI CD-ROM, jumpered as master on 2nd IDE channel	hdc=cdrom
non-IDE Mitsumi CD-ROM on port 340, IRQ 11	mcd=0x340,11
Three non-IDE Mitsumi CD-ROM drives using the experimental driver, io ports 300, 304, and 320 with IRQs 5, 10 and 11	mcdx=0x300,5,0x304,10,0x320,11
Sony CDU 31 or 33 at port 340, no IRQ	cdu31=0x340,0 cdu31_port=0x340 cdu31a_irq=0
Aztech CD-ROM at port 220	aztcd=0x220
Panasonic-type CD-ROM on a SoundBlaster interface at port 230	sbpcd=0x230,1
Phillips/LMS cm206 and cm260 at IO 340 and IRQ 11	cm206=0x340,11
Goldstar R420 at IO 300	gscd=0x300
Mitsumi drive on a MAD16 soundcard at IO Addr 330 and IRQ 1, probing DMA	isp16=0x330,11,0,Mitsumi
Sony CDU 531 at IO address 320	sonycd535=0x320

Please Note

Most newer Sound Blaster cards come with IDE interfaces. For these cards, you do not need to use `sbpcd` parameters, only use `hdx` parameters.

A.3 SCSI parameters

Table A–3 SCSI Parameters

Hardware	Module	Parameters
NCR53c810/820/720, NCR53c700/710/700-66	`53c7,8xx.o`	
AM53/79C974 PC-SCSI Driver Qlogic PCI-Basic	`AM53C974.o`	AM53C974=host-scsi-id, target-scsi-id,max-rate, max-offset
Most Buslogic (now Mylex) cards with "BT" part number	`BusLogic.o`	BusLogic_Options=option,option,... (See `README.BusLogic` in ...`drivers/scsi/`)
	`NCR53c406a.o`	ncr53c406a=io_port[, IRQ[, FASTPIO]] ncr53c406a io=io_port irq=IRQ fastpio=FASTPIO
Advansys SCSI Cards	`advansys.o`	
Adaptec AHA 152x	`aha152x.o`	aha152x=io_base, IRQ, scsi_id, reconnect, parity
Adaptec AHA 1542	`aha1542.o`	aha1542=io_base,buson,busoff,dmaspeed
Adaptec AHA 1740	`aha1740.o`	

Hardware	Module	Parameters
Adaptec AHA-274x, AHA-284x, AHA-29xx, AHA-394x, AHA-398x, AHA-274x, AHA-274xT, AHA-2842, AHA-2910B, AHA-2920C, AHA-2930/U/U2, AHA-2940/W/U/UW/AU/, U2W/U2/U2B/, U2BOEM, AHA-2944D/WD/UD/UWD, AHA-2950U2/W/B, AHA-3940/U/W/UW/, AUW/U2W/U2B, AHA-3950U2D, AHA-3985/U/W/UW, AIC-777x, AIC-785x, AIC-786x, AIC-787x, AIC-788x , AIC-789x, AIC-3860	`aic7xxx.o`	aic7xxx=string
Data Technology Corp DTC3180/3280	`dtc.o`	

Hardware	Module	Parameters
DTP SCSI host adapters (EATA/DMA) PM2011B/9X ISA, PM2021A/9X ISA, PM2012A, PM2012B, PM2022A/9X EISA, PM2122A/9X, PM2322A/9X, SmartRAID PM3021, PM3222, PM3224	`eata.o`	eata=port0,port1,port2,… options eata io_port=port0,port1,port2,… option=value
DTP SCSI Adapters PM2011, PM2021, PM2041, PM3021, PM2012B, PM2022, PM2122, PM2322, PM2042, PM3122, PM3222, PM3332, PM2024, PM2124, PM2044, PM2144, PM3224, PM3334	`eata_dma.o`	
DTP EATA-PIO boards	`eata_pio.o`	

Hardware	Module	Parameters
Future Domain TMC-16x0- based cards TMC-1800, TMC-18C50, TMC-18C30, TMC-36C70, Future Domain TMC-1650, TMC-1660, TMC-1670, TMC-1680, TMC-1610M/MER/MEX, TMC-3260 (PCI), Quantum ISA-200S, ISA-250MG, Adaptec AHA-2920A (PCI) (NOT AHA-2920C)	`fdomain.o`	fdomain=io_base, IRQ[, ADAPTER_ID]
NCR5380 and NCR53c400 cards	`g_NCR5380.o`	ncr5380=io_port,IRQ,dma ncr53c400=io_port,IRQ ncr5380 io=io_port irq=IRQ dma=dma ncr53c400 io=io_port irq=IRQ
GDT ISA/EISA/PCI Disk Array Controller	`gdth.o`	gdth=IRQ0,IRQ1,IRQ2,… options:values
IOMEGA MatchMaker parallel port SCSI adapter	`imm.o`	
Always IN2000 ISA SCSI card	`in2000.o`	in2000=setup_string:value in2000 setup_string=value
Initio INI-9X00U/UW SCSI host adapters	`initio.o`	
AMI MegaRAID 418, 428, 438, 466, 762	`megaraid.o`	

Hardware	Module	Parameters
NCR SCSI controllers with 810/810A/815/825/825A/860/875/876/895 chipsets	`ncr53c8xx.o`	ncr53c8xx=option1:value1,option2:value2,… ncr53c8xx="option1:value1 option2:value2…"
Pro Audio Spectrum/Studio 16	`pas16.o`	pas16=port,irq
IOMEGA PPA3 parallel port SCSI host adapter	`ppa.o`	
Perceptive Solutions PSI-240I EIDE	`psi240i.o`	
QLogic Fast SCSI FASXXX ISA/VLB/PCMCIA	`qlogicfas.o`	
QLogic ISP2100 SCSI-FCP	`qlogicfc.o`	
QLogic ISP1020 Intelligent SCSI cards IQ-PCI, IQ-PCI-10, IQ-PCI-D	`qlogicisp.o`	
Seagate ST01/ST02	`seagate.o`	controller_type=1 base_address=base_addr irq=irq
Future Domain TMC-885, TMC-950	`seagate.o`	controller_type=2 base_address=base_addr irq=irq

Hardware	Module	Parameters
Cards with the sym53c416 chipset	`sym53c416.o`	sym53c416=PORT-BASE[,IRQ] sym53c416 io=PORTBASE irq=IRQ
Trantor T128/T128F/T228 SCSI Host Adapter	`t128.o`	
Tekram DC390 and other AMD53C974A based PCI SCSI adapters	`tmscsim.o`	tmscsim=ID,SPEED
UltraStor 14F/34F SCSI host adapters (14F, 24F, 34F)	`u14-34f.o`	u14-34f=io_port1,io_port2,… io_port10 u14-34f io_port=io_port1,io_port2, … io_port10
UltraStor 14F, 24F, and 34F	`ultrastor.o`	
WD7000-FASST2,WD7000-ASC, WD7000-AX/MX/EX	`wd7000.o`	wd7000=IRQ,dma,io_port wd7000 io=io_port irq=IRQ dma=dma

Here are some examples of these modules in use:

Table A–4 SCSI Parameters Configuration Examples

Configuration	Example
Adaptec AHA1522 at port 330, IRQ 11, SCSI ID 7	aha152x=0x330,11,7
Adaptec AHA1542 at port 330	bases=0x330
Future Domain TMC-800 at CA000, IRQ 10	controller_type=2 base_address=0xca000 irq=10

When a parameter has commas, make sure you do *not* put a space after a comma.

A.4 Ethernet parameters

Table A–5 Ethernet Module Parameters

Hardware	Module	Parameters
3Com 3c501	`3c501.o`	3c501=io_port,IRQ
3Com 3c503 and 3c503/16	`3c503.o`	3c503=io_port,IRQ 3c503 io=io_port_1,io_port_n irq=IRQ_1,IRQ_n
3Com EtherLink Plus (3c505)	`3c505.o`	3c505=io_port,IRQ,DMA 3c505 io=io_port_1,io_port_n irq=IRQ_1,IRQ_2 dma=dma_1,dma_n
3Com EtherLink 16	`3c507.o`	3c507=io_port,IRQ 3c507 io=io_port irq=IRQ
3Com EtherLink III	`3c509.o`	3c509=IRQ
3Com ISA EtherLink XL "Corkscrew"	`3c515.o`	
3Com EtherLink PCI III/XL Vortex (3c590, 3c592, 3c595, 3c597) Boomerang (3c900, 3c905, 3c595)	`3c59x.o`	
Apricot 680x0 VME, 82596 chipset	`82596.o`	82596=IRQ 82596 irq=IRQ
Ansel Communications AC3200 EISA	`ac3200.o`	ac3200=io_port,IRQ ac3200 io=io_port_1,io_port_n irq=IRQ_1,IRQ_n

Hardware	Module	Parameters
Alteon AceNIC Gigabit Ethernet driver	`acenic.o`	acenic=trace,link acenic trace=trace link=val
Allied Telesis AT1700	`at1700.o`	at1700=io_port,IRQ at1700 io=io_port irq=IRQ
Tangent ATB-II, Novel NL-10000, Daystar Digital LT-200, Dayna DL2000, DaynaTalk PC (HL), COPS LT-95, Farallon PhoneNET PC II, III	`cops.o`	cops=io_port,IRQ cops io=io_port irq=IRQ
Modular driver for the COSA or SRP synchronous serial card	`cosa.o`	cosa=io_port,IRQ,dma
Crystal LAN CS8900/CS8920	`cs89x0.o`	cs89x0=io_port,IRQ,MEDIA_TYPE cs89x0 io=io_port irq=IRQ media=TYPE
EtherWORKS DE425 TP/COAX EISA, DE434 TP PCI, DE435/450 TP/COAX/AUI PCI DE500 10/100 PCI Kingston, LinkSys, SMC8432, SMC9332, Znyx31[45], and Znyx346 10/100 cards with DC21040 (no SROM), DC21041[A], DC21140[A], DC21142, DC21143 chipsets	`de4x5.o`	de4x5=io_port de4x5 io=io_port de4x5 args='ethX[fdx] autosense=MEDIA_STRING'

Hardware	Module	Parameters
D-Link DE-600 Ethernet Pocket Adapter	`de600.o`	
D-Link DE-620 Ethernet Pocket Adapter	`de620.o`	de620 io=io_port irq=IRQ bnc=1 utp=1
DIGITAL DEPCA & EtherWORKS DEPCA, DE100, DE101, DE200 Turbo, DE201Turbo DE202 Turbo TP/BNC, DE210, DE422 EISA	`depca.o`	depca=io_port,IRQ depca io=io_port irq=IRQ
Digi Intl. RightSwitch SE-X EISA and PCI	`dgrs.o`	
Cabletron E2100 series ethercards	`e2100.o`	e2100=io_port,IRQ e2100 io=io_port irq=IRQ
Intel i82595 ISA EtherExpressPro10/10+ driver	`eepro.o`	eepro=io_port,IRQ,mem eepro io=io_port irq=IRQ mem=mem
Intel i82557/i82558 PCI EtherExpressPro driver	`eepro100.o`	
Intel EtherExpress 16 (i82586)	`eexpress.o`	eexpress=io_port,IRQ eexpress io=io_port irq=IRQ
SMC EtherPower II 9432 PCI (83c170/175 EPIC series)	`epic100.o`	
Racal-Interlan ES3210 EISA Network Adapter	`es3210.o`	es3210=io_port,IRQ,mem es3210 io=io_port irq=IRQ mem=mem

Hardware	Module	Parameters
ICL EtherTeam 16i/32	`eth16i.o`	eth16i=io_port,mediatype eth16i ioaddr=io_port mediatype=type
EtherWORKS 3 (DE203, DE204 and DE205)	`ewrk3.o`	ewrk=io_port,IRQ ewrk io=io_port irq=IRQ
Fujitsu FMV-181/182/183/184	`fmv18x.o`	fmv18x=io_port,IRQ fmv18x io=io_port irq=IRQ
Modular driver for the Comtrol Hostess SV11	`hostess_sv11.o`	hostess_sv11=io_port, IRQ, DMABIT hostess_sv11 io=io_port irq=IRQ dma=DMABIT
HP PCLAN/plus	`hp-plus.o`	hp-plus=io_port,IRQ hp-plus io=io_port irq=IRQ
HP LAN Ethernet	`hp.o`	hp=io_port,IRQ hp io=io_port irq=IRQ
100VG-AnyLan Network Adapters HP J2585B, J2585A, J2970, J2973, J2573 Compex ReadyLink ENET100-VG4, FreedomLine 100/VG	`hp100.o`	hp100=io_port,name hp100 hp100_port=io_port hp100_name=name
IBM Token Ring 16/4	`ibmtr.o`	ibmtr=io_port,IRQ,mem ibmtr io=io_port irq=IRQ mem=mem

Hardware	Module	Parameters
AMD LANCE/PCnet Allied Telesis AT1500, HP J2405A, NE2100, NE2500	`lance.o`	lance=io_port,IRQ,dma lance io=io_port_1,io_port_n irq=IRQ_1,IRQ_2 dma=dma_1,dma_n
Mylex LNE390 EISA cards (LNE390A, LNE390B)	`lne390.o`	lne390=io_port,IRQ,mem lne390 io=io_port irq=IRQ mem=mem
	`ltpc.o`	ltpc=io_port,IRQ ltpc io=io_port irq=IRQ
NE1000 / NE2000 (non-pci)	`ne.o`	ne=io_port,IRQ ne io=io_port irq=IRQ
PCI NE2000 cards RealTEk RTL-8029, Winbond 89C940, Compex RL2000, KTI ET32P2, NetVin, NV5000SC, Via 82C926, SureCom NE34	`ne2k-pci.o`	
Novell NE3210 EISA Network Adapter	`ne3210.o`	ne3210=io_port,IRQ,mem ne3210 io=io_port irq=IRQ mem=mem
MiCom-Interlan NI5010 ethercard	`ni5010.o`	ni5010=io_port,IRQ ni5010 io=io_port irq=IRQ
NI5210 card (i82586 Ethernet chip)	`ni52.o`	ni52=io_port,IRQ ni52 io=io_port irq=IRQ
NI6510, ni6510 EtherBlaster	`ni65.o`	ni65=io_port,IRQ,dma ni65 io=io_port irq=IRQ dma=dma

Hardware	Module	Parameters
AMD PCnet32 and AMD PCnetPCI	`pcnet32.o`	
RedCreek Communications PCI	`rcpci.o`	
RealTek cards using RTL8129 or RTL8139 Fast Ethernet chipsets	`rtl8139.o`	
Sangoma S502/S508 multi-protocol FR	`sdla.o`	
Sangoma S502A, ES502A, S502E, S503, S507, S508, S509	`sdladrv.o`	
SysKonnect Token Ring ISA/PCI Adapter, TR4/16(+) ISA or PCI, TR4/16 PCI, and older SK NET TR4/16 ISA cards	`sktr.o`	sktr=io_port,IRQ,mem sktr io=io_port irq=IRQ mem=mem
SMC Ultra and SMC EtherEZ ISA ethercard (8K, 83c790)	`smc-ultra.o`	smc-ultra=io_port,IRQ smc-ultra io=io_port irq=IRQ
SMC Ultra32 EISA Ethernet card (32K)	`smc-ultra32.o`	
SMC 9000 series of Ethernet cards	`smc9194.o`	smc9194=io_port,IRQ smc9194 io=io_port irq=IRQ ifport=[0,1,2]

Hardware	Module	Parameters
Compaq Netelligent 10/100 TX PCI UTP Compaq Netelligent 10 T PCI UTP Compaq Integrated NetFlex 3/P Compaq Netelligent Dual 10/100 TX PCI UTP Compaq Netelligent Integrated 10/100 TX UTP Compaq Netelligent 10/100 TX Embedded UTP Compaq Netelligent 10 T/2 PCI UTP/Coax Compaq Netelligent 10/100 TX UTP Compaq NetFlex 3/P Olicom OC-2325, OC-2183, OC-2326	`tlan.o`	tlan=io_port,IRQ,aui,debug tlan io=io_port irq=IRQ Other Module Options: speed=10Mbs,100Mbs debug=0x0[1,2,4,8] aui=1 duplex=[1,2]
Digital 21x4x Tulip PCI Ethernet cards SMC EtherPower 10 PCI(8432T/8432BT) SMC EtherPower 10/100 PCI(9332DST) DEC EtherWorks 100/10 PCI(DE500-XA) DEC EtherWorks 10 PCI(DE450) DEC QSILVER's, Znyx 312 etherarray Allied Telesis LA100PCI-T Danpex EN-9400, Cogent EM110	`tulip.o`	

Hardware	Module	Parameters
VIA Rhine PCI Fast Ethernet cards with either the VIA VT86c100A Rhine-II PCI or 3043 Rhine-I D-Link DFE-930-TX PCI 10/100	`via-rhine.o`	
AT&T GIS (nee NCR) WaveLan ISA Card	`wavelan.o`	wave-lan=[IRQ,0],io_port,NWID
WD8003 and WD8013 "compatible" ethercards	`wd.o`	wd=io_port, IRQ, mem,mem_end wd io=io_port irq=IRQ mem=mem mem_end=end
Packet Engines Yellowfin	`yellowfin.o`	
G-NIC PCI Gigabit Ethernet adapter		
Z8530 based HDLC cards for AX.25	`z85230.o`	

Here are some examples of these modules in use:

Table A–6 Ethernet Parameter Configuration Examples

Configuration	Example
NE2000 ISA card at IO address 300 and IRQ 11	ne=0x300,11 ether=0x300,11,eth0
Wavelan card at IO 390, autoprobe for IRQ, and use the NWID to 0x4321	wavelan=0,0x390,0x4321 ether=0,0x390,0x4321,eth0

A.4.1 Using Multiple Ethernet Cards

You can use multiple Ethernet cards in one machine. If each card uses a different driver (e.g., a 3c509 and a DE425), you simply need to add `alias` (and possibly `options`) lines for each card to `/etc/conf.modules`. See Section 3.2.2, *Loading Kernel Modules* for more information.

If any two Ethernet cards use the same driver (e.g., two 3c509's or a 3c595 and a 3c905), you will need to either give the two card addresses on the driver's options line (in the case of ISA cards), or (for PCI cards) simply add one `alias` line for each card.

http://metalab.unc.edu/LDP/HOWTO/Ethernet-HOWTO.html For more information about using more than one Ethernet card, see the *Linux Ethernet-HOWTO* at http://metalab.unc.edu/LDP/HOWTO/Ethernet-HOWTO.html. If you installed the `howto` package when you installed Red Hat Linux, you can find it in the file `/usr/doc/HOWTO/Ethernet-HOWTO`.

B An Introduction to Disk Partitions

Disk partitions are a standard part of the personal computer landscape, and have been for quite some time. However, with so many people purchasing computers featuring preinstalled operating systems, relatively few people understand how partitions work. This chapter attempts to explain how disk partitions work so you'll find your Red Hat Linux installation is as simple as possible.

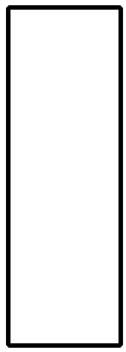
If you're reasonably comfortable with disk partitions, you could skip ahead to Section B.1.4, *Making Room For Red Hat Linux* for more information on the process of freeing up disk space to prepare for a Red Hat Linux installation. This section also discusses the partition naming scheme used by Linux systems, sharing disk space with other operating systems, and related topics.

B.1 Hard Disk Basic Concepts

Hard disks perform a very simple function -- they store data and reliably retrieve it on command.

When discussing issues such as disk partitioning, it's important to know a bit about the underlying hardware; unfortunately, it's easy to become bogged down in details. Therefore, let's use a simplified diagram of a disk drive to help us explain what goes on "under the hood." Figure B–1, *An Unused Disk Drive* shows a brand-new, unused disk drive.

Figure B–1 An Unused Disk Drive

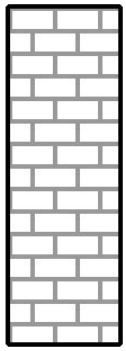


Not much to look at, is it? But if we're talking about disk drives on a basic level, it will do. Let's say that we'd like to store some data on this drive. As things stand now, it won't work. There's something we need to do first…

B.1.1 It's Not What You Write, it's How You Write It

The old-timers in the audience probably got this one on the first try. We need to **format** the drive. Formatting (usually known as "making a **filesystem**" in Linux parlance) writes information to the drive, creating order out of the empty space in an unformatted drive.

Figure B–2 Disk Drive with a Filesystem



As Figure B–2, *Disk Drive with a Filesystem* implies, the order imposed by a filesystem involves some tradeoffs:

- A small percentage of the drive's available space is used to store filesystem-related data, and can be considered as overhead.
- A filesystem splits the remaining space into small, consistently-sized segments. In the Linux world, these segments are known as **blocks**. [1]

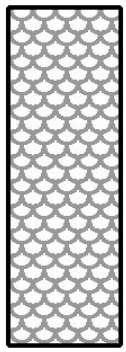
Given that filesystems make things like directories and files possible, these tradeoffs are usually seen as a small price to pay.

It's also worth noting that there is no single, universal filesystem; as Figure B–3, *Disk Drive with a Different Filesystem* shows, a disk drive may have one of many different filesystems written on it. As you might guess, different filesystems tend

[1] Blocks really *are* consistently sized, unlike our illustrations. Keep in mind, also, that an average disk drive contains thousands of blocks. But for the purposes of this discussion, please ignore these minor discrepancies.

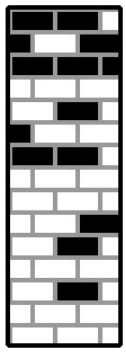
to be incompatible; that is, an operating system that supports one filesystem (or a handful of related filesystem types) may not support another. This last statement is not a hard-and-fast rule, however. For example, Red Hat Linux supports a wide variety of filesystems (including many commonly used by other operating systems), making data interchange easy.

Figure B–3 Disk Drive with a Different Filesystem



Of course, writing a filesystem to disk is only the beginning. The goal of this process is to actually *store* and *retrieve* data. Let's take a look at our drive after some files have been written to it.

Figure B–4 Disk Drive with Data Written to It



As Figure B–4, *Disk Drive with Data Written to It* shows, 14 of the previously-empty blocks are now holding data. We cannot determine how many files reside on this drive; it may be as few as one or as many as 14, as all files use at least one block. Another important point to note is that the used blocks do not have to form a contiguous region; used and unused blocks may be interspersed. This is known as **fragmentation**. Fragmentation can play a part when attempting to resize an existing partition.

As with most computer-related technologies, disk drives continued to change over time. In particular, they changed in one specific way -- they got bigger. Not bigger in size, but bigger in capacity. And it was this additional capacity that drove a change in the way disk drives were used.

B.1.2 Partitions – Turning One Drive Into Many

As disk drive capacities soared, some people started wondering if having all that space in one big chunk wasn't such a great idea. This line of thinking was driven by several

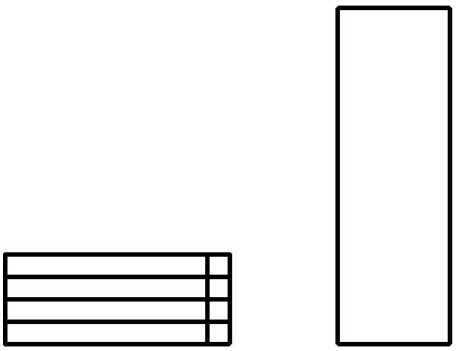
issues, some philosophical, some technical. On the philosophical side, above a certain size, it seemed that the additional space provided by a larger drive created more clutter. On the technical side, some filesystems were never designed to support larger drives. Or the filesystems *could* support larger drives, but the overhead imposed by the filesystem became excessive.

The solution to this problem was to divide disks into **partitions**. Each partition can be accessed as if it was a separate disk. This is done through the addition of a **partition table**.

Please Note

While the diagrams in this chapter show the partition table as being separate from the actual disk drive, this is not entirely accurate. In reality, the partition table is stored at the very start of the disk, before any filesystem or user data. But for clarity, we'll keep it separate in our diagrams.

Figure B–5 Disk Drive with Partition Table



As Figure B–5, *Disk Drive with Partition Table* shows, the partition table is divided into four sections. Each section can hold the information necessary to define a single partition, meaning that the partition table can define no more than four partitions.

Each partition table entry contains several important characteristics of the partition:

- The points on the disk where the partition starts and ends;
- Whether the partition is "active";
- The partition’s type.

Let’s take a closer look at each of these characteristics. The starting and ending points actually define the partition’s size and location on the disk. The "active" flag is used by some operating systems’ boot loaders. In other words, the operating system in the partition that is marked "active" will be booted.

The partition’s type can be a bit confusing. The type is a number that identifies the partition’s anticipated usage. If that statement sounds a bit vague, that’s because the

meaning of the partition type is a bit vague. Some operating systems use the partition type to denote a specific filesystem type, to flag the partition as being associated with a particular operating system, to indicate that the partition contains a bootable operating system, or some combination of the three.

Table B–1, *Partition Types* contains a listing of some popular (and obscure) partition types, along with their numeric values.

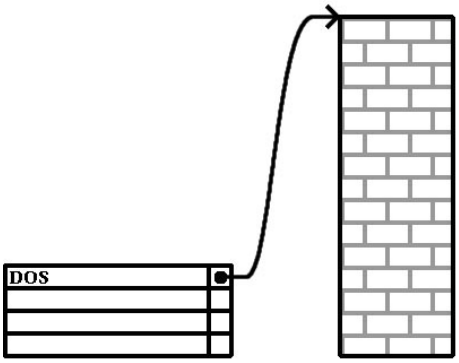
Table B–1 Partition Types

Partition Type	Value	Partition Type	Value
Empty	00	Novell Netware 386	65
DOS 12-bit FAT	01	PIC/IX	75
XENIX root	02	Old MINIX	80
XENIX usr	03	Linux/MINUX	81
DOS 16-bit <=32M	04	Linux swap	82
Extended	05	Linux native	83
DOS 16-bit >=32	06	Linux extended	85
OS/2 HPFS	07	Amoeba	93
AIX	08	Amoeba BBT	94
AIX bootable	09	BSD/386	a5
OS/2 Boot Manager	0a	OpenBSD	a6
Win95 FAT32	0b	NEXTSTEP	a7
Win95 FAT32 (LBA)	0c	BSDI fs	b7
Win95 FAT16 (LBA)	0e	BSDI swap	b8
Win95 Extended (LBA)	0f	Syrinx	c7
Venix 80286	40	CP/M	db
Novell?	51	DOS access	e1

Partition Type	Value	Partition Type	Value
Microport	52	DOS R/O	e3
GNU HURD	63	DOS secondary	f2
Novell Netware 286	64	BBT	ff

Now you might be wondering how all this additional complexity is normally used. See Figure B–6, *Disk Drive With Single Partition* for an example.

Figure B–6 Disk Drive With Single Partition



That’s right -- in many cases there is but a single partition spanning the entire disk, essentially duplicating the pre-partitioned days of yore. The partition table has only one entry used, and it points to the start of the partition.

We’ve labeled this partition as being of type "DOS," although as you can see from Table B–1, *Partition Types*, that’s a bit simplistic, but adequate for the purposes of this discussion. This is a typical partition layout for most newly purchased computers with some version of Windows pre-installed.

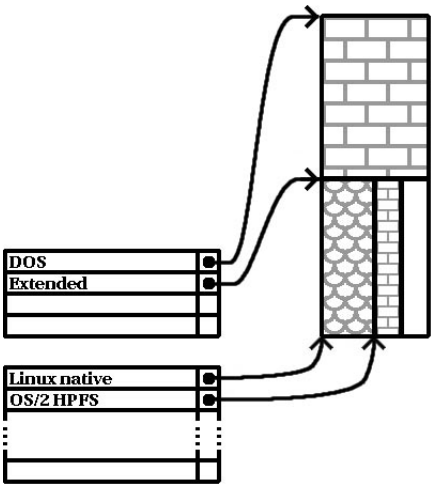
B.1.3 Partitions within Partitions – An Overview of Extended Partitions

Of course, in time it became obvious that four partitions would not be enough. As disk drives continued to grow, it became more and more likely that a person could configure four reasonably-sized partitions and still have disk space left over. There needed to be some way of creating more partitions.

Enter the extended partition. As you may have noticed in Table B–1, *Partition Types*, there is an "Extended" partition type; it is this partition type that is at the heart of extended partitions. Here's how it works.

When a partition is created and its type is set to "Extended," an extended partition table is created. In essence, the extended partition is like a disk drive in its own right -- it has a partition table that points to one or more partitions (now called **logical partitions**, as opposed to the four **primary partitions**) contained entirely within the extended partition itself. Figure B–7, *Disk Drive With Extended Partition* shows a disk drive with one primary partition, and one extended partition containing two logical partitions (along with some unpartitioned free space).

Figure B–7 Disk Drive With Extended Partition



As this figure implies, there is a difference between primary and logical partitions -- there can only be four primary partitions, but there is no fixed limit to the number of logical partitions that can exist. (However, in reality it is probably not a good idea to try to define and use more than 12 logical partitions on a single disk drive.)

Now that we've discussed partitions in general, let's see how to use this knowledge to get Red Hat Linux installed.

B.1.4 Making Room For Red Hat Linux

There are three possible scenarios you may face when attempting to repartition your hard disk:

- Unpartitioned free space is available.
- An unused partition is available.
- Free space in an actively used partition is available.

Let's look at each scenario in order.

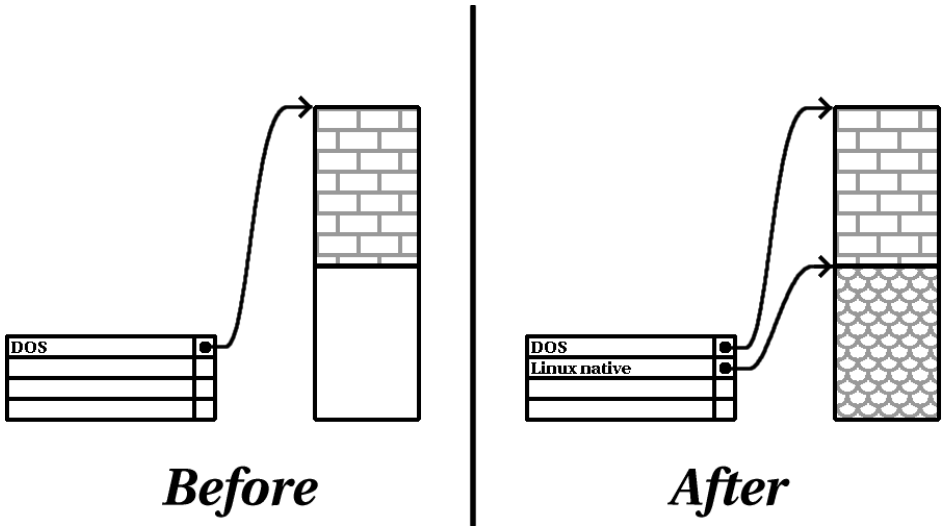
Please Note

Please keep in mind that the following illustrations are simplified in the interest of clarity, and do not reflect the exact partition layout that you will encounter when actually installing Red Hat Linux.

Using Unpartitioned Free Space

In this situation, the partitions already defined do not span the entire hard disk, leaving unallocated space that is not part of any defined partition. Figure B–8, *Disk Drive with Unpartitioned Free Space* shows what this might look like.

Figure B–8 Disk Drive with Unpartitioned Free Space



If you think about it, an unused hard disk also falls into this category; the only difference is that *all* the space is not part of any defined partition.

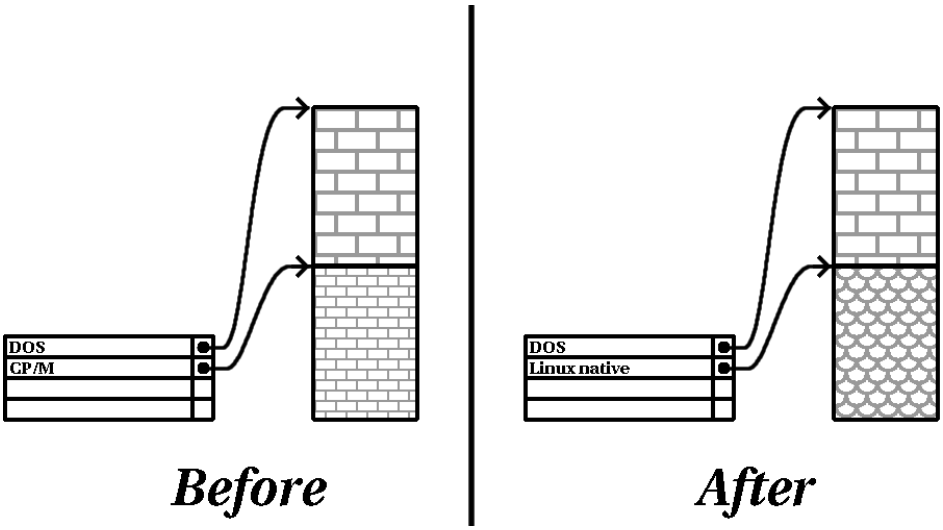
In any case, you can simply create the necessary partitions from the unused space. Unfortunately, this scenario, although very simple, is not very likely (unless you've just purchased a new disk just for Red Hat Linux).

Let's move on to a slightly more common situation.

Using Space From An Unused Partition

In this case, maybe you have one or more partitions that you just don't use any longer. Perhaps you've dabbled with another operating system in the past, and the partition(s) you've dedicated to it never seem to be used anymore. Figure B–9, *Disk Drive With an Unused Partition* illustrates such a situation.

Figure B–9 Disk Drive With an Unused Partition



If you find yourself in this situation, you can use the space allocated to the unused partition. You'll first need to delete the partition, and then create the appropriate Linux partition(s) in its place. You can either delete the partition using DOS `fdisk`, or you'll be given the opportunity to do so during a custom-class installation.

Using Free Space From An Active Partition

This is the most common situation. It is also, unfortunately, the hardest to work with. The main problem is that, even if you have enough free space, it's presently allocated to a partition that is in use. If you purchased a computer with pre-installed software, the hard disk most likely has one massive partition holding the operating system and data.

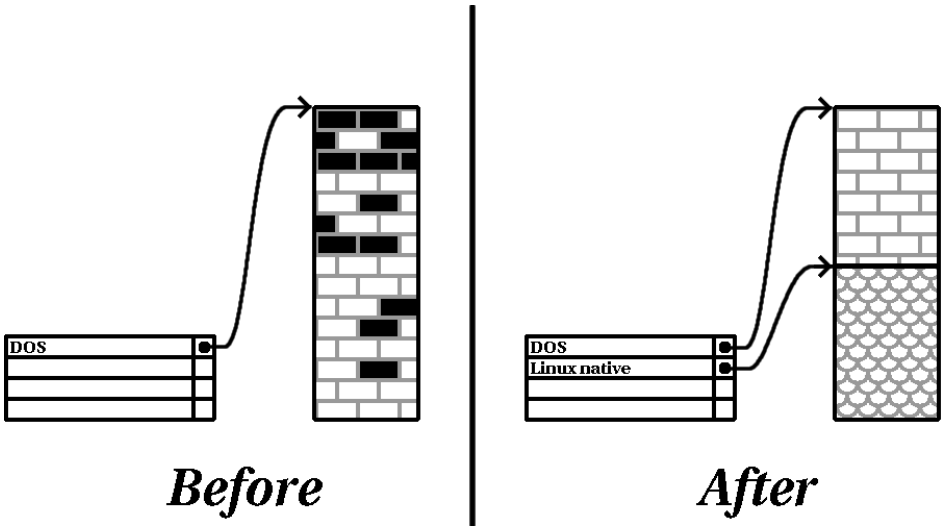
Aside from adding a new hard drive to your system, you have two choices:

Destructive Repartitioning

Basically, you delete the single large partition, and create several smaller ones. As you might imagine, any data you had in the original partition is destroyed. This means that making a complete backup is necessary. For your own sake, make two backups, use verification (if available in your backup software), and try to read data from your backup *before* you delete the partition. Note also that if there was an operating system of some type installed on that partition, it will need to be reinstalled as well.

After creating a smaller partition for your existing software, you can reinstall any software, restore your data, and continue with your Red Hat Linux installation. Figure B–10, *Disk Drive Being Destructively Repartitioned* shows this being done.

Figure B–10 Disk Drive Being Destructively Repartitioned



CAUTION

As Figure B–10, *Disk Drive Being Destructively Repartitioned* shows, any data present in the original partition will be lost without proper backup!

Non-Destructive Repartitioning

Here, you run a program that does the seemingly impossible: it makes a big partition smaller without losing any of the files stored in that partition. Many people have found this method to be reliable and trouble-free. What software should you use to perform this feat? There are several disk management software products on the market; you'll have to do some research to find the one that is best for your situation.

While the process of non-destructive repartitioning is rather straightforward, there are a number of steps involved:

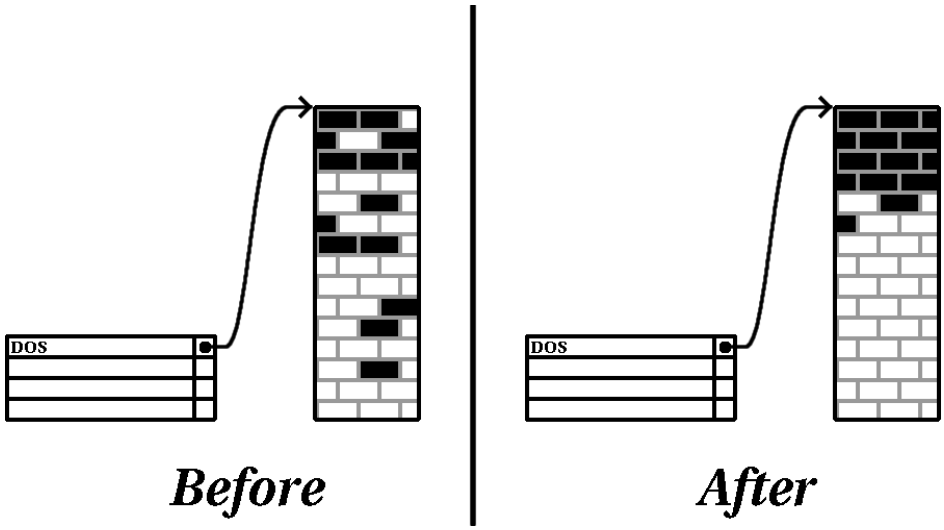
- Compress existing data
- Resize partition
- Create new partition(s)

Let’s take a look at each step in a bit more detail.

Compress existing data

As Figure B–11, *Disk Drive Being Compressed* shows, the first step is to compress the data in your existing partition. The reason for doing this is to rearrange the data such that it maximizes the available free space at the "end" of the partition.

Figure B–11 Disk Drive Being Compressed

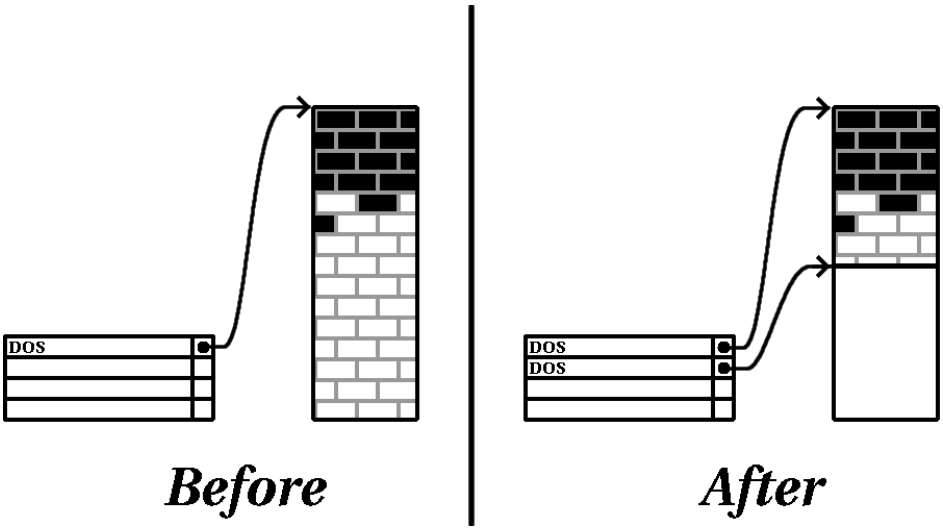


This step is crucial; without it, it is possible that the location of your data could prevent the partition from being resized to the extent desired. Note also that, for one reason or another, some data cannot be moved. If this is the case (and it restricts the size of your new partition(s)), you may be forced to destructively repartition your disk.

Resize partition

Figure B–12, *Disk Drive with Partition Resized* shows the actual resizing process. While the actual end-product of the resizing operation varies depending on the software used, in most cases the newly freed space is used to create an unformatted partition of the same type as the original partition.

Figure B–12 Disk Drive with Partition Resized

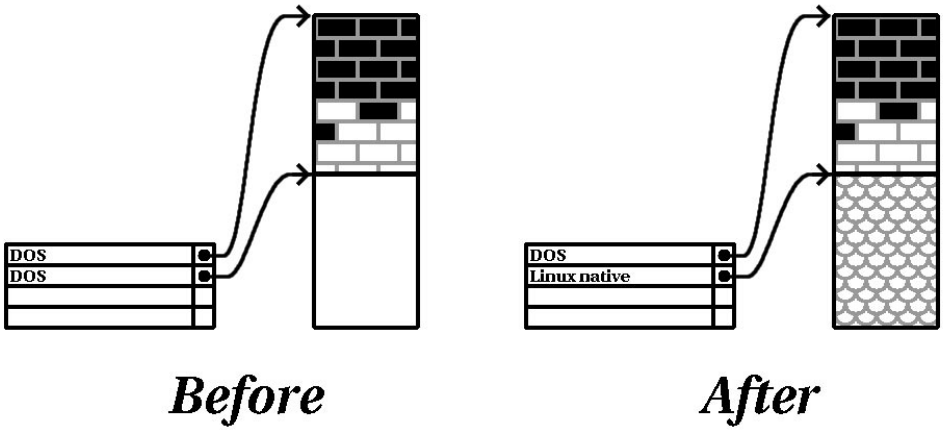


It's important to understand what the resizing software you use does with the newly freed space, so that you can take the appropriate steps. In the case we've illustrated, it would be best to simply delete the new DOS partition, and create the appropriate Linux partition(s).

Create new partition(s)

As the previous step implied, it may or may not be necessary to create new partitions. However, unless your resizing software is Linux-aware, it is likely you'll need to delete the partition that was created during the resizing process. Figure B–13, *Disk Drive with Final Partition Configuration* shows this being done.

Figure B–13 Disk Drive with Final Partition Configuration



Intel

The following information is specific to Intel-based computers only.

As a convenience to our customers, we provide the `fips` utility. This is a freely available program that can resize FAT (File Allocation Table) partitions. It's included on the Red Hat Linux/Intel CD-ROM in the `dosutils` directory.

Please Note

Many people have successfully used `fips` to repartition their hard drives. However, because of the nature of the operations carried out by `fips`, and the wide variety of hardware and software configurations under which it must run, Red Hat cannot guarantee that `fips` will work properly on your system. Therefore, no installation support whatsoever is available for `fips`; use it at your own risk.

That said, if you decide to repartition your hard drive with `fips`, it is *vital* that you do two things:

- Perform a Backup -- Make two copies of all the important data on your computer. These copies should be to removable media (such as tape or diskettes), and you should make sure they are readable before proceeding.
- Read the Documentation -- Completely read the `fips` documentation, located in the `/dosutils/fipsdocs` subdirectory on Red Hat Linux/Intel CD 1.

Should you decide to use `fips`, be aware that after `fips` runs you will be left with *two* partitions: the one you resized, and the one `fips` created out of the newly freed space. If your goal is to use that space to install Red Hat Linux, you should delete the newly created partition, either by using `fdisk` under your current operating system, or while setting up partitions during a custom-class installation.

B.1.5 Partition Naming Scheme

Linux refers to disk partitions using a combination of letters and numbers which may be confusing, particularly if you’re used to the "C drive" way of referring to hard disks and their partitions. In the DOS/Windows world, here is how partitions are named:

- Each partition’s type is checked to determine if it can be read by DOS/Windows.
- If the partition’s type is compatible, it is assigned a "drive letter." The drive letters start with "C".

- The drive letter can then be used to refer to that partition as well as the filesystem contained on that partition.

Red Hat Linux uses a naming scheme that is more flexible and conveys more information than the approach used by other operating systems. The naming scheme is file-based, with filenames in the form:

`/dev/xxyN`

Here's how to decipher the partition naming scheme:

`/dev/`

This string is the name of the directory in which all device files reside. Since partitions reside on hard disks, and hard disks are devices, the files representing all possible partitions reside in `/dev/`.

xx

The first two letters of the partition name indicate the type of device on which the partition resides. You'll normally see either `hd` (for IDE disks), or `sd` (for SCSI disks).

y

This letter indicates which device the partition is on. For example, `/dev/hda` (the first IDE hard disk) or `/dev/sdb` (the second SCSI disk).

N

The final number denotes the partition. The first four (primary or extended) partitions are numbered `1` through `4`. Logical partitions start at `5`. E.g., `/dev/hda3` is the third primary or extended partition on the first IDE hard disk; `/dev/sdb6` is the second logical partition on the second SCSI hard disk.

Please Note

There is no part of this naming convention that is based on partition type; unlike DOS/Windows, *all* partitions can be identified under Red Hat Linux. Of course, this doesn't mean that Red Hat Linux can access data on every type of partition, but in many cases it is possible to access data on a partition dedicated to another operating system.

Keep this information in mind; it will make things easier to understand when you're setting up the partitions Red Hat Linux requires.

B.1.6 Disk Partitions and Other Operating Systems

If your Red Hat Linux partitions will be sharing a hard disk with partitions used by other operating systems, most of the time you'll have no problems. However, there are certain combinations of Linux and other operating systems that require extra care. Information on creating disk partitions compatible with other operating systems is available in several HOWTOs and Mini-HOWTOs, available on the Red Hat Linux CD in the `doc/HOWTO` and `doc/HOWTO/mini` directories. In particular, the Mini-HOWTOs whose names start with `Linux+` are quite helpful.

Intel

If Red Hat Linux/Intel will coexist on your machine with OS/2 , you must create your disk partitions with the OS/2 partitioning software---otherwise, OS/2 may not recognize the disk partitions. During the installation, do not create any new partitions, but do set the proper partition types for your Linux partitions using the Linux `fdisk`.

B.1.7 Disk Partitions and Mount Points

One area that many people new to Linux find confusing is the matter of how partitions are used and accessed by the Linux operating system. In DOS/Windows, it is relatively simple: If you have more than one partition, each partition gets a "drive letter." You then use the drive letter to refer to files and directories on a given partition.

This is entirely different from how Red Hat Linux deals with partitions and, for that matter, with disk storage in general. The main difference is that each partition is used to form part of the storage necessary to support a single set of files and directories. This is done by associating a partition with a directory through a process known as **mounting**. Mounting a partition makes its storage available starting at the specified directory (known as a **mount point**).

For example, if partition `/dev/hda5` were mounted on `/usr`, that would mean that all files and directories under `/usr` would physically reside on `/dev/hda5`. So the file `/usr/doc/FAQ/txt/Linux-FAQ` would be stored on `/dev/hda5`, while the file `/etc/X11/gdm/Sessions/Gnome` would not.

Continuing our example, it is also possible that one or more directories below `/usr` would be mount points for other partitions. For instance, a partition (say, `/dev/hda7`) could be mounted on `/usr/local`, meaning that, for example, `/usr/local/man/whatis` would then reside on `/dev/hda7` rather than `/dev/hda5`.

B.1.8 How Many Partitions?

At this point in the process of preparing to install Red Hat Linux, you will need to give some consideration to the number and size of the partitions to be used by your new operating system. The question of "how many partitions" continues to spark debate within the Linux community and, without any end to the debate in sight, it’s safe to say that there are probably as many partition layouts as there are people debating the issue.

Keeping this in mind, we recommend that, unless you have a reason for doing otherwise, you should create the following partitions:

- A swap partition -- Swap partitions are used to support virtual memory. In other words, data is written to swap when there is not RAM to hold the data your system is processing. If your computer has 16 MB of RAM or less, you *must* create a swap partition. Even if you have more memory, a swap partition is still recommended. The minimum size of your swap partition should be equal to your computer's RAM, or 16 MB (whichever is larger).
-

Intel

The following partition is specific to Red Hat Linux/Intel installations.

A `/boot` partition -- The partition mounted on `/boot` contains the operating system kernel (which allows your system to boot Red Hat Linux), along with a few other files used during the bootstrap process. Due to the limitations of most PC BIOSes, creating a small partition to hold these files is a good idea. This partition should be no larger than 16MB.

Please Note

Make sure you read Section B.1.9, *One Last Wrinkle: Using LILO* -- the information there applies to the `/boot` partition!

-

Alpha

The following partition is specific to Red Hat Linux/Alpha installations.

A MILO partition -- Alpha owners that will be using MILO to boot their systems should create a 2MB DOS partition where MILO can be copied after the installation is complete. We recommend using `/dos` as the mount point.

- A root partition (/) -- The root partition is where `/` (the root directory) resides. In this partitioning layout, all files (except those stored in `/boot`) reside on the root partition. Because of this, it's in your best interest to maximize the size of your root partition. A 600MB root partition will permit the equivalent of a workstation-class installation (with *very* little free space), while a 1.5GB root partition will let you install every package.

B.1.9 One Last Wrinkle: Using LILO

LILO (the LInux LOader) is the most commonly used method to boot Red Hat Linux on Intel-based systems. An operating system loader, LILO operates "outside" of any operating system, using only the Basic I/O System (or BIOS) built into the computer hardware itself. This section describes LILO's interactions with PC BIOSes, and is specific to Intel-compatible computers.

BIOS-Related Limitations Impacting LILO

LILO is subject to some limitations imposed by the BIOS in most Intel-based computers. Specifically, most BIOSes can't access more than two hard drives and they can't access any data stored beyond cylinder 1023 of any drive. Note that some recent BIOSes do not have these limitations, but this is by no means universal.

All the data LILO needs to access at boot time (including the Linux kernel) are located in the `/boot` directory. If you follow the partition layout recommended above, or you are performing a workstation- or server-class install, the `/boot` directory will be in a small, separate partition. Otherwise, it will reside in the root partition. In either

case, the partition in which `/boot` resides must conform to the following guidelines if you are going to use LILO to boot your Red Hat Linux system:

On First Two IDE Drives

If you have 2 IDE (or EIDE) drives, `/boot` must be located on one of them. Note that this two-drive limit also includes any IDE CD-ROM drives on your primary IDE controller. So, if you have one IDE hard drive, and one IDE CD-ROM on your primary controller, `/boot` must be located on the first hard drive *only*, even if you have other hard drives on your secondary IDE controller.

On First IDE or First SCSI Drive

If you have one IDE (or EIDE) drive and one or more SCSI drives, `/boot` must be located either on the IDE drive or the SCSI drive at ID 0. No other SCSI IDs will work.

On First Two SCSI Drives

If you have only SCSI hard drives, `/boot` must be located on a drive at ID 0 or ID 1. No other SCSI IDs will work.

Partition *Completely* Below Cylinder 1023

No matter which of the above configurations apply, the partition that holds `/boot` must be located entirely below cylinder 1023. If the partition holding `/boot` straddles cylinder 1023, you may face a situation where LILO will work initially (because all the necessary information is below cylinder 1023), but will fail if a new kernel is to be loaded, and that kernel resides above cylinder 1023.

As mentioned earlier, it is possible that some of the newer BIOSes may permit LILO to work with configurations that don't meet our guidelines. Likewise, some of LILO's more esoteric features may be used to get a Linux system started, even if the configuration doesn't meet our guidelines. However, due to the number of variables involved, Red Hat cannot support such extraordinary efforts.

Please Note

Disk Druid as well as the workstation- and server-class installs take these BIOS-related limitations into account.

C Driver Disks

C.1 Why Do I Need a Driver Disk?

While the Red Hat Linux installation program is loading, you may see a screen that asks you for a driver disk. The driver disk screen is most often seen in three scenarios:

- if you are running the installation program in `expert mode`
- if you run the installation program by entering `linux dd` at the `boot:` prompt
- if you run the installation program on a computer which does not have any PCI devices

C.1.1 So What Is a Driver Disk Anyway?

A driver disk adds support for hardware that is not otherwise supported by the installation program. The driver disk could be produced by Red Hat, it could be a disk you make yourself, or it could be a disk that a hardware vendor includes with a piece of hardware.

There is really no need to use a driver disk unless you need a particular device in order to install Red Hat Linux. You will most likely use a driver disk for SCSI adapters and NICs, as those are really the only devices which are used during the installation that might require driver disk support. If an unsupported device is not needed to install Red Hat Linux on your system, continue with a regular installation and then add support for the new piece of hardware once the installation is complete.

C.1.2 How Do I Obtain a Driver Disk?

Your best option for finding driver disk information is on Red Hat's website at http://www.redhat.com/support/errata/ under the section called **Bug Fixes**.

If you find a driver disk that is appropriate for your device support needs, create a boot disk using that *`filename`*`.img` file. For instructions on how to make a boot disk, see the *Official Red Hat Linux Installation Guide* section called *Making Installation Diskettes*.

Once you have created your driver disk, boot your system using the diskette as a boot disk and enter either `linux expert` or `linux dd` at the `boot:` prompt.

D How to Create a Dual-Boot System

This document explains your options for installing Red Hat Linux on a computer that currently runs another operating system, and how to create a dual-boot environment.

D.1 If Your Computer Already Has An Operating System ...

If the computer you want to install Red Hat Linux on is currently running Windows (or some other operating system), you have an important decision to make. Your choices are:

- Do you want to install Red Hat Linux but feel timid about disk partitioning? You can install Red Hat Linux on your system without creating any Linux partitions by performing a **partitionless** installation. The installation program can install Red Hat Linux on an existing, formatted Windows partition and you'll only need to create a boot disk during the installation to access Red Hat Linux on your system.

  This method is perfect for those who do not want to install Red Hat Linux as the primary OS or as a dual-boot OS on your system. It is a great way of trying out Red Hat Linux without creating Linux partitions on your system.

  If this is your preferred choice, refer to *Official Red Hat Linux Installation Guide, Appendix B. Installing Without Partitioning*, for those instructions.

- Do you want to install Red Hat Linux and have the option of booting either Red Hat Linux or your other operating system? Performing a workstation- or custom-class installation will allow Red Hat Linux to install on your system without affecting the other operating system. A workstation-class installation will do this by default. In a custom-class installation, you must install LILO (the LInux LOader) on the MBR (Master Boot Record) -- unless Windows NT is your other OS or if you are already using another boot loader on your system. The *Official Red Hat Linux Installation Guide* has instructions on installing and configuring LILO. After the installation, whenever you start the computer, you will indicate whether you want to start Red Hat Linux or the other operating system.

WARNING

Do not perform a workstation-class installation if you're sharing a disk with Windows NT; if you do, you will be unable to boot Windows NT. LILO will overwrite NT's boot loader and you will be unable to boot NT. You must perform a custom-class installation and configure LILO so that it is not installed on the MBR.

To create a dual-boot environment on a system that currently has NT, you must install LILO on the first sector of the root partition, not the MBR. Please be sure to create a boot disk. In a case such as this, you will either need to use the boot disk, or configure the NT system loader to boot LILO from the first sector of the root partition. Be sure to check out http://www.linux-doc.org/HOWTO/mini/Linux+NT-Loader.html for more information on setting up LILO and NT.

If this is your preferred choice, read Section D.2, *Setting Up a Dual-Boot Environment*

- Do you want Red Hat Linux to be the only operating system on your computer? The Red Hat Linux installation program will remove the other operating system and also any data you have created using that operating system.

Please Note

In order to install Red Hat Linux and keep another OS on your system, there must be sufficient space for Red Hat Linux to be installed on. Otherwise, Red Hat Linux will replace the current OS and files on your system. If you have not partitioned your hard drive to make room for Red Hat Linux or made sure that there is sufficient unpartitioned space available for your installation, Red Hat Linux will install over the existing information by default. It will also happen if you select a server-class installation (note that a server-class installation does not install the X Window System so no GUI environment will be present [1]). So a dual-boot environment is incompatible with this choice.

If this is your preferred choice, first back-up any information on your computer that you want to save or perform a full backup if you think you may want to restore your system to its original configuration), then proceed with installation as explained in the *Official Red Hat Linux Installation Guide*.

D.2 Setting Up a Dual-Boot Environment

Sharing a computer between two operating systems requires dual booting. You can use either operating system on the computer but not both at once. Each operating system boots from and uses its own hard drives or disk partitions (a partition is a physical division on a hard drive).

For clarity, we will assume that the other operating system is Windows. But the general procedures are similar for other operating systems.

[1] A server-class installation is most appropriate for you if you'd like your system to function as a Linux-based server, and you don't want to heavily customize your system configuration.

Please Note

If Red Hat Linux will coexist on your system with OS/2, you must create your disk partitions with the OS/2 partitioning software -- otherwise, OS/2 may not recognize the disk partitions. During the installation, do not create any new partitions, but do set the proper partition types for your Linux partition using fdisk.

Before starting the installation program, you must first make room for Red Hat Linux. Your choices are:

- Add a new hard drive
- Use an existing hard drive or partition
- Create a new partition

D.2.1 Add a New Hard Drive

The simplest way to make room for Red Hat Linux is to add a new hard drive to the computer and then install Red Hat Linux on that drive. For example, if you add a second IDE hard drive to the computer, the Red Hat Linux installation program will recognize it as `hdb` and the existing drive (the one used by Windows) as `hda`. (With SCSI hard drives, the newly installed hard drive would be recognized as `sdb` and the other hard drive as `sda`.)

If you choose to install a new hard drive for Linux, you don't need to read any further. After starting the Red Hat Linux installation program, just make sure you tell it to install Linux on the newly installed hard drive (`hdb`, `sdb`) rather than the one Windows uses.

D.2.2 Use an Existing Hard Drive or Partition

The next simplest way to make room for Linux is to use a hard drive or disk partition that is currently being used by Windows. For example, suppose that Windows Explorer shows two hard drives, `C:` and `D:`. This could indicate either that the computer has

two hard drives, or a single hard drive with two partitions. In either case (assuming it is large enough), you can install Red Hat Linux on the hard drive or disk partition that Windows recognizes as `D:`.

This choice is available to you only if the computer has two or more hard drives or disk partitions.

Please Note

Windows uses letters to refer to removable drives (for example, a ZIP drive) and network storage (virtual drives) as well as for local hard drive space: you cannot install Linux on a removable or network drive.

If a local Windows partition is available that you want to install Linux in, you don't need to read any further. Just do the following:

1. Copy all data you want to save from the selected hard drive or partition (`D:` in this example) to another location.
2. Start the Red Hat Linux installation program and tell it to install Linux in the designated drive or partition -- in this example, in the hard drive or partition that Windows designates as `D:`. Note that Linux distinguishes between hard drives and disk partitions. Thus:
   - If `C:` and `D:` on this computer refer to two separate hard drives, the installation program will recognize them as `hda` and `hdb` (IDE) or `sda` and `sdb` (SCSI). Tell the installation program to use `hdb` or `sdb`.
   - If `C:` and `D:` refer to partitions on a single drive, the installation program will recognize them as `hda1` and `hda2` (or `sda1` and `sda2`). During the partitioning phase of Linux installation, you'll delete the second partition (`hda2` or `sda2`), then partition the unallocated free space for Linux. (You don't have to delete the second partition prior to beginning Linux partitioning. But if you don't, Windows will complain whenever you boot that it cannot read `Drive D`; and should someone accidentally format `D`, your Linux system would be destroyed.)

D.2.3 Create a New Partition

The third way to make room for Linux is to create a new partition for Red Hat Linux on the hard drive being used by the other operating system. If Windows Explorer shows only one hard drive (`C:`), and you don't want to add a new hard drive, you must partition the drive. After partitioning, Windows Explorer will reveal a smaller `C:` drive; and, when you run the Red Hat Linux installation program, it will partition the remainder of the drive for Linux.

You can use a destructive partitioning program, such as fdisk, to divide the hard drive, but doing so will require you to re-install Windows. (This is probably not your best option.)

A number of non-destructive third-party partitioning programs are available for the Windows operating system. If you choose to use one of these, consult their documentation.

For instructions on how to partition with FIPS, a program that is on the Red Hat Linux CD-ROM, turn to Section D.3, *Partitioning with* FIPS.

D.3 Partitioning with FIPS

As a convenience to our customers, we provide the FIPS utility. This is a freely available program that can resize FAT (File Allocation Table) partitions. It's included on the Red Hat Linux CD-ROM in the `dosutils` directory.

Please Note

Many people have successfully used FIPS to repartition their hard drives. However, because of the nature of the operations carried out by FIPS, and the wide variety of hardware and software configurations under which it must run, Red Hat cannot guarantee that FIPS will work properly on your system. Therefore, no installation support whatsoever is available for FIPS; use it at your own risk.

That said, if you decide to repartition your hard drive with FIPS, it is vital that you do two things:

- `Perform a Backup` -- Make two copies of all the important data on your computer. These copies should be to removable media (such as tape or diskettes), and you should make sure they are readable before proceeding.
- `Read the Documentation` -- Completely read the FIPS documentation, located in the FIPS directory on Red Hat Linux CD-ROM.

Should use decide to use FIPS, be aware that after FIPS runs you will be left with two partitions: the one you resized, and the one FIPS created out of the newly freed space. If your goal is to use that space to install Red Hat Linux, you should delete the newly created partition, either by using fdisk under your current operating system, or while setting up partitions during a custom-class installation.

The following instructions are a simplified version of the FIPS documentation file, `fips.doc`, located in the FIPS directory (`/dosutils/fips20/*`). These instructions should apply in most instances. If you encounter any problems, see the documentation file.

1. From Windows:
   - Do a full backup.
   - Run scandisk to verify that the hard drive contains no bad clusters.
   - Decide how to distribute the available space on the hard drive between the operating systems. Use Windows Explorer to see the free space on the drive. Make a note of the space (in megabytes) that each operating system will have.
   - If you don't have one, create a DOS boot disk.

     To create a DOS boot disk, first boot your machine to DOS.

     Next, insert a blank, formatted diskette into the floppy drive.

     Type the following at the command prompt and press [Enter]:

     ```
     FORMAT A: /S
     ```

If you're using Windows 95, first insert a blank formatted diskette into the floppy drive. Next, go to **Start**/**Run**, and type:

```
FORMAT A: /S
```

The diskette will be formatted, and `COMMAND.COM`, along with the associated hidden files (`IO.SYS`, `MSDOS.SYS`, and `BDLSAPCE.BIN`), will be copied to the diskette.

- Copy the following files on the Red Hat Linux CD-ROM to the DOS boot disk.

```
/mnt/cdrom/dosutils/fips20/fips.exe
/mnt/cdrom/dosutils/fips20/restorrb.exe
/mnt/cdrom/dosutils/fips20/errors.txt
/mnt/cdrom/dosutils/fips20/fips.doc
/mnt/cdrom/dosutils/fips20/fips.faq
```

- Defragment the hard drive.

2. Insert the DOS boot disk into the floppy drive and reboot the system.
3. Start FIPS (type `fips` at the prompt).

When FIPS begins, you'll find a welcome screen similar to the following:

Figure D–1 FIPS Welcome Screen

```
FIPS version 2.0, Copyright (C) 1993/4 Arno Schaefer
FAT32 Support, Copyright (C) 1997 Gordon Chaffee

DO NOT use FIPS in a multitasking environment like Windows, OS/2, Desqview,
Novell Task manager or the Linux DOS emulator; boot from a DOS boot disk fir

If you use OS/2 or a disk compressor, read the relevant sections in FIPS.DOC

FIPS comes with ABSOLUTELY NO WARRANTY, see file COPYING for details.

This is free software, and you are welcome to redistribute it
under certain conditions; again, see file COPYING for details.
```

```
Press any key.
```

When you press a key, a root partition screen similar to the following appears. (Note that, if the computer has more than one hard drive, you'll be asked to select which one you want to partition.)

Figure D–2 FIPS Root Partition Screen

```
Partition table:

     |        |     Start      |      |      End       | Start  |Number of|
Part.|bootable|Head Cyl. Sector|System|Head Cyl. Sector| Sector |Sectors  |  MB
-----+--------+----------------+------+----------------+--------+---------+----
1    |    yes |   0  148      1|   83h|  15  295     63|  149184|   149184|  72
2    |     no |   1    0      1|   06h|  15  139     63|      63|   141057|  68
3    |     no |   0  140      1|   06h|  15  147     63|  141120|     8064|   3
4    |     no |   0    0      0|   00h|   0    0      0|       0|        0|   0

Checking root sector ... OK

Press any key.
```

When you press a key, details about the hard drive, such as the following, will appear.

Figure D–3 FIPS Boot Sector Screen

```
Boot sector:
Bytes per sector: 512
Sectors per cluster: 8
Reserved sectors: 1
Number of FATs: 2
Number of rootdirectory entries: 512
Number of sectors (short): 0
Media descriptor byte: f8h
Sectors per FAT: 145
Sectors per track: 63
Drive heads: 16
Hidden sectors: 63
Number of sectors (long): 141057
Physical drive number: 80h
Signature: 29h
```

```
Checking boot sector ... OK
Checking FAT ... OK
Searching for free space ... OK

Do you want to make a backup copy of your root and boot sector before
proceeding? (y/n)
```

You should select **y**, for `yes`, to make a backup copy of your root and boot sector before proceeding with FIPS.

Next, you'll be presented with the following message:

```
Do you have a bootable floppy disk in drive A: as described in the
documentation? (y/n)
```

Verify that a DOS boot disk is in the floppy drive, and type **y**, for `yes`. A screen similar to the following will appear, allowing you to resize the partition.

Figure D–4 Partition Resizing Screen

```
Writing file a:\rootboot:000

Enter start cylinder for new partition (33-526)

Use the cursor keys to choose the cylinder, <enter> to continue

Old partition           Cylinder              New partition
 258.9 MB                  33                    3835.8 MB
```

The initial values allocate *all* free space on the disk to the new partition. This is not what you want, because this setting would leave no free space on your Windows partition. Press the [right arrow] to increase the size of the Windows partition and decrease the size of the new (Linux) partition; press the [left arrow] to decrease the size of the Windows partition and increase the size of the Linux partition. When the sizes are what you want, press [Enter]. A verification screen similar to the following appears:

Figure D–5 FIPS Verification Screen

```
First Cluster: 17442
```

```
Last Cluster: 65511

Testing if empty ... OK

New partition table:

     |        |     Start      |      |      End       | Start  |Number of|
Part.|bootable|Head Cyl. Sector|System|Head Cyl. Sector| Sector |Sectors  |  MB
-----+--------+----------------+------+----------------+--------+---------+----
1    |    yes |   0  148      1|   83h|  15  295     63|  149184|   149184|  1090
2    |     no |   0  139      1|   06h| 254  521     63| 2233035|  6152995|  3084
3    |     no |   0  140      1|   06h|  15  147     63|  141120|     8064|   3
4    |     no |   0    0      0|   00h|   0    0      0|       0|        0|   0

Checking root sector ... OK

Do you want to continue or reedit the partition table (c/r)?
```

If you answer **r** (to re-edit the partition tables), *Figure 4* reappears, allowing you to change the partition sizes. If you answer **c**, a confirmation screen (*Figure 6*) appears:

Figure D–6 FIPS Confirmation Screen

```
New boot sector:

Boot sector:
Bytes per sector: 512
Sectors per cluster: 8
Reserved sectors: 1
Number of FATs: 2
Number of rootdirectory entries: 512
Number of sectors (short): 0
Media descriptor byte: f8h
Sectors per FAT: 145
Sectors per track: 63
Drive heads: 16
Hidden sectors: 63
Number of sectors (long): 141057
Physical drive number: 80h
Signature: 29h

Checking boot sector ... OK
```

```
Ready to write new partition scheme to disk
Do you want to proceed (y/n)?
```

Answering **y** completes the resizing operation. A harmless error message may occur, stating, in effect that FIPS cannot reboot the system.

After a successful operation, the disk will have two partitions. The first partition (`hda1` or `sda1`) will be used by Windows. We recommend that you start Windows (remember to remove the boot disk from drive `A:`) and run scandisk on drive `C:`.

If you encounter any problems, (for example, Windows will not boot), you can reverse the FIPS resizing operation with the restorrb.exe command, which you copied to your DOS boot disk. In case of any errors, read the FIPS documentation files (`fips.doc` and `fips.faq`), which indicate a number of factors that could cause the resizing operation to fail. If all else fails, you can restore Windows with the backup you made.

The second partition (`hda2` or `sda2`) contains the space that the Red Hat Linux installation program will use. When the Disk Druid screen appears during installation, delete this partition (the installation manual explains how), then proceed with Linux partitioning.

E RAID (Redundant Array of Independent Disks)

E.1 What is RAID?

The basic idea behind RAID is to combine multiple small, inexpensive disk drives into an array which yields performance exceeding that of one large and expensive drive. This array of drives will appear to the computer as a single logical storage unit or drive.

RAID is a method in which information is spread across several disks, using techniques such as **disk striping** (RAID Level 0) and **disk mirroring** (RAID level 1) to achieve redundancy, lower latency and/or higher bandwidth for reading and/or writing to disks, and maximize recoverability from hard-disk crashes.

The underlying concept in RAID is that data may be distributed across each drive in the array in a consistent manner. To do this, the data much first be broken into consistently-sized "chunks" (often 32K or 64K in size, although different sizes can be used). Each chunk is then written to each drive in turn. When the data is to be read, the process is reversed, giving the illusion that multiple drives are actually one large drive.

E.1.1 Who Should Use RAID?

Those of you who need to keep large quantities of data on hand (such as an average administrator) would benefit by using RAID technology. Primary reasons to use RAID include:

- enhanced speed
- increased storage capacity
- greater efficiency in recovering from a disk failure

E.1.2 RAID: Hardware vs. Software

There are two possible approaches to RAID: Hardware RAID and Software RAID.

Hardware RAID

The hardware-based system manages the RAID subsystem independently from the host and presents to the host only a single disk per RAID array.

An example of a hardware RAID device would be one that connects to a SCSI controller and presents the RAID arrays as a single SCSI drive. An external RAID system moves all RAID handling "intelligence" into a controller located in the external disk subsystem. The whole subsystem is connected to the host via a normal SCSI controller and appears to the host as a single disk.

RAID controllers also come in the form of cards that *act* like a SCSI controller to the operating system, but handle all of the actual drive communications themselves. In these cases, you plug the drives into the RAID controller just like you would a SCSI controller, but then you add them to the RAID controller's configuration, and the operating system never knows the difference.

Software RAID

Software RAID implements the various RAID levels in the kernel disk (block device) code. It also offers the cheapest possible solution: Expensive disk controller cards or hot-swap chassis [1]are not required, and software RAID works with cheaper IDE disks as well as SCSI disks. With today's fast CPUs, software RAID performance can excel against hardware RAID.

The MD driver in the Linux kernel is an example of a RAID solution that is completely hardware independent. The performance of a software-based array is dependent on the server CPU performance and load.

E.1.3 Some Features of RAID

For those interested in learning more about what software RAID has to offer, here is a brief list of few of those features:

- Threaded rebuild process
- Fully kernel-based configuration

[1] A hot-swap chassis allow you to remove a hard drive without having to power-down your system.

- Portability of arrays between Linux machines without reconstruction
- Backgrounded array reconstruction using idle system resources
- Hot-swappable drive support
- Automatic CPU detection to take advantage of certain CPU optimizations

Levels and linear support

RAID also offers levels 0, 1, 4, 5, and linear support. These RAID types act as follows:

- *Level 0* -- RAID level 0, often called "striping," is a performance- oriented striped data mapping technique. That means the data being written to the array is broken down into strips and written across the member disks of the array. This allows high I/O performance at low inherent cost but provides no redundancy. Storage capacity of the array is equal to the total capacity of the member disks.
- *Level 1* -- RAID level 1, or "mirroring," has been used longer than any other form of RAID. Level 1 provides redundancy by writing identical data to each member disk of the array, leaving a "mirrored" copy on each disk. Mirroring remains popular due to its simplicity and high level of data availability. Level 1 operates with two or more disks that may use parallel access for high data-transfer rates when reading, but more commonly operate independently to provide high I/O transaction rates. Level 1 provides very good data reliability and improves performance for read-intensive applications but at a relatively high cost[2]. Array capacity is equal to the capacity of one member disk.
- *Level 4* -- Level 4 uses parity[3]concentrated on a single disk drive to protect data. It's better suited to transaction I/O rather than large file transfers. Because the dedicated parity disk represents an inherent bottleneck, level 4 is seldom used

[2] RAID level 1 is at a high cost because you write the same information to all of the disks in the array, which wastes drive space. For example, you have RAID level 1 set up so that your "/" (root) partition spans across two 4G drives. You have 8G total but are only able to access 4G of that 8G. The other 4G acts like a mirror of the first 4G.

[3] Parity information is calculated based on the contents of the rest of the member disks in the array. This information can then be used to reconstruct data when a disk in the array fails. The reconstructed data can then be used to satisfy I/O requests to the failed disk, and to repopulate the failed disk after it has been repaired or replaced.

without accompanying technologies such as write-back caching. Although RAID level 4 is an option in some RAID partitioning schemes, it is not an option allowed in Red Hat Linux RAID installations[4]. Array capacity is equal to the capacity of member disks, minus capacity of one member disk.

- *Level 5* -- The most common type of RAID. By distributing parity across some or all of an array's member disk drives, RAID level 5 eliminates the write bottleneck inherent in level 4. The only bottleneck is the parity calculation process. With modern CPUs and software RAID, that isn't a very big bottleneck. As with level 4, the result is asymmetrical performance, with reads substantially outperforming writes. Level 5 is often used with write-back caching to reduce the asymmetry. Array capacity is equal to the capacity of member disks, minus capacity of one member disk.
- *Linear RAID* -- Linear RAID is a simple grouping of drives to create a larger virtual drive. In linear RAID, the chunks are allocated sequentially from one member drive, going to the next drive only when the first is completely filled. This grouping provides no performance benefit, as it is unlikely that any I/O operations will be split between member drives. Linear RAID also offers no redundancy, and in fact decreases reliability -- if any one member drive fails, the entire array cannot be used. The capacity is total of all member disks.

E.1.4 Creating RAID Partitions

RAID is available in both the GUI and kickstart installation modes. You can use fdisk or Disk Druid to create your RAID configuration, but these instructions will focus mainly on using Disk Druid to complete this task.

Before you can create a RAID device, you must first create RAID partitions, using the following step-by-step instructions.

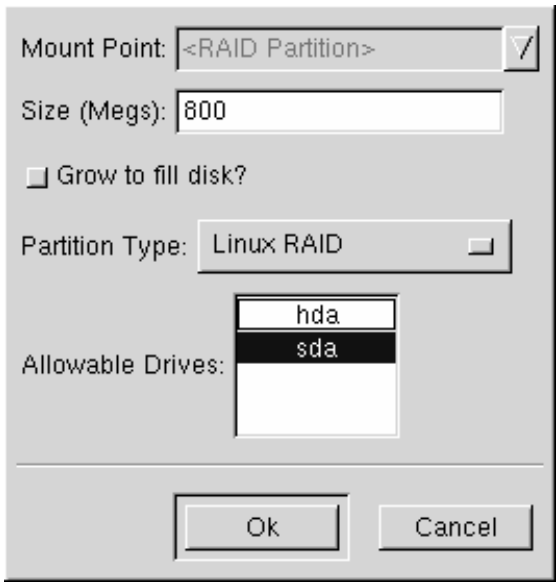
[4] RAID level 4 takes up the same amount of space as RAID level 5, but level 5 has many advantages over level 4 which is why it is not supported.

Tip: If You Use fdisk

If you are using fdisk to create a RAID partition, bear in mind that instead of creating a partition as type `83`, which is Linux native, you must create the partition as type `fd` (Linux RAID).

- Create a partition. In Disk Druid, choose **Add** to create a new partition (see Figure E–1, *Creating a New RAID Partition*).

Figure E–1 Creating a New RAID Partition

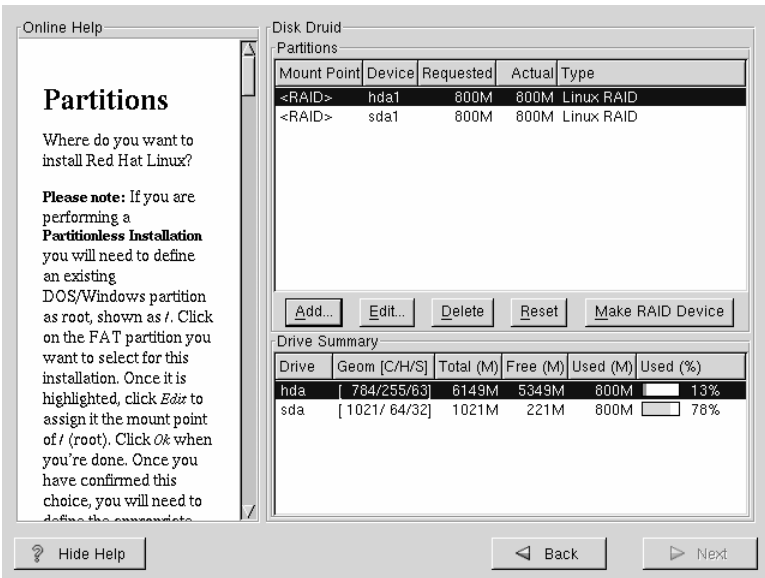


- You will not be able to enter a mount point (you will be able to do that once you've created your RAID device).
- Enter the size that you want the partition to be.

- Select **Grow to fill disk** if you want the partition to grow to fill all available space on the hard disk. In this case, the partition's size will expand and contract as other partitions are modified. If you make more than one partition grow-able, the partitions will compete for the available free space on the disk.
- Enter the partition type as RAID.
- Finally, for **Allowable Drives**, select the drive that RAID will be created on. If you have multiple drives, all drives will be selected here and you must deselect those drives which will *not* have RAID array on it.

Continue these steps to create as many partitions as needed for your RAID setup.

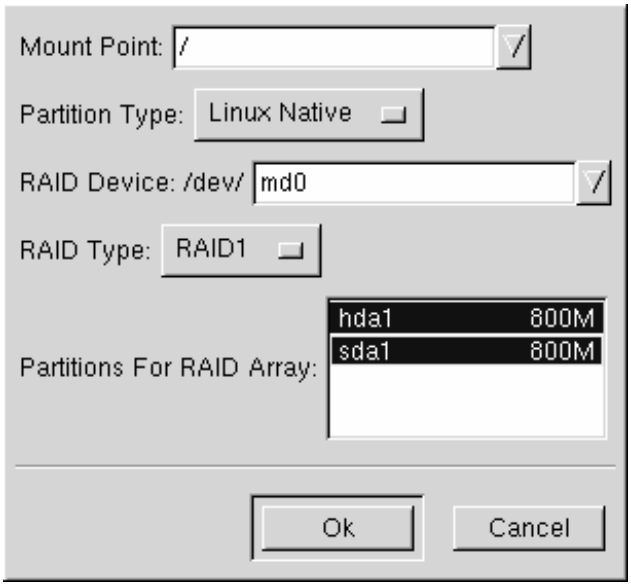
Figure E–2 RAID Partitions



Once you have all of your partitions created as RAID partitions, select the **Make RAID Device** button on the Disk Druid main partitioning screen (see Figure E–2, *RAID Partitions*).

Next, Figure E–3, *Making a RAID Device* will appear which will allow you to make a RAID device.

Figure E–3 Making a RAID Device



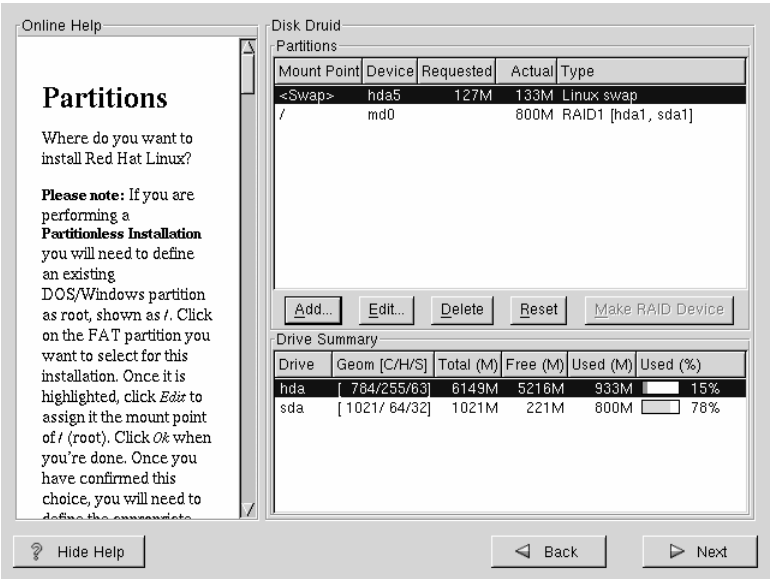
- First, enter a mount point.
- Next, make sure the partition type is set as **Linux Native** (which will be the default).
- Choose your RAID device. You should choose `md0` for your first device, `md1` for your second device, and so on, unless you have a specific reason to make it something else. Raid devices range from md0 to md7, and each may only be used once.
- Choose your RAID type. You can choose from **RAID 0**, **RAID 1**, and **RAID 5**.

Please Note

If you are making a RAID partition of `/boot`, you must choose RAID level 1. If you are not creating a RAID partition of `/boot`, and are making a RAID partition of `/`, it must be RAID level 1.

- Finally, select which partitions will go into this RAID array (as in Figure E–4, *Creating a RAID Array*) and then click **Next**.

Figure E–4 Creating a RAID Array



- From here, you can continue with your installation process. Refer back to the *Official Red Hat Linux Installation Guide* for further instructions.

F Kickstart Installations

F.1 What are Kickstart Installations

Due to the need for automated installation, Red Hat has created the kickstart installation method. With this method, a system administrator can create a single file containing the answers to all the questions that would normally be asked during a typical Red Hat Linux installation.

Kickstart files can be kept on single server system, and read by individual computers during the installation. The kickstart installation method is powerful enough that often a single kickstart file can be used to install Red Hat Linux on multiple machines, making it ideal for network and system administrators.

Kickstart lets you automate most of a Red Hat Linux installation, including:

- Language selection
- Network configuration
- Keyboard selection
- Boot loader installation (LILO)
- Disk partitioning
- Mouse Selection
- X Window System configuration

F.2 How Do You Perform a Kickstart Installation?

Kickstart installations can only be performed using a local CD-ROM, a local hard drive, or NFS installation methods. FTP and HTTP installations cannot be automated using kickstart mode.

To use kickstart mode, you must first create a kickstart file (`ks.cfg`), and make it available to the Red Hat Linux installation program.

F.2.1 Where to Put A Kickstart File

A kickstart file must be placed in one of two locations:

- on a boot disk
- on a network

Normally a kickstart file is copied to the boot disk, or made available on the network. The network-based approach is most commonly used, as most kickstart installations tend to be performed on networked computers.

Let's take a more in-depth look at where the kickstart file may be placed.

To perform a diskette-based kickstart installation, the kickstart file must be named `ks.cfg`, and reside in the boot disk's top-level directory. Note that the Red Hat Linux boot disks are in MS-DOS format, making it easy to copy the kickstart file under Linux using the `mcopy` command:

```
mcopy ks.cfg a:
```

(or, if you insist, you can also use Windows). You can also mount the MS-DOS boot disk and `cp` the file over. Although there's no technological requirement for it, most diskette-based kickstart installations install Red Hat Linux from a local CD-ROM.

Network installations using kickstart are quite common, because system administrators can easily automate the installation on many networked computers quickly and painlessly. In general, the approach most commonly used is for the administrator to have both a BOOTP/DHCP server and an NFS server on the local network. The BOOTP/DHCP server is used to give the client system its networking information, while the actual files used during the installation are served by the NFS server. Often, these two servers run on the same physical machine, but there is no requirement for this.

To do a network-based kickstart installation, you must have a BOOTP/DHCP server on your network, and it must include configuration information for the machine on which you are attempting to install Red Hat Linux. The BOOTP/DHCP server will be used to give the client its networking information as well as the location of the kickstart file.

If a kickstart file is specified by the BOOTP/DHCP server, the client system will attempt an NFS mount of the file's path, and will copy the specified file to the client, using it as the kickstart file. The exact settings required vary depending on the BOOTP/DHCP server you use.

Here's an example of a line from the `dhcpd.conf` file for the DHCP server shipped with Red Hat Linux:

```
filename "/usr/new-machine/kickstart/";
next-server blarg.redhat.com;
```

Note that you should use `filename` for the kickstart file's name (or the directory in which the kickstart file resides) and `next-server` to set the NFS server name.

If the filename returned by the BOOTP/DHCP server ends with a slash ("/"), then it is interpreted as a path only. In this case, the client system mounts that path using NFS, and searches for a specially-named file. The filename the client searches for is:

```
<ip-addr>-kickstart
```

The *<ip-addr>* section of the filename should be replaced with the client's IP address in dotted decimal notation. For example, the filename for a computer with an IP address of 10.10.0.1 would be `10.10.0.1-kickstart`.

Note that if you don't specify a server name, then the client system will attempt to use the server that answered the BOOTP/DHCP request as its NFS server. If you don't specify a path or filename, the client system will try to mount `/kickstart` from the BOOTP/DHCP server, and will try to find the kickstart file using the same *<ip-addr>*`-kickstart` filename as described above.

F.3 Starting a Kickstart Installation

To begin a kickstart installation, you must boot the system from a Red Hat Linux boot disk, and enter a special boot command at the boot prompt. If the kickstart file resides on the boot disk, the proper boot command would be:

```
boot: linux ks=floppy
```

If, on the other hand, the kickstart file resides on a server, the appropriate boot command would be:

```
boot: linux ks
```

Anaconda looks for a kickstart file if the `ks` command line argument is passed to the kernel. It can take a number of forms:

`ks=floppy`

The installation program looks for the file `ks.cfg` on a VFAT filesystem on the floppy in drive /dev/fd0.

`ks=hd:`*`<device>/<file>`*

The installation program will mount the filesystem on <*device*> (which must be VFAT or ext2), and look for the kickstart configuration file as <*file*> in that filesystem (for example, `ks=hd:sda3/mydir/ks.cfg`).

`ks=file:/`*`<file>`*

The installation program will try to read the file <*file*> from the filesystem; no mounts will be done. This is normally used if the kickstart file is already on the `initrd` image.

`ks=nfs:`*`<server:>/<path>`*

The installation program will look for the kickstart file on the NFS server <*server*>, as file <*path*>. The installation program will use DHCP to configure the Ethernet card.

`ks`

If `ks` is used alone, the installation program will configure the Ethernet card in the system using DHCP. The system will use the "bootServer" from the DHCP response as an NFS server to read the kickstart file from (by default, this is the same as the DHCP server). The name of the kickstart file is one of the following:

- If DHCP is specified and the "bootfile" begins with a `/`, that file is looked for on the NFS server.

- If DHCP is specified and the "bootfile" begins with something other then a `/`, that file is looked for in the `/kickstart` directory on the NFS server.
- If DCHP did not specify a "bootfile," then the installation program tries to read the file `/kickstart/1.2.3.4-kickstart`, where *1.2.3.4* is the numeric IP address of the machine being installed.

F.4 The Kickstart File

Now that you have some background information on kickstart installations, let's take a look at the kickstart file itself. The kickstart file is a simple text file, containing a list of items, each identified by a keyword. You can create it by editing a copy of the `sample.ks` file found in the `/doc` directory of the Red Hat Linux CD-ROM, or you can create it from scratch. You should be able to edit it with any text editor or word processor that can save files as ASCII text.

First, some ground rules to keep in mind while creating your kickstart file:

- Items must be specified *in order*. It is not a good idea to try to change the order of the required items.
- Items that aren't required can be omitted.
- Omitting any required item will result in the installation program prompting the user for an answer to the related item, just as during a typical installation. Once the answer is given the installation will continue unattended (unless it comes across another missing item).
- Lines starting with a pound sign ("#") are treated as comments, and are ignored.
- For kickstart *upgrades*, the following items are required:
  - language
  - installation method
  - device specification (if device is needed to perform installation)
  - keyboard setup
  - the `upgrade` keyword

– LILO configuration

If any other items are specified for an upgrade, those items will be ignored (note that this includes package selection).

- Kickstart files are split into three sections: commands, package list, and scripts. The file must be of the form:
  - *<kickstart commands>*
  - %packages
  - *<package list>*
  - %post
  - *<post script>*

The order matters; it can't be random. The post section goes to the end of the file and ends the file, no marker is necessary to end the file other than the post section itself.

F.5 Kickstart Commands

The following commands can be placed in a kickstart file.

F.5.1 `auth` – Authentication Options

`auth` (required)

Sets up the authentication options for the system. It's similar to the `authconfig` command that can be run after the install. By default, passwords are normally encrypted and are not shadowed.

`--enablemd5`

Use md5 encryption for user passwords.

`--enablenis`

Turns on NIS support. By default, `--enablenis` uses whatever domain it finds on the network. A domain should almost always be set by hand (via `--nisdomain`).

`--nisdomain`

NIS domain name to use for NIS services.

`--nisserver`

Server to use for NIS services (broadcasts by default).

`--useshadow`

Use shadow passwords.

F.5.2 `clearpart` – Removing partitions based on partition type

`clearpart` (optional)

Removes partitions from the system, prior to creation of new partitions. By default, no partitions are removed.

`--linux`

Erases Linux (type 0x82, 0x83, and 0xfd [RAID]) partitions

`--all`

Erases all partitions from the system.

F.5.3 `device --opts`

`device` (optional)

On most PCI systems, the installation program will autoprobe for Ethernet and SCSI cards properly. On older systems, and some PCI systems, kickstart needs a hint to find the proper devices, however. The device command tells Anaconda to install extra modules. It is of the form:

```
device <type> <moduleName> --opts <options>
```

<*type*> should be one of "scsi" or "eth", and <*moduleName*> is the name of the kernel module which should be installed.

`--opts`

Options to pass to the kernel module. Note that multiple options may be passed if put in quotes. For example:

```
--opts "aic152x=0x340 io=11"
```

F.5.4 driver disk

`driverdisk` (optional)

During kickstart, driver disks can be used by copying the contents of a driver disk to the root directory of a partition on the system's hard drive and using the `driverdisk` command to tell the installation program where to look for it.

```
driverdisk <partition> [--type <fstype>]
```

<*partition*> is the partition containing the driver disk.

`--type`

Filesystem type (for example, VFAT or ext2).

F.5.5 `install`

`install` (optional)

Tells the system to install a fresh system rather than upgrade an existing system. This is the default mode.

F.5.6 Installation methods

You must use one of the these three commands to specify what type of kickstart is being done:

NFS

Install from the NFS server specified.

- --server <*server*>

  Server from which to install (hostname or IP).

- --dir <*dir*>

  Directory containing the Red Hat installation tree.

For example:

```
nfs --server <server> --dir <dir>
```

CD-ROM

Install from the first CD-ROM drive on the system.

For example:

```
cdrom
```

hard drive

Install from a Red Hat installation tree on a local drive, which must be either VFAT or ext2.

- --partition <*partition*>

  Partition to install from (such as, sdb2).

- --dir <*dir*>

  Directory containing the Red Hat installation tree.

For example:

```
harddrive --partition <partition> --dir <dir>
```

F.5.7 `keyboard`

`keyboard` (required)

Sets system keyboard type. Here's the list of available keyboards on i386 and Alpha machines:

```
azerty, be-latin1, be2-latin1, fr-latin0, fr-latin1, fr-pc, fr,
wangbe, ANSI-dvorak, dvorak-l, dvorak-r, dvorak, pc-dvorak-latin1,
tr_f-latin5, trf, bg, cf, cz-lat2-prog, cz-lat2, defkeymap,
defkeymap_V1.0, dk-latin1, dk. emacs, emacs2, es, fi-latin1, fi,
gr-pc, gr, hebrew, hu101, is-latin1, it-ibm, it, it2, jp106,
la-latin1, lt, lt.l4, nl, no-latin1, no, pc110, pl, pt-latin1,
pt-old, ro, ru-cp1251, ru-ms, ru-yawerty, ru, ru1, ru2, ru_win,
se-latin1, sk-prog-qwerty, sk-prog, sk-qwerty, tr_q-latin5, tralt,
trf, trq, ua, uk, us, croat, cz-us-qwertz, de-latin1-nodeadkeys,
de-latin1, de, fr_CH-latin1, fr_CH, hu, sg-latin1-lk450,
sg-latin1, sg, sk-prog-qwertz, sk-qwertz, slovene
```

Here's the list for SPARC machines:

```
sun-pl-altgraph, sun-pl, sundvorak, sunkeymap, sunt4-es,
sunt4-no-latin1, sunt5-cz-us, sunt5-de-latin1, sunt5-es,
sunt5-fi-latin1, sunt5-fr-latin1, sunt5-ru, sunt5-uk, sunt5-us-cz
```

F.5.8 `language`

`lang` (required)

Sets the default language for the installed system. The language you specify will be used during the installation as well as to configure any language-specific aspect of the installed system. For example, to set the language to English, the kickstart file should contain the following line:

```
lang en_US
```

Valid languages codes are:

```
cs_CZ, en_US, fr_FR, de_DE, hu_HU, is_IS, id_ID, it_IT,
ja_JP.ujis, no_NO, pl_PL, ro_RO, sk_SK, sl_SI, es_MX,
ru_RU.KOI8-R, uk_UA
```

F.5.9 `lilo`

`lilo` (required)

Specifies how the boot loader should be installed on the system. By default, LILO installs on the MBR of the first disk, and installs a dual-boot system if a DOS partition is found (the DOS/Windows system will boot if the user types **`dos`** at the `LILO:` prompt).

`--append` <*params*>

Specifies kernel parameters.

`--linear`

Use the `linear` LILO option; this is only for backwards compatibility (and linear is now used by default).

`--location`

Specifies where the LILO boot record is written. Valid values are **`mbr`** (default), **`partition`** (installs the boot loader on the first sector of the partition containing the kernel), or none, which prevents any bootloader from being installed.

F.5.10 `lilocheck`

`lilocheck` (optional)

If this is present, the installation program checks for LILO on the MBR of the first hard drive, and reboots the system if it is found -- No installation is done in this case. This can prevent the kickstart from reinstalling an already installed system.

F.5.11 `mouse`

`mouse` (required)

Configures the mouse for the system, both in GUI and text modes. Options are:

`--device` <*dev*>

Device the mouse is on (such as --device ttyS0)

`--emulthree`

If present, the X Window System uses simultaneous left+right mouse buttons to emulate the middle button (should be used on two button mice).

After options, the mouse type may be specified as one of the following:

```
alpsps/2, ascii, asciips/2, atibm, generic, generic3,
genericps/2, generic3ps/2, geniusnm, geniusnmps/2,
geniusnsps/2, thinking, thinkingps/2, logitech,
logitechcc, logibm, logimman, logimmanps/2, logimman+,
logimman+ps/2, microsoft, msnew, msintelli, msintellips/2,
msbm, mousesystems, mmseries, mmhittab, sun, none
```

If the mouse command is given without any arguments, or it is omitted, the installation program will attempt to autodetect the mouse (which works for most modern mice).

F.5.12 `network`

`network` (optional)

Configures network information for the system. If it is not given, and the kickstart installation does not require networking (in other words, it's not installed over NFS), networking is not configured for the system. If the installation does require networking, Anaconda assumes that the install should be done over eth0 via a dynamic IP address (BOOTP/DHCP), and configures the final, installed system to dynamically determine its IP address. The network command configures the networking information for the installation for network kickstarts as well as for the final, installed system.

`--bootproto`

One of **`dhcp`**, **`bootp`**, or **`static`** (defaults to DHCP, and **`dhcp`** and **`bootp`** are treated the same). Must be **`static`** for static IP information to be used.

`--ip`

IP address for machine to be installed.

`--gateway`

Default gateway as an IP address.

`--nameserver`

Primary name server, as an IP address.

`--netmask`

Netmask for the installed system

There are three different methods of network configuration:

- DHCP
- BOOTP
- static

The DHCP method uses a DHCP server system to obtain its networking configuration. As you might guess, the BOOTP method is similar, requiring a BOOTP server to supply the networking configuration.

The static method requires that you enter all the required networking information in the kickstart file. As the name implies, this information is static, and will be used during the installation, and after the installation as well.

To direct a system to use DHCP to obtain its networking configuration, use the following line:

```
network --bootproto dhcp
```

To direct a machine to use BOOTP to obtain its networking configuration, use the following line in the kickstart file:

```
network --bootproto bootp
```

The line for static networking is more complex, as you must include all network configuration information on one line. You'll need to specify:

- IP address

- netmask
- gateway IP address
- name server IP address

Here's an example static line:

```
network --bootproto static
--ip 10.0.2.15
--netmask 255.255.255.0
--gateway 10.0.2.254
--nameserver 10.0.2.1
```

Please Note

The entire `network` configuration *must* appear on one line! We've wrapped it here to make it easier to read.

There are two restrictions you must keep in mind should you use the static method:

- All static networking configuration information must be specified on *one* line; you cannot wrap lines using a backslash, for example.
- You can only specify one name server here. However, you can use the kickstart file's `%post` section (described in Section F.5.23, *`%post` -- Post-Installation Configuration Section*) to add more name servers, if needed.

F.5.13 `partition`

`part` (required for installs, ignored for upgrades)

Create a partition on the system. Partition requests are of the form:

```
part <mntpoint> --size <size> [--grow] [--onpart <partc>] [--ondisk <disk>
```

The <*mntpoint*> is where the partition will be mounted, and must be of one of the following forms:

/<*mntpoint*>

(i.e. `/`, `/usr`, `/home`)

swap

The partition will be used as swap space.

raid.<*id*>

The partition will be used for software RAID (see the `raid` command later).

--size <*size*>

Sets the minimum size for the partition

--grow

Tells the partition to grow to fill available space (if any).

--onpart <*part*>

Tells the installation program to put the partition on the *already existing* device <*part*>. For example, `partition /home --onpart hda1` will put `/home` on `/dev/hda1`, which must already exist.

--ondisk <*disk*>

Forces the partition to be created on a particular disk. For example, `--ondisk sdb` will put the partition on the second disk on the system.

All partitions created will be formatted as part of the installation process.

F.5.14 `raid`

`raid` (optional)

Assembles a software RAID device. This command is of the form:

```
raid <mntpoint> --level <level> --device
<mddevice><partitions*>
```

The <*mntpoint*> is the location to mount the RAID filesystem. If it is `/`, the RAID level must be 1 unless a boot partition (`/boot`) is present in which the `/boot` partition has to be level 1 and the root (`/`) partition can be any of the available types. The <*partitions**> (which denotes that multiple partitions can be listed) lists the RAID identifiers to add to the RAID array.

--level <*level*>

RAID level to use (0, 1, or 5).

--device <*mddevice*>

Name of the RAID device to use (such as md0 or m1). RAID devices range from md0 to md7, and each may only be used once.

Here's an example of how to create a RAID level 1 partition for `/`, and a RAID level 5 for `/usr`, assuming there are three SCSI disks on the system. It also creates three swap partitions, one on each drive.

```
part raid.01 --size 60 --ondisk sda
part raid.02 --size 60 --ondisk sdb
part raid.03 --size 60 --ondisk sdc

part swap --size 128 --ondisk sda part swap --size 128 --ondisk
sdb part swap --size 128 --ondisk sdc

part raid.11 --size 1 --grow --ondisk sda part raid.12 --size 1
--grow --ondisk sdb part raid.13 --size 1 --grow --ondisk sdc

raid / --level 1 --device md0 raid.01 raid.02 raid.03 raid /usr
--level 5 --device md1 raid.11 raid.12 raid.13
```

F.5.15 `reboot`

`reboot` (optional)

Reboot after the installation is complete (no arguments). Normally, kickstart displays a message and waits for the user to press a key before rebooting.

F.5.16 `rootpw`

`rootpw` (required)

usage: rootpw [--iscrypted] <*password*>

Set the system's root password to the <*password*> argument.

--iscrypted

If this is present, the password argument is assumed to already be encrypted.

F.5.17 `skipx`

`skipx` (optional)

If present, X is not configured on the installed system.

F.5.18 `timezone`

`timezone` (required)

timezone [--utc] <*timezone*>

Sets the system time zone to <*timezone*> which may be any of the time zones listed in "timeconfig".

--utc

If present, the system assumes the hardware clock is set to UTC (Greenwich Mean) time.

F.5.19 `upgrade`

`upgrade` (optional)

Tells the system to upgrade an existing system rather than install a fresh system.

F.5.20 `xconfig`

`xconfig` (optional)

Configures the X Window System. If this option is not given, the user will need to configure X manually during the installation, if X was installed; this option should not be used if X is not installed on the final system.

--noprobe

Don't probe the monitor.

--card <*card*>

Use card <*card*>; this card name should be from the list of cards in Xconfigurator. If this argument is not provided, Anaconda will probe the PCI bus for the card.

--monitor <*mon*>

Use monitor <*mon*>; this monitor name should be from the list of monitors in Xconfigurator. This is ignored if **`--hsync`** or **`--vsync`** is provided; if no monitor information is provided, the monitor is probed via plug-and-play.

--hsync <*sync*>

Specifies the horizontal sync frequency of the monitor.

--vsync <*sync*>

Specifies the vertical sync frequency of the monitor.

--startxonboot

Use a graphical login (runlevel 5) for the installed system.

F.5.21 `zerombr` – Partition table initialization

`zerombr` (optional)

If "zerombr" is specified, and "yes" is its sole argument, any invalid partition tables found on disks are initialized. This will destroy all of the contents of disks with invalid partition tables. This command should be used as:

```
zerombr yes
```

No other format is effective.

F.5.22 `%packages` – Package Selection

Use the `%packages` command to begin a kickstart file section that lists the packages you'd like to install (this is for installations only, as package selection during upgrades is not supported).

Packages can be specified by component or by individual package name. The installation program defines several components that group together related packages. See the `RedHat/base/comps` file on any Red Hat Linux CD-ROM for a list of components. The components are defined by the lines that begin with a number followed by a space, and then the component name. Each package in that component is then listed, line-by-line. Individual packages lack the leading number found in front of component lines.

Additionally, there are three other types of lines in the `comps` file you may run across:

Architecture specific (alpha:, i386:, and sparc64:)

If a package name begins with an architecture type, you only need to type in the package name, not the architecture name. For example:

For `i386: netscape-common` you only need to use the `netscape-common` part for that specific package to be installed.

Lines beginning with ?

Lines that begin with a `?`, are specific to the installation program. You do not have to do anything with these type of lines.

Lines beginning with `--hide`

If a package name begins with `--hide`, you only need to type in the package name, minus the `--hide`. For example:

For `--hide KDE Workstation` you only need to use the `KDE Workstation` part for that specific package to be installed.

In most cases, it's only necessary to list the desired components and not individual packages. Note that the `Base` component is always selected by default, so it's not necessary to specify it in the `%packages` section.

Here's an example `%packages` selection:

```
%packages
@ Networked Workstation
@ C Development
@ Web Server
@ X Window System
bsd-games
```

As you can see, components are specified, one to a line, starting with an `@` symbol, a space, and then the full component name as given in the `comps` file. Specify individual packages with no additional characters (the `bsd-games` line in the example above is an individual package).

Please Note

You can also direct the kickstart installation to use the workstation- and server-class installations. To do this, simply add *one* of the following lines to the `%packages` section:

```
@ Gnome Workstation
@ KDE Workstation
@ Server
```

F.5.23 `%post` – Post-Installation Configuration Section

You have the option of adding commands to run on the system once the installation is complete. This section must be at the end of the kickstart file and must start with the `%post` command. Note, you can access the network in the `%post` section; however, **name service** has not been configured at this point, so only IP addresses will work. Here's an example `%post` section:

```
%post

# add comment to /etc/motd
echo "Kickstart-installed Red Hat Linux ‘/bin/date‘" > /etc/motd

# add another nameserver
echo "nameserver 10.10.0.2" >> /etc/resolv.conf
```

This section creates a message-of-the-day file containing the date the kickstart installation took place, and gets around the `network` command’s "one name server only" limitation by adding another name server to `/etc/resolv.conf`.

Please Note

Note that the post-install script is run in a chroot’ed environment; therefore performing tasks such as copying scripts or RPMs from the installation media will not work.

--nochroot

Allows you to specify commands that you would like to run outside of the chroot’ed environment.

Index

A

accounts
deleting with linuxconf 88
disabling with linuxconf.....87
management 80
modifying 86
adding partitions
with Disk Druid 231
administration, system 23
AMD.....44
anonymous FTP.....50
authentication configuration..... 254
MD5 Password 254
NIS..... 254
Shadow Password..... 254
autoboot 209
automatic partitioning 224

B

BIOS, issues related to LILO 318
boot disk 269
boot options
installation 209
boot process 52
what happens when you boot.....52
bootable CD-ROM..... 209
booting
emergency mode 73
rescue mode 71
a trick 74
single user.....68
single-user mode 73

C

CD-ROM
bootable 209
CD-ROM drive
mounting 133–134
CD-ROM module parameters 276
chkconfig utility.....70
clock 250
component
selecting..... 257
configuration
anonymous FTP.....50
clock 250
ethernet..... 130
hosts 127
network device, adding..... 127
network routes..... 131
NFS 51
pocket network adaptors 130
PPP 129
selecting nameservers 126
system 75
time zone..... 250
token rings 130
X Window System..... 263
XFree86 264
Configuration
Gnome-RPM 169
configuration, hardware..... 198
finding with Windows..... 198
configuration, PLIP

SLIP-PLIP 130
configuration, video 202
configuring console access 32
console access
configuring.....32
defining.....34
disabling 33
disabling all.....33
enabling 35
making files accessible.....34
controlpanel..... 115
Costales, Bryan.....49
create user account
login account, create..... 253
user account, create..... 253
CSLIP 275
Cyrix.....44

D

date
setting 132
dependencies, packages..... 258
destructive partitioning..... 308
`/dev` directory.....23
devices
network, clone..... 129
directories
`/dev` 23
`/etc` 23
`/lib` 24
`/proc`.....24
`/sbin`.....24
`/usr` 25
`/usr/local`..... 25, 27
`/var` 26
disk
boot 269
driver 321
Disk Druid
adding partitions 231
buttons..... 230
current partitions screen..... 228
deleting partitions with..... 234
drive summary screen 229
editing partitions with 233
finishing up 234
function keys 230
problems adding partitions 232
DocBook 20
documentation
PAM 40
documentation CD-ROM 20
driver disk..... 321
produced by Red Hat..... 321
drivers, kernel 275
dual-boot 323
FIPS partitioning tool 328
making room for
adding a new hard drive..... 326
creating new partitions 328
using current partitions or hard drive 326
using FIPS to partition 328
options
booting Red Hat Linux or Windows 323
partitionless installation..... 323

Red Hat Linux as the only OS..... 324
Windows NT warning..... 324
OS/2 326
setting up..... 325

E

enabling accounts.....88
encryption-related features 19
`/etc` directory.....23
`/etc/hosts` file, managing 127
`/etc/pam.conf` 37
`/etc/pam.d`.....37
`/etc/sysconfig`, files in.....55
ethernet..... 130
Ethernet
module parameters..... 285
supporting multiple cards..... 293
expert mode, installation..... 210
exporting NFS filesystems.....51
extended partitions..... 304

F

FAT32 filesystems, accessing 98
fdisk
overview of 237
using..... 236
features, new to 6.2
(See new features)
FHS 23
filesystem
overview of 94
standard 23
structure 23
viewing filesystem with linuxconf .96
filesystem formats, overview of... 296
filesystems
NFS, exporting of.....51
NFS, mounting of.....51
fips partitioning utility 312
floppy group, use of 36
formatting partitions..... 235
FTP
anonymous.....50
ftpaccess.....50
ftphosts 50
ftpusers 51
FTP installation 217

G

Gnome-RPM 161
configuration 169
installing packages..... 166
package display 164
package manipulation 175
querying packages 175
removing packages with 179
selecting packages 165
starting..... 163
uninstalling packages with..... 179
upgrading packages with..... 181
verifying packages..... 178
group management 91
groups 28
creating.....91
deleting.....92
floppy, use of 36
modifying 93

standard 29
user-private 28, 30
rationale 31

H

halt 70
hard disk
basic concepts 295
extended partitions..... 304
filesystem formats 296
partition introduction..... 299
partition types 302
partitioning of 295
hardware configuration..... 198
finding with Windows..... 198
hardware RAID
RAID, hardware..... 335
hostname 126, 245
hosts, managing 127
hosts.allow 50
example 50
hosts.deny 50
HTTP Installation..... 218

I

ident run as daemon 17
impressing friends with RPM 157
information, network 202
information, pre-installation..... 197
init, SysV-style 66
initrd 47
initscript utilities 70
installation
boot options
text mode..... 209
booting without diskette 209
component selection..... 257
finishing 271
keyboard navigation 208
NFS server information 216
package selection 256, 258
text mode..... 209
upgrade 220
installation method
CD-ROM..... 214
FTP 214
hard drive 214
HTTP 214
NFS Image..... 214
selecting..... 212, 220
installation mode
expert 210
serial 210
installation program
text mode user interface..... 205
installation, kickstart
(See kickstart installations)
installation, text mode..... 205
installing packages..... 256
Intel 44

K

kernel 275
building..... 42, 48
custom..... 42, 48
initrd image for.....47
modular..... 42–43
monolithic 48

kernel drivers 275
kernel module (kmod) loader 124
kernel options 210
keyboard navigation, installation
program 208
keyboard type
selecting..... 211
keymap
(See keyboard type)
kickstart
how the file is found..... 346
kickstart file
`auth` 348
`clearpart` 349
`device`..... 349
diskette-based 344
driver disk..... 350
format of 347
`install`..... 350
installation methods 350
`keyboard` 351
lang 352
`lilo` 352
`lilocheck` 353
`mouse`..... 353
`network`..... 354, 356
network-based..... 344
package selection specification 361
post-installation configuration . 362
`raid` 357
`reboot`..... 358
`rootpw`..... 359
`skipx`..... 359
`timezone` 359
`upgrade`..... 359
what it looks like..... 347
`xconfig`..... 360
`zerombr`..... 360
kickstart installations..... 343
diskette-based 344
file format..... 347
file locations 344
network-based..... 344
other commands..... 348
starting..... 345

L

language
selecting..... 211
LDAP
authentication using 189
daemons and utilities 187
files 185
modules for extra functionality 188
more information 192
overview..... 183
pros and cons 184
terminology..... 185
uses for 184
`/lib` directory.....24
LILO
Adding options to..... 239
alternatives to..... 241
commercial products 242
LOADLIN 241
SYSLINUX..... 241
`/etc/lilo.conf` 46
installing 239

installing on boot partition..... 243
installing on MBR 243
skipping 239
SMP Motherboards 245
LILO, BIOS-related issues 318
LILO, partition-related issues 318
linux kernel, 2.2.*x*.....17
linuxconf.....75
account management with 80
account modification 86
changing root password with.....86
changing user's passwords 86
configuring network connections with 107
deleting an account with 88
deleting groups with.....92
disabling account with 87
enabling accounts with.....88
group creation with 91
group management with 91
group modification with 93
modem configuration with..... 101
nameserver specification with.. 110
network configuration with..... 101
NFS mount addition with.....99
overview of 75
PPP configuration with..... 101
PPP/SLIP configuration modification 105
quick reference..... 114
reviewing filesystem.....96
SLIP configuration with 101
tree menu.....78
user interfaces 77
Web access.....79
loading kernel modules 124
LOADLIN 241
local media installations 212
`ls` command 20

M

master boot record
(See MBR)
Maximum RPM..... 159
MBR
installing LILO on 243
Mesa 18
modems
configuring with linuxconf..... 101
module parameters..... 275
modules
PAM 36
mount points and partitions..... 316
mounting CD-ROM drive..... 134
mounting NFS filesystems.....51
mouse
configuring..... 248
mtools and the floppy group.....36

N

nameserver
selecting..... 126
nameservers, specifying
using linuxconf 110
naming your computer 245
network
interface

aliasing 126
network adapters, pocket 130
network configuration..... 125
adding device..... 127
with linuxconf..... 101
network connections
configuring using linuxconf..... 107
network devices
clone 129
network information..... 202
network installations
FTP 217
HTTP 218
network routes, managing 131
networking..... 127
new features
client and server network services.....15
colorized `ls`command 20
disabled services 18
DocBook 20
documentation CD-ROM 20
encryption-related.....19
fonts, automatic recognition of...19
guru 15
ident run as daemon 17
installation-related
(See *Official Red Hat Linux Installation Guide*)
kernel, 2.2.*x*.....17
man pages, compression of.....18
Mesa graphics library 18
Migration to PowerTools 19
miscellaneous.....19
sawmill window manager.....18
security improvements 17
sysctl, usage of 16
System-related.....15
termcap, terminfo changes 20
X startup, changes to 18
XFree86 3.3.5 17
NFS
configuration 51
exporting 51
mounting 51
NFS mounts
adding with linuxconf.....99
non-destructive partitioning 309
ntsysv utility.....70

O

O'Reilly & Associates 49, 52
online help
text mode installation..... 209
OpenLDAP 183
options, kernel..... 210
OS/2 ..... 243, 315

P

package
installation screen..... 261
package manipulation, with Gnome-RPM 175
packages
dependencies 152, 258
determining file ownership with 157

finding deleted files from 157
freshening with RPM..... 154
Gnome-RPM 166
handy hints..... 157
installing 151, 256
locating documentation for 158
obtaining list of files..... 159
preserving config files..... 153
querying..... 155
querying uninstalled..... 158
removing 152
selecting..... 256
selecting individual 258
uninstalling with Gnome-RPM 179
upgrading 153
upgrading with Gnome-RPM .. 181
verifying..... 156
verifying with Gnome-RPM... 175, 178
PAM 36
additional information.....40
configuration files.....37
modules 36
rexec, access to.....40
services.....37
parameters
CD-ROM module..... 276
Ethernet modules 285
parameters, module..... 275
partition
/boot..... 317
MILO..... 318
root..... 318
swap 317
partitioning
auto-partitioning 224
basic concepts 295
destructive 308
introduction to..... 299
LILO issues related to..... 318
non-destructive..... 309
other operating systems 315
using free space 306
using in-use partition 308
using unused partition..... 307
partitions
basic concepts 295
changing table of..... 239
creating..... 226
extended..... 304
formatting..... 235
how many 316
introduction to..... 299
making room for 305
mount points relation with..... 316
naming of 313
numbering of 313
types of 302
using fdisk 236
password
root, setting 251
passwords
changing.....86
shadow 40
PLIP 275
PLIP interface 130
pluggable authentication modules
(See PAM)

pocket network adapters 130
PowerTools 133
installing
GNOME or KDE 133
in a GUI environment 133
shell prompt 134
reading the `CONTENTS` file 133
PowerTools, packages migrated to .19
PPP 129, 275
PPP/SLIP configuration
with linuxconf..... 101
PPP/SLIP configuration modification
using linuxconf 105
pre-installation information 197
printer configuration..... 117
LAN manager 123
local..... 120
NCP..... 123
NetWare..... 123
remote 121
SMB 123
test page 124
`/proc` directory 24
processor
AMD.....44
Cyrix.....44
Intel 44
programs, running at boot time 70

Q

querying packages with
Gnome-RPM 175

R

RAID 335
creating partitions..... 338
explanation of 335
hardware RAID 335
kernel features..... 336
level 0..... 337
level 1..... 337
level 4..... 337
level 5..... 337
levels..... 337
reasons to use..... 335
software RAID 335
`rc.local`, modifying 70
Red Hat Package Manager
(See RPM)
Red Hat-specific file locations 27
removing packages with
Gnome-RPM 179
rescue mode 71
a handy trick.....74
definition of.....71
from CD, diskette, network,
PCMCIA 72
using 72
utilities available 72
rexec, access to.....40
root password..... 251
changing.....86
routes, managing..... 131
RPM 149
book written about..... 159
dependencies 152
design goals..... 149

determining file ownership
with 157
documentation with..... 158
file conflicts, resolving..... 152
finding deleted files with..... 157
freshen..... 154
freshening packages 154
handy hints..... 157
installing 151
mailing list devoted to..... 160
other resources 159
preserving config files..... 153
querying..... 155
querying for file list 159
querying uninstalled packages . 158
uninstalling 152
upgrading 153
using 151
verifying..... 156
website devoted to 160

S

sawmill window manager.....18
/sbin directory.....24
SCSI 275
selecting components..... 257
selecting packages 256
selecting packages, Gnome-RPM 165
sendmail.....48
aliases.....49
masquerading.....49
with IMAP.....48
with UUCP 48
serial mode, installation..... 210
services
PAM 37
system
starting with chkconfig.....70
starting with ntsysv 70
services, controlling access to.....50
shadow passwords 40
shadow utilities.....41
shutdown.....70
SLIP 275
SLIP interface 130
SMP Motherboards
LILO..... 245
software RAID
RAID, software 335
standard groups 29
standard users 28
starting installation 209
striping
RAID fundamentals 335
structure, filesystem 23
SYSLINUX..... 241
system administration 23
System Commander 242
system configuration
with linuxconf.....75
system shutdown.....70
SysV init 66
directories used by.....67
runlevels used by.....69

T

TCP wrapper 50
TCP/IP networking 247

tcpd 50
termcap and terminfo, changes to 20
test page, printer 124
text mode installation
online help 209
user interface 205
time and date
setting 132
time zone, setting 250
token ring 130

U

uninstalling packages with
Gnome-RPM 179
Update Agent..... 137
configuring..... 138
starting..... 137
using 144
Updating
(See Update Agent)
upgrade..... 220
upgrading packages with
Gnome-RPM 181
user interface
text mode installation..... 205
user-private groups 28, 30
user-private groups, rationale
behind 31
users 28
adding 80
standard 28
/usr directory.....25
/usr/local directory..... 25, 27
utilities
shadow 41

V

/var directory.....26
verifying packages with
Gnome-RPM 178
video configuration 202

W

Windows
finding hardware configuration
with 198

X

X Windows, configuring..... 263
Xconfigurator..... 264
XFree86 17
configuration 264