Red Hat Linux 6.2

The Official Red Hat High Availability Server Installation Guide

ISBN: N/A

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

HighAvail(EN)-1.0-Print-RHI ($Date: 2000/06/20 16:18:27 $)

Printed in the United States, Ireland, and Japan

Contents

Red Hat Linux 6.2

Introduction

Thank you for purchasing Red Hat High Availability Server 1.0. With this product, it is possible to create solutions that can withstand many common hardware and software failures, and still provide service to your customers. In addition, because Red Hat High Availability Server allows more than one computer to work together to service your customers' needs, maintenance and upgrades can be planned and executed without an interruption in service.

This manual will guide you through the following steps in deploying a solution based on Red Hat High Availability Server:

- Begin to learn more about the underlying technology in this introduction, so that you can start to think about the configuration that would best fit your needs.
- Learn how to install Red Hat Linux 6.2 (the operating system on which Red Hat High Availability Server is based).
- Learn about the **Failover Services** (FOS) technology. FOS is used to create pairs of Linux-based systems that provide the services used by your customers, as well as implementing a backup system that automatically takes control at the first sign of problems.
- Learn about the **Linux Virtual Server** (LVS) technology. Like FOS, LVS can be used to create pairs of Linux-based systems that act as each other's backup. However, LVS goes a step further. Instead of the two Linux-based systems actually providing your services, they instead monitor and load-balance a pool of systems (that may or may not be Linux-based), making possible a wider array of performance and capacity options.
- Learn about the Piranha Web Interface, which uses an intuitive graphical interface to configure your Piranha system.

Let's start by taking a quick look at the technology behind Red Hat High Availability Server.

Technology Overview

Red Hat High Availability Server uses Piranha to implement highly-available solutions. Piranha is a collection of programs that interact with each other to provide a clustering solution. It is vital to note that **cluster** computing consists of two distinct branches:

- **Compute clustering** (such as beowulf) uses multiple machines to provide greater computing power for computationally-intensive tasks. This type of clustering is not addressed by Piranha.
- **High Availability** (or HA) clustering uses various technologies to gain an extra level of reliability for a service. HA clustering is the focal point for Piranha.

Please Note

The clustering technology described in this document should *not* be confused with fault tolerance. Fault tolerant systems use highly-specialized (and expensive) hardware to implement a fully-redundant environment in which services can run, uninterrupted by the most common failure modes.

Red Hat High Availability Server is designed to run on readily-available hardware and to take proactive measures when a system fault is detected. This results in an environment that approaches (but does not reach) the availability of fault tolerant systems, but at a fraction of the cost.

Let's take a look at some sample configurations, using both FOS and LVS as the base technology. Note that the boxes in the following diagrams (and the terms used to describe them) designate *roles* rather than specific systems. Also keep in mind that, due to the variety of ways in which the underlying technologies may be deployed, you will find that various terms may be used interchangably throughout this and other cluster-related documents. Although a cluster may be configured such that each role is carried out by a dedicated system, there is no technological requirement for this.

However, capacity planning or system administration-related issues may dictate that dedicated systems be used for the various roles in a given cluster.

Please Note

Due to the variety of ways in which the underlying technologies may be deployed, you will find that various terms may be used interchangably throughout this and other documents. While every attempt has been made to define each term as it is first used,

Sample Configurations

While Red Hat High Availability Server can be configured in a variety of different ways, the configurations can be broken into two major categories:

- Configurations based on FOS
- Configurations based on LVS

The choice of FOS or LVS as the underlying technology for your Red Hat High Availability Server cluster has an impact on the hardware requirements and the service levels that can be supported. Let's take a look at some sample configurations, and discuss the implications of each.

FOS Configurations

Red Hat High Availability Server clusters based on FOS consist of two Linux systems. Each system must be sized to support the full load anticipated for all services [1]. This is necessary, because at any given time only one system (the **active node**) provides the services to your customers.

During the initial configuration of an FOS cluster, one system is defined as the **primary node**, and the other as the **backup node**. This distinction is made to determine

[1] Although there is no technological requirement that each system be identically configured (only that each system be capable of supporting all services), from a system administration perspective it makes a great deal of sense to configure each system identically.

which system will be declared active should both systems find themselves in the active state at the same time. In such a case, the primary node will "win".

The active node responds to service requests through a **virtual IP** (or VIP) address. The VIP address is an IP address that is distinct from the active node's normal IP address.

The other system (the **inactive node**) does not actually run the services; instead it monitors the services on the active node, and makes sure the active node is still functional. If the inactive node detects a problem with either the active node or the services running on it, a failover will be initiated.

During a failover, the following steps are performed:

- The active node is directed to shut down all services (if it is still running and has network connectivity)
- The inactive node starts all services
- The active node disables its use of the VIP address (if it is still running and has network connectivity)
- The inactive node is now the active node, and enables its use of the VIP address
- If it is still running and has network connectivity, the formerly-active node is now the inactive node, begins monitoring the services and general well-being of the active node

Let's take a look at the most basic of FOS configurations.

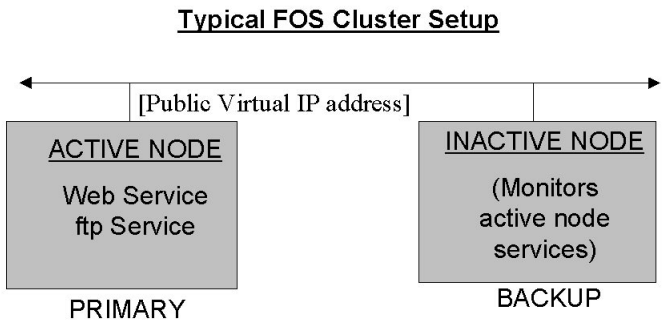
A Basic FOS Configuration

Figure 1, *A Simple FOS Configuration* shows a Red Hat High Availability Server FOS cluster. In this case, the active node provides Web and FTP services, while the inactive node monitors the active node and its services.

While not shown in this diagram, it is possible for both the active and inactive nodes to be used for other, less critical, services while still performing their roles in an FOS cluster. For example, the inactive system could run inn to manage an NNTP newsfeed. However, care must be taken to either keep sufficient capacity free should a failover occur, or to select services that could be shut down (with little ill effect)

in the event of a failover. Again, from a system administration standpoint, it may be desirable to dedicate the primary and backup nodes to cluster-related services, even if this does result in less then 100% utilization of system resources.

Figure 1 A Simple FOS Configuration



One aspect of an FOS cluster is that the capacity of the entire cluster is limited to the capacity of the currently-active node. This means that bringing additional capacity online will require upgrades (or replacements) for each system. While the FOS technology means that such upgrades can be done without impacting service availability, doing upgrades in such an environment means a greater exposure to possible service outages, should the active node fail while the inactive node is being upgraded or replaced.

It should also be noted that FOS is *not* a data-sharing technology. In other words, if a service reads and writes data on the active node, FOS includes no mechanism to replicate that data on the inactive node. This means data used by the services on an FOS cluster must fall into one or two categories:

- The data does not change; it is read-only

- The data is not read-only, but it is made available to both the active and inactive node equally.

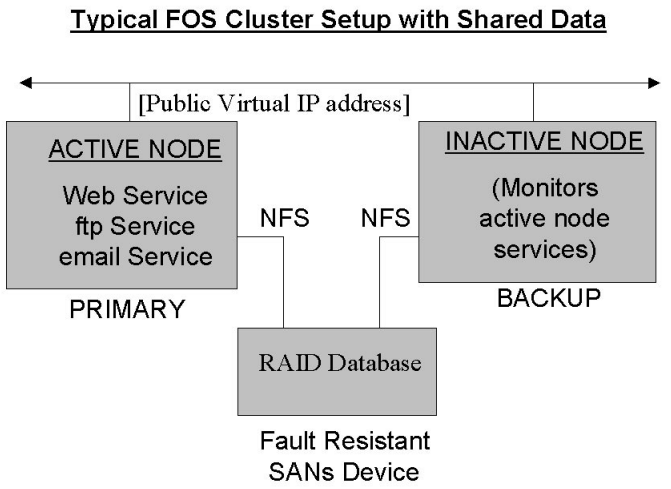
While the sample configuration shown above might be appropriate for a Website containing static pages, it would not be appropriate for a busy FTP site, particularly one where new files are constantly uploaded by users of the FTP site. Let's look at one way an FOS cluster can use more dynamic data.

An FOS Configuration With Shared Data

As noted above, some mechanism must be used to make read-write data available to both nodes in an FOS cluster. One such solution is to use NFS-accessible storage. By using this approach, failure of the active node will not result in the data being inaccessible from the inactive node.

However, care must be taken to prevent the NFS-accessible storage from becoming a single point of failure. Otherwise, the loss of this storage would result in a service outage, even if both the active and inactive nodes were otherwise healthy. The solution, as shown in Figure 2, *An FOS Configuration With Shared Data*, is to use RAID and other technologies to implement a fault-resistant NFS server.

Figure 2 An FOS Configuration With Shared Data



While it's certainly possible that various modifications to the basic FOS cluster configuration are possible, in general, the options are limited to one system providing all services, while a backup system monitors those services, and initiates a failover if and when required. For more flexibility and/or more capacity, an alternative is required. That alternative is LVS.

Typical LVS Configurations

In many ways, an LVS cluster can be thought of as an FOS cluster with a single difference — instead of actually providing the services, the active node in an LVS cluster *routes* requests to one or more servers that actually provide the services. These additional servers (called **real servers**) are load-balanced by the active node (in LVS terms, the **active router**).

As in an FOS cluster, there are two systems that share the responsibility of ensuring that one of the two systems is active, while the other (the **inactive router**) stands by to initiate a failover should the need arise. However, that is where the similarities end.

Because the active router's main responsibility is to accept incoming service requests and direct them to the appropriate real server, it is necessary for the active router to keep track of the real servers' status, and to determine which real server should receive the next inbound service request. Therefore, the active router monitors every service on each real server. In the event of a service failure on a given real server, the active router will stop directing service requests to it until the service again becomes operative.

In addition, the active router can optionally use one of several scheduling metrics to determine which real server is most capable of handling the next service request. The available scheduling metrics are:

- Round robin
- Least connections
- Weighted round robin
- Weighted least connections

We'll go into more detail regarding scheduling in Chapter 8, *Linux Virtual Server (LVS)*; however, for now the important thing to realize is that the active router can take into account the real servers' activity and (optionally) an arbitrarily-assigned weighting factor when routing service requests. This means that's it's possible to create a group of real servers using a variety of hardware configurations, and have the active router load each real server evenly.

A Basic LVS Configuration

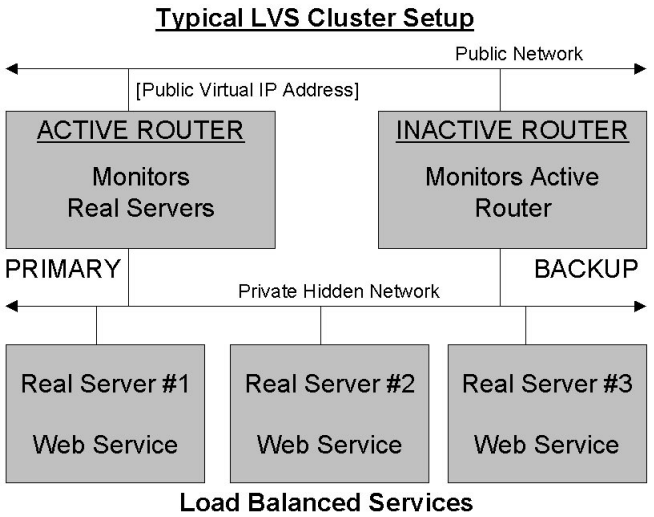
Figure 3, *A Typical LVS Configuration* shows a typical Red Hat High Availability Server LVS cluster with three real servers providing Web service. In this configuration, the real servers are connected to a dedicated private network segment. The routers have two network interfaces:

- One interface on the public network
- One interface on a private network

Service requests directed to the cluster's VIP address are received by the active router on one interface, and then passed to the most appropriate (as determined by the schedule/weighting algorithm in use) real server through the other interface. In this way,

the routers are also able to act the part of firewalls; passing only valid service-related traffic to the real servers' private network.

Figure 3 A Typical LVS Configuration



An LVS cluster may use one of three different methods of routing traffic to the real servers:

- Network Address Translation (NAT), which enables a private-LAN architecture
- Direct routing, which enables LAN-based routing
- Tunneling (IP encapsulation), which makes possible WAN-level distribution of real servers

In Figure 3, *A Typical LVS Configuration*, NAT routing is in use. While NAT-based routing makes it possible to operate the real servers in a more sheltered environment, it does exact additional overhead on the router, as it must translate the addresses of all

traffic to and from each real server. In practice, this limits the size of a NAT-routed LVS cluster to approximately ten to twenty real servers [2].

This overhead is not present when using tunneled or direct routing, because in these routing techniques the real servers respond directly to the requesting systems. With tunneling there is a bit more overhead than with direct routing, but it is minimal, especially when compared with NAT-based routing.

One interesting aspect of an LVS-based cluster is that the real servers do not have to run a particular operating system. Since the routing techniques used by LVS are all based on industry-standard TCP/IP features, any platform that supports a TCP/IP stack can conceivably be part of an LVS cluster [3].

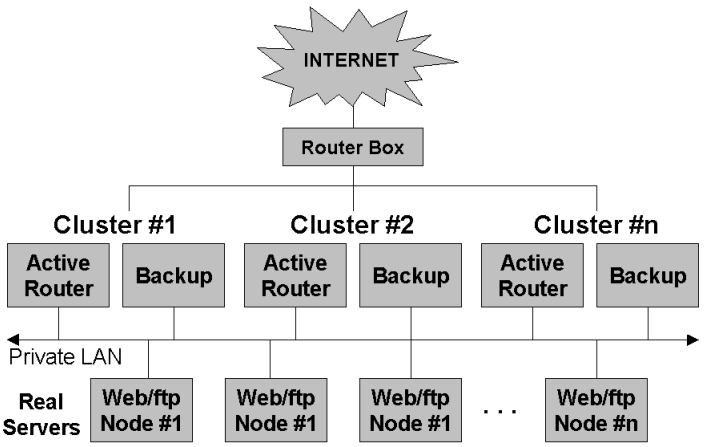
A More Complex LVS Configuration

Figure 4, *A More Complex LVS Configuration* shows a more esoteric approach to deploying LVS. This configuration shows a possible approach for deploying clusters comprised of an extremely large number of systems. The approach taken by this configuration is to share a single pool of real servers between multiple routers (and their associated backup routers).

[2] It is possible to alleviate the impact of NAT-based routing by constructing more complex cluster configurations. For example, a two-level approach may be created where the active router load-balances between several real servers, each of which is itself an active router with its own group of real servers actually providing the service.

[3] The scheduling options may be somewhat restricted if a real server's operating system does not support the `uptime`, `ruptime`, or `rup` commands. These commands are used to dynamically adjust the weights used in some of the scheduling methods.

Figure 4 A More Complex LVS Configuration



Because each active router is responsible for routing requests directed to a unique VIP address (or set of addresses), this configuration would normally present the appearance of multiple clusters. However, **round-robin DNS** can be used to resolve a single hostname to one of the VIP addresses managed by one of the active routers. Therefore, each service request will be directed to each active router in turn, effectively spreading the traffic across all active routers.

Acknowledgements

The following people contributed their time, effort, and documents to make this manual possible:

- Keith Barrett — The *Piranha — Failover Services (FOS)* white paper
- Philip Copeland — The *Piranha Web GUI — Features and Configuration* white paper
- Sandra Moore — The *Official Red Hat Linux Installation Guide*

- Wayne Sherrill — The *LVS Cluster Configuration HOWTO*

Thank you all — without your assistance and insightful documents, producing this manual would have been much more difficult!

Part I Installing Red Hat High Availability Server

1 New Features of Red Hat Linux 6.2

Before you can use Red Hat High Availability Server, you must first install Red Hat Linux. The following chapters will guide you through this process.

This chapter describes features that are new to the Red Hat Linux 6.2 graphical installation process. To learn about non-installation-related new features, please refer to the *Official Red Hat Linux Reference Guide*.

1.1 Installation-Related Enhancements

Improvements to Red Hat Linux 6.2 which will make installation even easier include:

Additional GUI Partitioning Tool

Previously available only in expert mode, fdisk has been added to the GUI installation. You can now choose to partition with Disk Druid or fdisk, depending on your level of skill and personal preference.

Rescue Disk Improvements

New and improved options make using the rescue disk even more powerful than before. Improvements include mtools and RAID tools, and pico as the new default editor.

Software RAID Configuration in Kickstart Installations

New to kickstart installations is the ability to configure RAID.

RAID Upgrades

New to the installation program is the ability to perform RAID upgrades.

ATAPI Zip Drive Recognition

ATAPI Zip drives are now recognized by the installation program and automatically configured to use SCSI emulation. If you add an ATAPI Zip drive after the installation, the hardware recognition program kudzu will recognize it once you reboot your system. The installation program will also create device files for Jaz drives as well.

2 Before You Begin

This chapter explains how to prepare for the Red Hat Linux installation. It's divided into two main sections:

- Seven steps to get you ready for the installation (such as checking for errata, hardware compatibility, making diskettes and more);
- System requirements table for gathering your hardware information.

While installing Red Hat Linux is a fairly straightforward process, taking time to prepare for it will make things go much more smoothly. In this chapter, we'll discuss the steps you should perform prior to the installation.

If you are an experienced user and do not need a review of the basics, you can skip ahead to Chapter 3, *Starting the Installation* to begin the installation process.

Tip

Refer to the Red Hat Frequently Asked Questions for answers to questions and problems that may occur before, during or after the installation. You'll find the FAQ online at: http://www.redhat.com/support/docs/faqs/rhl_general_faq/FAQ.html

---

2.1 Seven Steps to Get You Started

There are seven steps you should perform prior to installing Red Hat Linux:

2.1.1 Step 1 - Do You Have the Right Red Hat Linux Components?

If you've purchased the Official Red Hat Linux boxed set, you're ready to go! However, mistakes occasionally happen, so now is a good time to double-check the contents of your boxed set.

In your Red Hat High Availability Server box, there is a Terms and Conditions sheet. On the back is a list of the contents of your boxed set version. Please read over this list and check to make sure that you have all the diskettes and manuals that are available with your version.

If you've purchased the Official Red Hat High Availability Server boxed set from Red Hat, Inc. (or one of its distributors), and you're missing one or more of the items listed, please let us know!

In certain cases, you may need to create a boot diskette. For information on making diskettes, see *Making Installation Diskettes* in Section 2.1.6.

2.1.2 Step 2 - Is Your Hardware Compatible with Red Hat Linux 6.2?

Hardware compatibility is particularly important to those of you with older systems or systems that you may have built yourself. Red Hat Linux 6.2 should be compatible with most hardware in systems that were factory built within the last two years. However, with hardware specifications changing and improving almost daily, it is hard to guarantee that your hardware will be 100% compatible.

First, use Red Hat's online resources to make sure your hardware is compatible and/or supported. You'll find the hardware compatibility list at: http://www.redhat.com/hardware.

Second, gather all the system hardware information you can; the *Official Red Hat Linux Reference Guide* has instructions on doing this in the *Installation-Related Reference*. At the end of this chapter, a system requirements table (see Section 2.2, *System Requirements Table*) is available for you to fill out and reference during the installation.

2.1.3 Step 3 - Have You Checked for Errata?

Although most of the time it's not necessary to check for errata before the installation, it is also not a bad idea, either.

Red Hat offers updated diskette images, documentation and other errata downloads for your convenience.

There are two ways to review the errata:

1. Online — http://www.redhat.com/support/errata; supplies errata you can read online, and you can download diskette images easily.
2. E-mail — By sending an empty mail message to `errata@redhat.com`, you will receive an e-mail containing a text listing of the complete errata of the installation program and software itself (if errata exist at that time). Also included are URLs to each updated package and diskette image in the errata. Using these URLs, you can download any necessary diskette images. Please note: use binary mode when transferring a diskette image.

Occasionally, we find that the installation may fail, and that a revised diskette image is needed for the installation to work properly. In these cases, we make special images available via the Red Hat Linux errata listing.

Since this is relatively rare, you will save time if you try to use the standard diskette images first. Review the errata only if you experience problems completing the installation.

If you experience problems, focus on entries that include new diskette images (the filenames always end in `.img`). If you find an entry that applies to your problem, get a copy of the diskette images, and create them using the instructions in *Making Installation Diskettes* in Section 2.1.6.

Also available are documentation errata. When significant changes are made to the manuals, we make sure to update these online as well. Documentation updates can be found at http://www.redhat.com/support/errata/doc_errata/.

2.1.4 Step 4 - Do You Have Enough Disk Space?

Nearly every modern-day operating system uses **disk partitions**, and Red Hat Linux is no exception. When installing Red Hat Linux, it may be necessary to work with disk partitions. If you have not worked with disk partitions before (or would like a quick review of the basic concepts) please read *An Introduction to Disk Partitions* in the appendix of the *Official Red Hat Linux Reference Guide* before proceeding.

If you are not performing a "fresh" installation, in which Red Hat Linux will be the only OS on your system, and you are not performing an upgrade, you will need to

make sure you have enough available **disk space** on your hard drive(s) for this installation.

This disk space must be separate from the disk space used by other OSes you may have installed on your system, such as Windows, OS/2, or even a different version of Linux. This is done by dedicating one or more partitions to Red Hat Linux.

Before you start the installation process, one of the following conditions must be met:

- Your computer must have enough *unpartitioned* disk space available to install Red Hat Linux.
- Your computer must have one or more partitions that may be deleted, thereby freeing up enough disk space to install Red Hat Linux.
- You must have a preexisting, formatted FAT partition, and install using the partitionless installation method (Appendix D, *Installing Without Partitioning*).

You'll need about 1.2GB for the basic installation, and 1.7GB if you select everything.

If you are not sure that you meet these conditions or want to know how to free up more space for your Red Hat Linux installation, please refer to the partitioning appendix in the *Official Red Hat Linux Reference Guide*.

2.1.5 Step 5 - How Do You Want to Install Red Hat Linux?

Next, you must decide which type of installation best fits your needs. Options include:

CD-ROM

If you purchased a Red Hat Linux 6.2 boxed set (or have a Red Hat Linux CD-ROM) and have a CD-ROM drive. This method requires a boot disk, a bootable CD-ROM, or a PCMCIA boot disk.

Hard Drive

If you have copied the Red Hat Linux files to a local hard drive. This method requires a boot disk or PCMCIA boot disk.

NFS Image

If you are installing from an NFS Image server which is exporting the Red Hat Linux CD-ROM or a mirror image of Red Hat Linux. Requires a network or PCMCIA boot disk.

FTP

If you are installing directly from an FTP server. Requires a network or PCMCIA boot disk.

HTTP

If you are installing directly from an HTTP Web server. Requires a network or PCMCIA boot disk.

2.1.6 Step 6 - How Do You Want to Start the Installation?

Depending on the installation method you chose in Step 5, you must decide how you want to start the installation process itself. What **boot media** will you use?

Bootable CD-ROM

If your system will allow you to boot from your CD-ROM drive, you can use the Red Hat Linux CD-ROM to boot into the installation program to perform a local CD-ROM installation.

Local Media Boot Disk

You will find a **local boot disk** in the box. This diskette can be used for CD-ROM installations for which your CD-ROM drive is not bootable, or for a hard drive installation.

Network Boot Disk

If you are performing an installation via FTP, HTTP, or NFS you must create your own **network boot disk**. The network boot disk image file is `bootnet.img`, and is located in the `images` directory on your Red Hat Linux/Intel CD.

PCMCIA Boot Disk

Here's a checklist to help you determine if you'll need to create a **PCMCIA boot disk**:

- If you'll be installing Red Hat Linux from a CD-ROM, and your CD-ROM drive is attached to your computer through a PCMCIA card, you'll need a PCMCIA boot disk.
- If you will be using a PCMCIA network adapter during the installation, you may need a PCMCIA boot disk.

If you need a PCMCIA boot disk, you must make one. The PCMCIA boot disk image file is `pcmcia.img`, and is located in the `images` directory on your Red Hat Linux/Intel CD.

Making Installation Diskettes

It is sometimes necessary to create a diskette from an **image file**; for example, you may need to use updated diskette images obtained from the Red Hat Linux errata page or you may need to create a boot disk.

An image file contains an exact copy (or image) of a diskette's contents. Since a diskette contains filesystem information in addition to the data contained in files, the image file is not usable until it has been written to a diskette.

To start, you'll need a blank, formatted, high-density (1.44MB), 3.5-inch diskette. You'll need access to a computer with a 3.5-inch diskette drive, and capable of running an MS-DOS program, or the `dd` utility found on most Linux-like operating systems.

The `images` directory on your Red Hat Linux CD contains the boot images for Red Hat Linux/Intel.

Once you've selected the proper image, it's time to transfer the image file onto a diskette.

Making a Diskette Under MS-DOS

To make a diskette under MS-DOS, use the `rawrite` utility included on the Red Hat Linux CD in the `dosutils` directory. First, label a blank, formatted 3.5-inch diskette appropriately (such as "Boot Disk" or "Updates Disk"). Insert it into the diskette drive. Then, use the following commands (assuming your CD is drive `d:`):

```
C:\> d:
D:\> cd \dosutils
```

```
D:\dosutils> rawrite
Enter disk image source file name: ..\images\boot.img
Enter target diskette drive: a:
Please insert a formatted diskette into drive A: and
press --ENTER-- : [Enter]
D:\dosutils>
```

First, `rawrite` asks you for the filename of a diskette image; enter the directory and name of the image you wish to write (for example, `..\images\boot.img`). Then `rawrite` asks for a diskette drive to write the image to; enter `a:`. Finally, `rawrite` asks for confirmation that a formatted diskette is in the drive you've selected. After pressing [Enter] to confirm, `rawrite` copies the image file onto the diskette. If you need to make another diskette, label that diskette, and run `rawrite` again, specifying the appropriate image file.

Making a Diskette Under a Linux-Like OS

To make a diskette under Linux (or any other Linux-like operating system), you must have permission to write to the device representing a 3.5-inch diskette drive (known as `/dev/fd0` under Linux).

First, label a blank, formatted diskette appropriately (such as"Boot Disk," "Updates Disk"). Insert it into the diskette drive (but don't issue a `mount` command). After mounting the Red Hat Linux CD, change directory to the directory containing the desired image file, and use the following command (changing the name of the image file and diskette device as appropriate):

```
# dd if=boot.img of=/dev/fd0 bs=1440k
```

If you need to make another diskette, label that diskette, and run `dd` again, specifying the appropriate image file.

2.1.7 Step 7 - Which Installation Type is Best For You?

Red Hat Linux includes three different classes, or types of installations. They are:

- `Cluster Server` — This is the default installation class.
- `Custom` — Allows you to install additional software, including the X Window System, the KDE and GNOME desktop environments, and more.

- `Upgrade` — Upgrades a Red Hat Linux system to the latest version.

Please Note

The Upgrade installation class will *NOT* turn an existing Red Hat Linux system into a high availability server! You *must* perform a fresh Red Hat Linux installation in order to properly install Red Hat High Availability Server!

These classes give you the option of simplifying the installation process (with some potential for loss of configuration flexibility), or retaining flexibility with a slightly more complex installation process. Let's take a detailed look at each class, so you can see which one is right for you.

The Cluster Server-Class Installation

During the cluster server-class installation, 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.

What Does It Do?

If you choose *not* to partition manually, a cluster server-class installation removes *ALL existing partitions* on *ALL installed hard drives*, so choose this installation class only if you're sure you have nothing you want saved! When the installation is complete, you'll find the following partitions:

- A 64MB swap partition.
- A 256MB partition (mounted as `/`).
- A partition of at least 512MB (mounted as `/usr`).
- A partition of at least 512MB (mounted as `/home`).
- A 256MB partition (mounted as `/var`).

- A 16MB partition (mounted as `/boot`) in which the Linux kernel and related files are kept.

This approach to disk partitioning results in a reasonably flexible filesystem configuration for most server-class tasks.

Please Note

You will need at least 1.2GB of free disk space in order to perform a cluster server-class installation.

A server-class installation will remove *ALL existing partitions of ANY type on ALL existing hard drives of your system*. All drives will be erased of all information and existing operating systems, regardless if they are Linux partitions or not!

The Custom-Class Installation

As you might guess from the name, a custom-class installation puts the emphasis on flexibility. During a custom-class installation, *you* can choose how disk space should be partitioned. You have complete control over which packages will be installed on your system. You also determine whether you'll use LILO (the LInux LOader) to boot your system.

Behind the Scenes of a Custom-Class Installation

This section covers those installation steps that are *only* seen when performing a custom-class installation.

This may help those of you who are trying to decide which installation class will better suit your needs. If you think you'll have trouble performing any of the tasks on

this list, you should not perform a custom-class installation without reading through this manual and clarifying any questions you may have.

- Creating Partitions — In the custom-class installation it is necessary for you to specify where you want Red Hat Linux to be installed. (This is no longer specific to custom-class installations because you now have the *option* to manually partition in the workstation- and server-class installations.)
- Formatting Partitions — All newly created partitions must be formatted. Any partitions that contain old data (data you no longer need or want) should be formatted. (If you chose to manually partition your workstation- or server-class installation, you will need to choose which partitions to format.)
- Selecting and Installing Packages — This is performed after your partitions have been configured and selected for formatting. Here you may select groups of packages, individual packages, a combination of the two, or choose an "everything" install.
- LILO Configuration — In a custom-class installation, you are able to choose where you would like LILO to be installed — either on the master boot record (MBR) or on the first sector of your root partition — or you can choose not to install LILO at all.

Upgrading Your System

Please keep in mind that using your Red Hat High Availability Server CD to upgrade an existing Red Hat Linux system will *not* turn your system into a high availability server. However, upgrading you Red Hat Linux 2.0 (or greater) sytem will not delete any existing data. The installation program updates the modular 2.2.x kernel and all currently installed software packages. See Chapter 3, *Starting the Installation* and Chapter 5, *Upgrading Your Current System* for those instructions.

2.2 System Requirements Table

Use the space provided to fill in your system settings and requirements. This will help you keep a record of your current system, as well as make the installation process easier.

Table 2–1 System Requirements

Hard Drive(s): Number, size, type; ex: IDE hda=1.2G	1)
Partitions: map of partitions and mount points; ex: `/dev/hda1=/home`, `/dev/hda2=/` (fill this in once you know where they will reside).	2)
Memory: Amount of RAM installed on your system; ex: 64MB, 128MB	3)
CD-ROM: Interface Type; ex: SCSI, IDE (ATAPI)	4)
SCSI Adapter: If present, make and model number; ex: BusLogic SCSI Adapter, Adaptec 2940UW	5)
Network Card: If present, make and model number; ex: Tulip, 3COM 3C590	6)

Mouse: Type, protocol, and number of buttons; ex: generic 3 button PS/2 mouse, MouseMan 2 button serial mouse	7)
Monitor: Make, model, and manufacturer specifications; ex: Optiquest Q53, ViewSonic G773	8)
Video Card: Make, model number and VRAM; ex: Creative Labs Graphics Blaster 3D, 8MB	9)
Sound Card: Make, chipset and model number; ex: S3 SonicVibes, Sound Blaster 32/64 AWE	10)
IP Address: Four numbers, separated by dots; ex: 10.0.2.15 (*contact your netadmin for help*)	11)

Netmask: Usually four numbers, separated by dots; ex: 255.255.248.0 (*contact your netadmin for help*)	12)
Gateway IP address: Four numbers, separated by dots; ex: 10.0.2.245 (*contact your netadmin for help*)	13)
One or more name server IP Addresses: Usually one or more sets of dot-separated numbers; ex: 10.0.2.1 (*contact your netadmin for help*)	14)
Domain name: the name given to your organization; ex: Red Hat's would be `redhat.com` (*contact your netadmin for help*)	15)
Hostname: the name of your computer; your personal choice of names ex: `cookie`, `southpark`.	16)

3 Starting the Installation

This chapter explains how to start the Red Hat Linux installation process. We'll cover the following areas:

- Getting familiar with the installation program's user interface;
- Starting the installation program;
- Selecting an installation method;
- Beginning the installation.

By the end of this chapter, the installation program will be running on your system, and you will have begun the process of either installing or upgrading to Red Hat Linux 6.2.

3.1 The Installation Program User Interface

If you've used a **graphical user interface (GUI)** before, you'll be familiar with this process. If not, simply use your mouse to navigate the screens, "click" buttons or enter text fields. You can also navigate through the installation using the [Tab] and [Enter] keys.

Please Note

If you do not wish to use the GUI installation program, the text mode installation program is also available. To enter text mode, enter the following boot command:

```
boot: text
```

For text mode installation instructions, please refer to the *Official Red Hat Linux Reference Guide*.

3.1.1 A Note about Virtual Consoles

The Red Hat Linux installation program offers more than the dialog boxes of the installation process. Several different kinds of diagnostic messages are available to you, in addition to giving you a way to enter commands from a shell prompt. It presents this information on five **virtual consoles**, among which you can switch using a single keystroke.

These virtual consoles can be helpful if you encounter a problem while installing Red Hat Linux. Messages displayed on the installation or system consoles can help pinpoint a problem. Please see Table 3–1, *Console, Keystrokes, and Contents* for a listing of the virtual consoles, keystrokes to switch to them, and their contents.

Table 3–1 Console, Keystrokes, and Contents

Console	Keystrokes	Contents
1	[Ctrl]-[Alt]-[F1]	installation dialog
2	[Ctrl]-[Alt]-[F2]	shell prompt
3	[Ctrl]-[Alt]-[F3]	install log (messages from installation program)
4	[Ctrl]-[Alt]-[F4]	system-related messages
5	[Ctrl]-[Alt]-[F5]	other messages
7	[Ctrl]-[Alt]-[F7]	X graphical display

Generally, there's no reason to leave the default console (virtual console #7) unless you are attempting to diagnose installation problems. But if you get curious, feel free to look around.

3.2 Starting the Installation Program

Now it's time to begin installing Red Hat Linux. To start the installation, you must first boot the installation program. Please make sure you have all the resources you'll need for the installation. If you've already read through Chapter 2, *Before You Begin*, and followed the instructions, you should be ready to begin.

3.2.1 Booting the Installation Program

Please Note

If you need to create a boot disk, please refer to Section 2.1.6, *Step 6 - How Do You Want to Start the Installation?*.

Insert the boot disk into your computer's first diskette drive and reboot (or boot using the CD-ROM, if your computer supports this). Your BIOS settings may need to be changed to allow you to boot from the diskette or CD-ROM.

Tip

To change your BIOS settings, you will need to take note of the instructions given when your computer first begins to boot. Often you will see a line of text telling you to press the [Del] key to enter the BIOS settings. Once you have done whatever process is needed to enter your computer's BIOS, you can then change the boot order to allow your computer to boot from the CD-ROM drive or diskette drive first when bootable software is detected. For more information, please refer to the documentation that came with your system.

There are four possible boot methods:

- *Bootable CD-ROM* — your machine supports a bootable CD-ROM drive and you want to perform a local CD-ROM installation.
- *Local boot disk* — your machine will not support a bootable CD-ROM and you want to install from a local CD-ROM or a hard drive.
- *Network boot disk* — use to install from NFS, FTP and HTTP installation methods.

- *PCMCIA boot disk* — use in cases where you need PCMCIA support, but your machine does not support booting from the CD-ROM drive *or* if you need PCMCIA support in order to make use of the CD-ROM drive on your system. This boot disk offers you all installation methods (CD-ROM, hard drive, NFS, FTP, and HTTP).

After a short delay, a screen containing the `boot:` prompt should appear. The screen contains information on a variety of boot options. Each boot option also has one or more help screens associated with it. To access a help screen, press the appropriate function key as listed in the line at the bottom of the screen.

You should keep two things in mind:

- The initial screen will automatically start the installation program if you take no action within the first minute. To disable this feature, press one of the help screen function keys.
- If you press a help screen function key, there will be a slight delay while the help screen is read from diskette.

Normally, you'll only need to press [Enter] to boot. Watch the boot messages to see whether the Linux kernel detects your hardware. If it does not properly detect your hardware, you may need to restart the installation in "expert" mode. If your hardware is properly detected, please continue to the next section.

Expert mode can be entered using the following boot command:

```
boot: linux expert
```

Please Note

If you do not wish to perform a CD-ROM GUI installation, you can choose to perform a text mode installation by using the following boot command:

```
boot: text
```

For text mode installation instructions, please refer to the *Official Red Hat Linux Reference Guide*.

The command to start a **serial installation** has changed. If you need to 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).

Please Note

The initial boot messages will not contain any references to SCSI or network cards. This is normal, since these devices are supported by modules that are loaded during the installation process.

Options can also be passed to the kernel.

For example, to instruct the kernel to use all the RAM in a 128MB system, enter:

```
boot: linux mem=128M
```

After entering any options, press [Enter] to boot using those options.

If you do need to specify boot options to identify your hardware, please make note of them — they will be needed during the LILO configuration portion of the installation (please see Section 4.6, *Installing LILO* for more information).

Booting without diskettes

The Red Hat Linux/Intel CD-ROM can also be booted by computers that support bootable CD-ROMs. Not all computers support this feature, so if yours can't boot from the CD-ROM, there is one other way to start the installation without using a boot disk. The following method is specific to Intel-based computers only.

If you have MS-DOS installed on your system, you can boot directly from the CD-ROM drive without using a boot disk.

To do this (assuming your CD-ROM is drive `d:`), use the following commands:

```
C:\> d:
D:\> cd \dosutils
D:\dosutils> autoboot.bat
```

This method will not work if run in a DOS window — the `autoboot.bat` file must be executed with DOS as the only operating system. In other words, Windows cannot be running.

If your computer can't boot directly from CD-ROM (and you can't use a DOS-based autoboot), you'll have to use a boot diskette to get things started.

3.3 Selecting an Installation Method

Next, you will be asked what type of installation method you wish to use. You can install Red Hat Linux via the following basic methods:

CD-ROM

If you have a CD-ROM drive and the Red Hat Linux CD-ROM. Requires a boot disk, a bootable CD-ROM or a PCMCIA boot disk.

Hard Drive

If you copied the Red Hat Linux files to a local hard drive. Refer to the *Official Red Hat Linux Reference Guide* for hard drive installation instructions. Requires a boot disk or a PCMCIA boot disk.

NFS Image

If you are installing from an NFS Image server which is exporting the Red Hat Linux CD-ROM or a mirror image of Red Hat Linux. Requires a network or PCMCIA boot disk. Refer to the *Official Red Hat Linux Reference Guide* for network installation instructions. Please note: NFS installations may also be performed in GUI mode.

FTP

If you are installing directly from an FTP server. Requires a network or PCMCIA boot disk. Refer to the *Official Red Hat Linux Reference Guide* for FTP installation instructions.

HTTP

If you are installing directly from an HTTP Web server. Requires a network or PCMCIA boot disk. Refer to the *Official Red Hat Linux Reference Guide* for HTTP installation instructions.

3.4 Beginning the Installation

If you are planning to install via CD-ROM using the graphical interface, please read on.

Please Note

If you'd rather perform a text mode installation, reboot your system and at the `boot:` prompt, type **`text`**. Refer to the *Official Red Hat Linux Reference Guide* for further instructions.

3.4.1 Installing from CD-ROM

To install Red Hat Linux from CD-ROM, choose "CD-ROM" and select **OK**. When prompted, insert the Red Hat Linux CD into your CD-ROM drive (if you did not boot from the CD-ROM). Once done, select **OK**, and press [Enter].

The installation program will then probe your system and attempt to identify your CD-ROM drive. It will start by looking for an IDE (also known as ATAPI) CD-ROM drive. If found, you will continue to the next stage of the installation process (see Section 3.5, *Language Selection*).

If a drive is not detected, you'll be asked what type of CD-ROM drive you have. Choose from the following types:

SCSI

Select this if your CD-ROM drive is attached to a supported SCSI adapter; the installation program will then ask you to choose a SCSI driver. Choose the driver that most closely resembles your adapter. You may specify options for the driver if necessary; however, most drivers will detect your SCSI adapter automatically.

Other

If your CD-ROM drive is neither an IDE nor a SCSI, it's an "other." Sound cards with proprietary CD-ROM interfaces are good examples of this CD-ROM type. The installation program presents a list of drivers for supported CD-ROM drives — choose a driver and, if necessary, specify any driver options.

Tip

A partial list of optional parameters for CD-ROM drives can be found in the *Official Red Hat Linux Reference Guide*, in the *General Parameters and Modules* appendix.

What If the IDE CD-ROM Was Not Found?

If the installation program fails to find your IDE (ATAPI) CD-ROM (it asks you what type of CD-ROM drive you have), restart the installation, and at the `boot:` prompt enter **`linux hd`*`X`*`=cdrom`**. Replace the ***X*** with one of the following letters, depending on the interface the unit is connected to, and whether it is configured as master or slave:

- `a` - First IDE controller, master
- `b` - First IDE controller, slave
- `c` - Second IDE controller, master
- `d` - Second IDE controller, slave

(If you have a third and/or fourth controller, simply continue assigning letters in alphabetical order, going from controller to controller, and master to slave.)

Once identified, you will be asked to insert the Red Hat Linux CD into your CD-ROM drive. Select **OK** when you have done so. After a short delay, the next dialog box will appear.

After booting, the installation program begins by displaying the language screen.

Please Note

If you wish to abort the installation process at this time, simply reboot your machine then eject the boot diskette or CD-ROM. You can safely cancel the installation at any point before the **About to Install** screen, see Section 4.13, *Preparing to Install*.

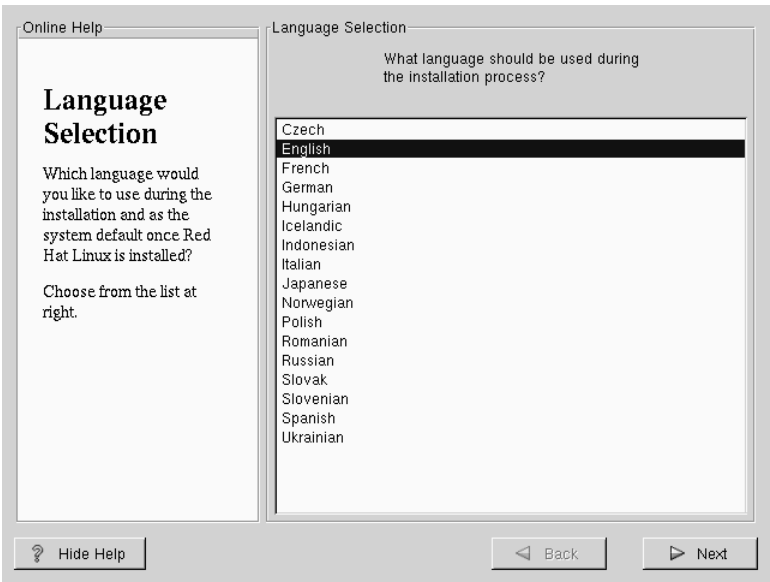
3.5 Language Selection

Please Note

Red Hat High Availability Server 1.0 supports only English during the installation process.

Using your mouse, select **English** as the language you would prefer to use for the installation and as the system default (see Figure 3–1, *Language Selection*).

Figure 3–1 Language Selection



3.6 Keyboard Configuration

Choose the model that best fits your system (see Figure 3–2, *Keyboard Configuration*). If you cannot find an exact match, choose the best **Generic** match for your keyboard type (for example, **Generic 101-key PC**).

Next, choose the correct layout type for your keyboard (for example, U.S. English).

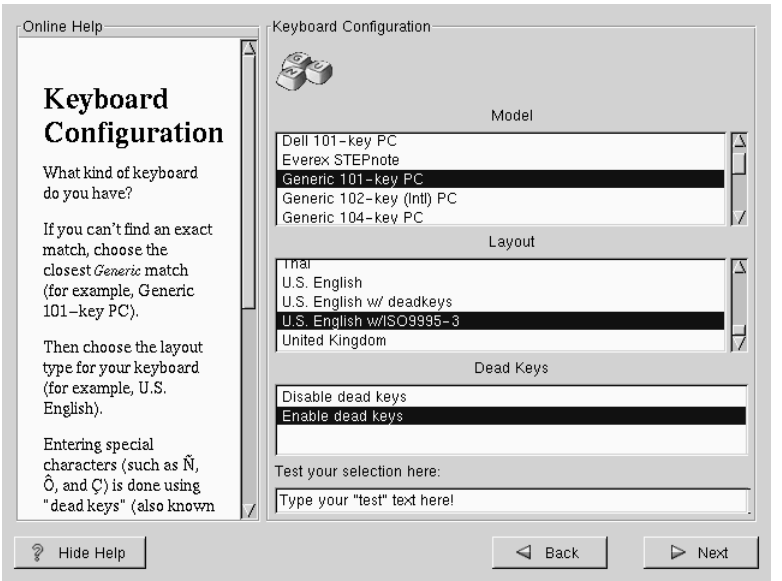
Creating special characters with multiple keystrokes (such as Ñ, Ô, and Ç) is done using "dead keys" (also known as compose key sequences). Dead keys are enabled by default. If you do not wish to use them, select **Disable dead keys**.

To test your configuration, use the blank text field at the bottom of the screen to enter text.

Tip

To change your keyboard type post-installation, become **root** and use the `/usr/sbin/kbdconfig` command, or you can type `setup` at the `root` prompt.

Figure 3–2 Keyboard Configuration



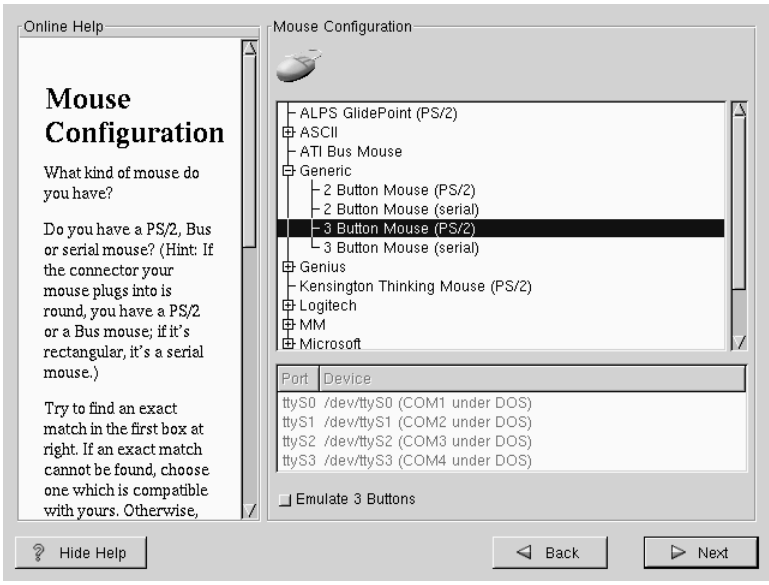
3.7 Mouse Configuration

Choose the correct mouse type for your system. If an exact match cannot be found, choose a mouse type that you are sure is compatible with your system (see Figure 3–3, *Mouse Configuration*).

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; 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 a mouse that you are sure is compatible with your system, select one of the **Generic** entries, based on your mouse's number of buttons, and its interface.

Figure 3–3 Mouse Configuration



If you have a PS/2 or a Bus mouse, you do not need to pick a port and device. If you have a serial mouse, you should choose the correct port and device that your serial mouse is on.

The **Emulate 3 Buttons** check box allows you to use a two-button mouse as if it had three buttons. In general, it's easiest to use the X Window System if you have a three-button mouse. If you select this check box, you can emulate a third, "middle" button by pressing both mouse buttons simultaneously.

Tip

To change your mouse configuration post-installation, become root. You can then use the `/usr/sbin/mouseconfig` command from the shell prompt.

To configure your mouse as a left-handed mouse, you can reset the order of the mouse buttons. This can be done after you have booted your Red Hat Linux system, by typing `gpm -B 321` at the shell prompt.

3.8 Welcome to Red Hat High Availability Server

The "Welcome" screen (see Figure 3–4, *Welcome to Red Hat High Availability Server*) does not prompt you for any installation input. Please read over the help text in the left panel for additional instructions and information on where to register your Official Red Hat Linux product.

Figure 3–4 Welcome to Red Hat High Availability Server

Please notice the **Hide Help** button at the bottom left corner of the screen. The help screen is open by default, but if you do not want to view the help information, click on the **Hide Help** to minimize the screen.

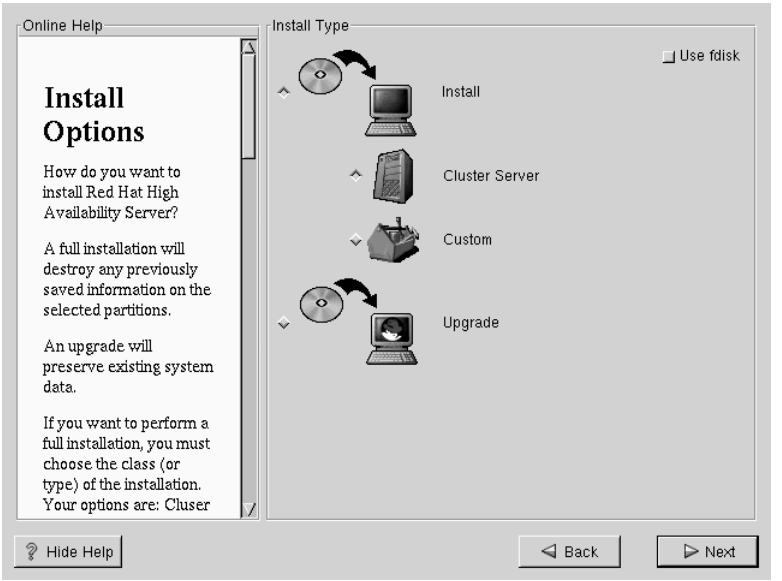
Click on the **Next** button to continue.

3.9 Install Options

Choose whether you would like to perform a full installation or an upgrade (see Figure 3–5, *Choosing Install or Upgrade*). Again, please keep in mind that you must perform a full installation in order to use the high availability features of Red Hat High Availability Server.

In the top right-hand corner of the **Install Type** screen there is a box you may select if you wish to partition using fdisk. Note that fdisk is not as intuitive to use as Disk Druid and is not selected by default. If you have not used fdisk before, you should read about both fdisk and Disk Druid to determine which will best suit your needs.

Figure 3–5 Choosing Install or Upgrade



To perform a full GUI installation, please refer to Chapter 4, *Installing Red Hat Linux 6.2* for those instructions.

To perform an upgrade, please refer to Chapter 5, *Upgrading Your Current System*.

4 Installing Red Hat Linux 6.2

Once you have finished this chapter, you will have completed a full installation of Red Hat Linux 6.2.

If you need information about performing an upgrade, please refer to Chapter 5, *Upgrading Your Current System* for those instructions.

4.1 Continuing the Installation

You usually install Red Hat Linux on a clean disk partition or set of partitions, or over another installation of Linux.

WARNING

Installing Red Hat Linux over another installation of Linux (including Red Hat Linux) does *not* preserve any information (files or data) from a prior installation. Make sure you save any important files! If you are worried about saving the current data on your existing system (without making a backup on your own), you should consider performing an upgrade instead (see Chapter 5, *Upgrading Your Current System*).

In choosing a full installation, you must also choose the class of the installation. Your options include: **Cluster Server** or **Custom.**

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.linuxdoc.org/HOWTO/mini/Linux+NT-Loader.html for more information on setting up LILO and NT.

WARNING

A cluster server-class installation will erase *all partitions* (both Linux and non-Linux) from *every one* of your computer's hard drive(s).

The *custom-class installation* allows you the most flexibility during your installation. The workstation-class and server-class installations automatically go through the installation process for you and omit certain steps. During a custom-class installation, it is up to *you* how disk space should be partitioned. You have complete control over the packages that will be installed on your system. You can also determine whether

you'll use LILO (the LInux LOader) to boot your system. Unless you have prior Linux experience, you should not select the custom-class installation method.

If you would like to know what steps are omitted by not performing a custom-class installation please refer to *Behind the Scenes of a Custom-Class Installation* in Section 2.1.7.

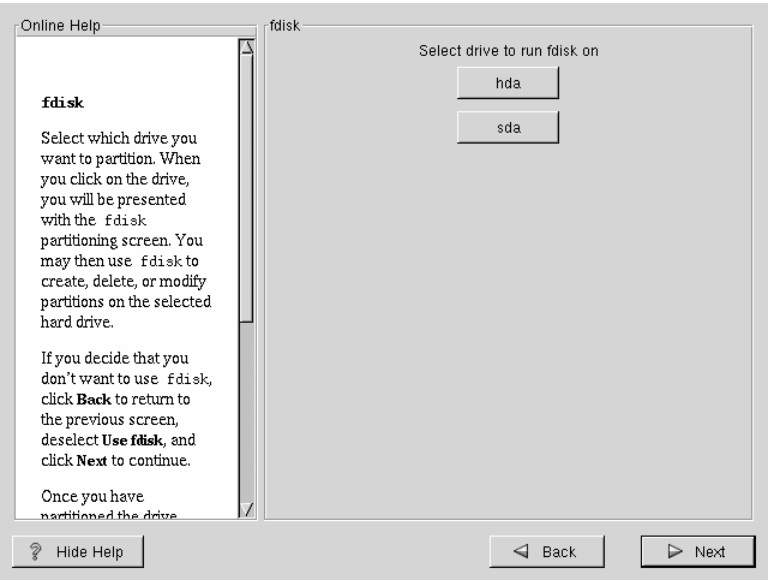
4.2 Partitioning with fdisk

Unless you have previously used fdisk and understand how it works, we do not recommend that you use it. Disk Druid is an easier and friendlier partitioning tool for those new to partitioning their system. To exit fdisk click **Back** to return to the previous screen, deselect fdisk, and then click **Next**.

This section applies only if you chose to use fdisk to partition your system. If are not using fdisk, please skip to Section 4.3, *Automatic Partitioning* for automatic partitioning or Section 4.4, *Partitioning Your System* for partitioning with Disk Druid.

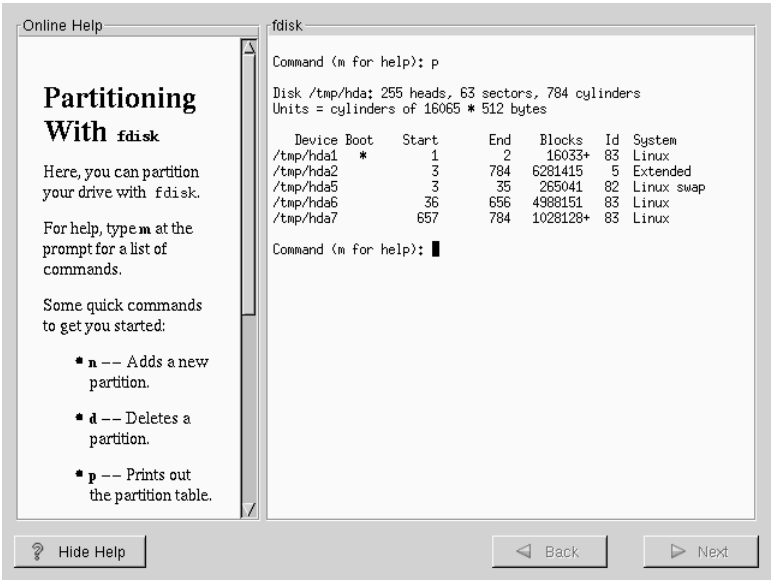
If you have chosen to use fdisk, the next screen (see Figure 4–1, *fdisk*) will prompt you to select a drive to partition using fdisk.

Figure 4–1 fdisk



Once you have chosen which drive to partition, you will be presented with the fdisk command screen (see Figure 4–2, *Partitioning with fdisk*). If you are unsure as to what command you should use, type [m] at the prompt for help. Please refer to the *Official Red Hat Linux Reference Guide* for an overview of fdisk. When you've finished making partitions, type w to save your changes and quit. You will be taken back to the original fdisk screen where you can choose to partition another drive or continue with your installation.

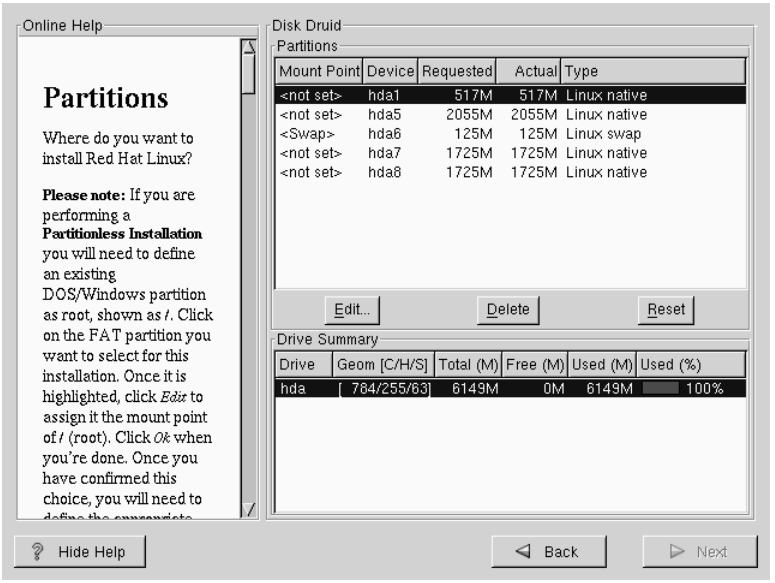
Figure 4–2 Partitioning with fdisk



After you have partitioned your drive(s), click **Next**. You will then use Disk Druid to assign **mount points** to your partitions.

You will not be able to add new partitions using Disk Druid, but you will be able to edit those you have already created.

Figure 4–3 Editing with Disk Druid



Skip to Section 4.5, *Choose Partitions to Format* for further installation instructions.

4.3 Automatic Partitioning

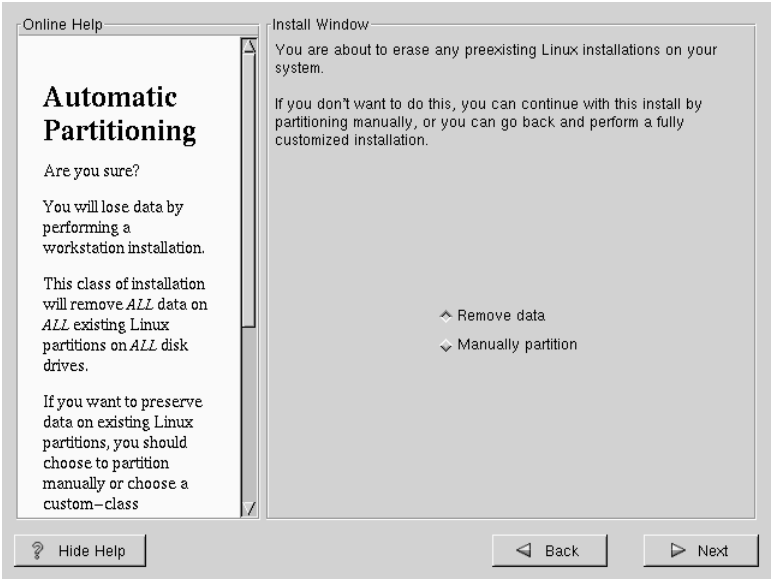
`Automatic Partitioning` allows you to perform an installation without having to partition your drive(s) yourself. If you do not feel comfortable with partitioning your system, it is recommended that *do not* choose to partition manually and instead let the installation program partition for you.

The `Automatic Partitioning` screen is only seen when performing a workstation- or server-class installation. If you are performing a custom-class installation, or choose to manually partition, please refer to Section 4.4, *Partitioning Your System*.

In this screen, you can choose to continue with this installation, to partition manually, or use the **Back** button to choose a different installation method (see Figure 4–4, *Automatic Partitioning*).

If you do *not* want to lose some or all of your data, you should either choose to partition manually or choose a different installation class.

Figure 4–4 Automatic Partitioning



CAUTION

A cluster server-class installation will remove all data on all partitions of all hard drives.

If you have another OS on your system that you wish to keep installed, if you do not want Red Hat Linux to be installed on your master boot record (MBR), or if you want to use a boot manager other than LILO, do not choose this installation method.

If you are unsure how you want your system to be partitioned, please read the chapter on partitioning in the *Official Red Hat Linux Reference Guide*.

4.4 Partitioning Your System

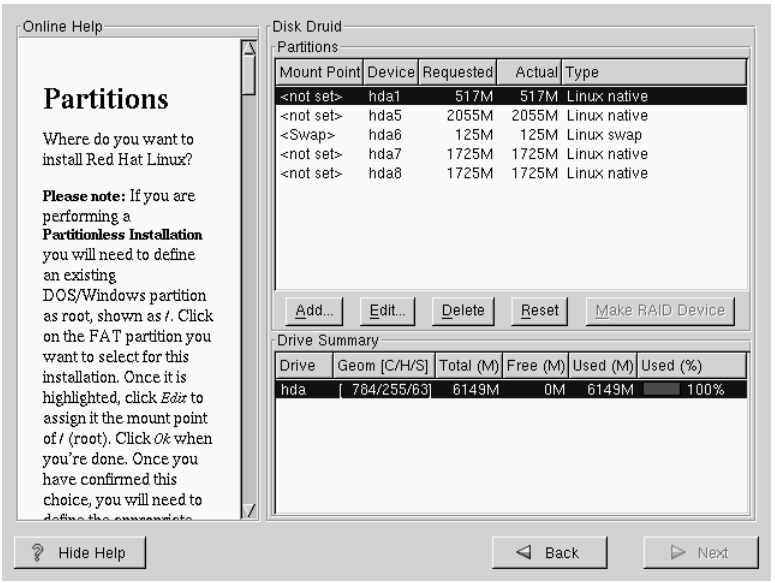
If you are performing a workstation- or server-class installation and you chose *not* to partition manually, please skip to Section 4.7, *Network Configuration*.

At this point, it's necessary to let the installation program know where it should install Red Hat Linux. This is done by defining mount points for one or more disk partitions in which Red Hat Linux will be installed. You may also need to create and/or delete partitions at this time (refer to Figure 4–5, *Partitioning with Disk Druid*).

Please Note

If you have not yet planned how you will set up your partitions, refer to the partitioning appendix in the *Official Red Hat Linux Reference Guide*. As a bare minimum, you'll need an appropriately-sized root partition, and a swap partition of at least 16 MB.

Figure 4–5 Partitioning with Disk Druid



The partitioning tool used in Red Hat Linux 6.2 is Disk Druid. With the exception of certain esoteric situations, Disk Druid can handle the partitioning requirements for a typical Red Hat Linux installation.

4.4.1 Partition Fields

Each line in the "Partitions" section represents a disk partition. Each line in this section has five different fields:

Mount Point:

A mount point is the location within the directory hierarchy at which a volume exists. The volume is said to be mounted at this location. This field indicates where the partition will be mounted. If a partition exists, but is "not set" you need to define its mount point. Double-click on the partition or use the **Edit** key.

Unless you have a reason for doing otherwise, we recommend that you create the following partitions:

- A swap partition (at least 16MB) — Swap partitions are used to support virtual memory. In other words, data is written to a swap partition when there is not enough RAM to store the data your system is processing. If your computer has 16MB 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 16MB (whichever is larger).
- A `/boot` partition (16MB, maximum) — The partition mounted on `/boot` contains the operating system kernel (which allows your system to boot Red Hat Linux), along with 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.
- A `root` partition (700MB-1.7GB) — This is where "`/`" (the root directory) resides. In this setup, all files (except those stored in `/boot`) reside on the root partition. A 700MB root partition will permit the equivalent of a workstation-class installation (with *very* little free space), while a 1.7GB root partition will let you install every package.

Device:

This field displays the partition's device name.

Requested:

This field shows the partition's original size. To re-define the size, you must delete the current partition and recreate it using the **Add** button.

Actual:

This field shows the space currently allocated to the partition.

Type:

This field shows the partition's type (such as Linux Native or DOS).

4.4.2 Problems When Adding a Partition

If you attempt to add a partition and Disk Druid can't carry out your request, you'll see a dialog box listing partitions that are currently unallocated, along with the reason they could not be allocated. Unallocated partition(s) are also displayed on Disk Druid's main screen (though you may have to scroll through the "Partitions" section to see them).

As you scroll through the **Partitions** section, you might see an "Unallocated Requested Partition" message (in red text), followed by one or more partitions. A common reason for this 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 requested mount point.

To fix an unallocated requested partition, you must move the partition to another drive which has the available space, resize the partition to fit on the current drive, or delete the partition entirely. Make changes using the **Edit** button or by double clicking on the partition.

4.4.3 Drive Summaries

Each line in the **Drive Summaries** section represents a hard disk on your system. Each line has the following fields:

Drive:

This field shows the hard disk's device name.

Geom [C/H/S]:

This field 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:

This field shows the total available space on the hard disk.

Free:

This field shows how much of the hard disk's space is still unallocated.

Used:

These fields show how much of the hard disk's space is currently allocated to partitions, in megabytes and percentage.

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. That is done using the **Allowable Drives** field in Section 4.4.5, *Adding Partitions*.

4.4.4 Disk Druid's Buttons

These buttons control Disk Druid's actions. They are used to add and delete partitions, and to change partition attributes. There are also buttons that are used to accept the changes you've made, or to exit Disk Druid. Let's take a look at each button in order.

Add:

used to request a new partition. When selected, a dialog box will appear containing fields (such as mount point and size) that must be filled in.

Edit:

used to modify attributes of the partition currently selected in the "Partitions" section. Selecting **Edit** will open up a dialog box. Some or all of the fields can be edited, depending on whether the partition information has already been written to disk.

Delete:

used to remove the partition currently highlighted in the **Current Disk Partitions** section. You'll be asked to confirm the deletion of any partition.

Reset:

used to restore Disk Druid to its original state. All changes made will be lost if you **Reset** the partitions.

Make RAID Device:

Make RAID Device can be used if you want to provide redundancy to any or all disk partitions. *It should only be used if you have experience using RAID.* To read more about RAID, please refer to the *Official Red Hat Linux Reference Guide*.

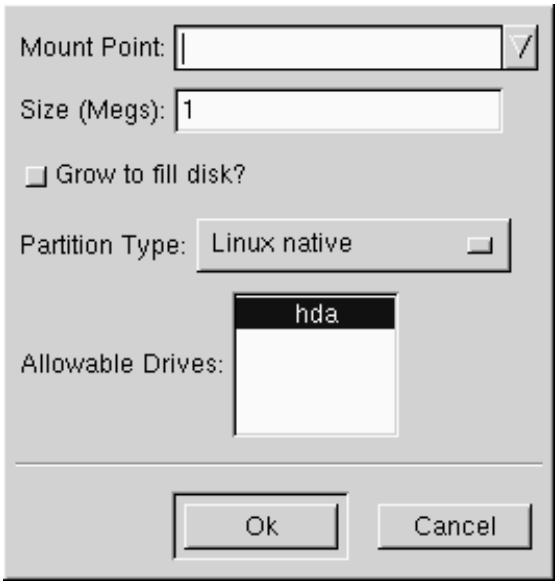
4.4.5 Adding Partitions

To add a new partition, select the **Add** button. A dialog box will appear (see Figure 4–6, *Adding a Partition*).

Please Note

You will need to dedicate at least one partition to Red Hat Linux, and optionally more. This is discussed more completely in Appendix C in the *Official Red Hat Linux Reference Guide*.

Figure 4–6 Adding a Partition

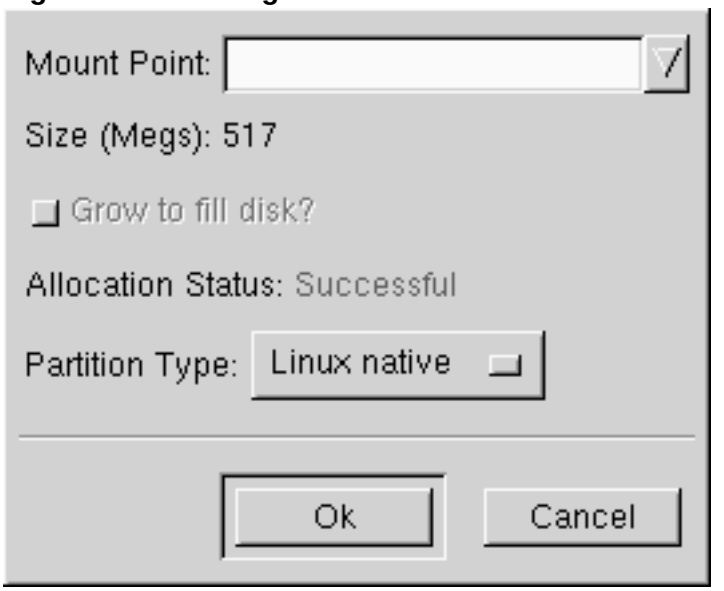


- **Mount Point**: Highlight and enter the partition's mount point. For example, if this partition should be the root partition, enter `/`; enter `/boot` for the `/boot` partition, and so on. You can also use the pull-down menu to choose the correct mount point for your partition.
- **Size (Megs)**: Enter the size (in megabytes) of the partition. Note this field starts with a "1" in it; unless changed you'll end up with a 1 MB partition.
- **Grow to fill disk**: This check box indicates if the size you entered in the previous field is to be considered the partition's exact size, or its minimum size. When selected, the partition will grow to fill all available space on the hard disk. The partition's size will expand and contract as other partitions are modified. You can make multiple partitions growable; if you do, the additional free space will be shared among all growable partitions.
- **Partition Type**: This field contains a list of different partition types (such as Linux Native or DOS). Select the appropriate partition type by using the mouse.
- **Allowable Drives**: This field contains a list of the hard disks installed on your system. If a hard disk's box is highlighted, then a desired partition can be created on that hard disk. If the box is *not* checked, then the partition will *never* 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 **Ok** once you're satisfied with the settings, and wish to create the partition.
- **Cancel**: Select **Cancel** if you don't want to create the partition.

4.4.6 Editing Partitions

To edit a partition, select the **Edit** button or double-click on the existing partition (see Figure 4–7, *Editing a Partition*).

Figure 4–7 Editing a Partition



Please Note

If the partition already existed on your hard disk, you will only be able to change the partition's mount point. If you want to make any other changes, you will need to delete the partition and recreate it.

4.4.7 Deleting a Partition

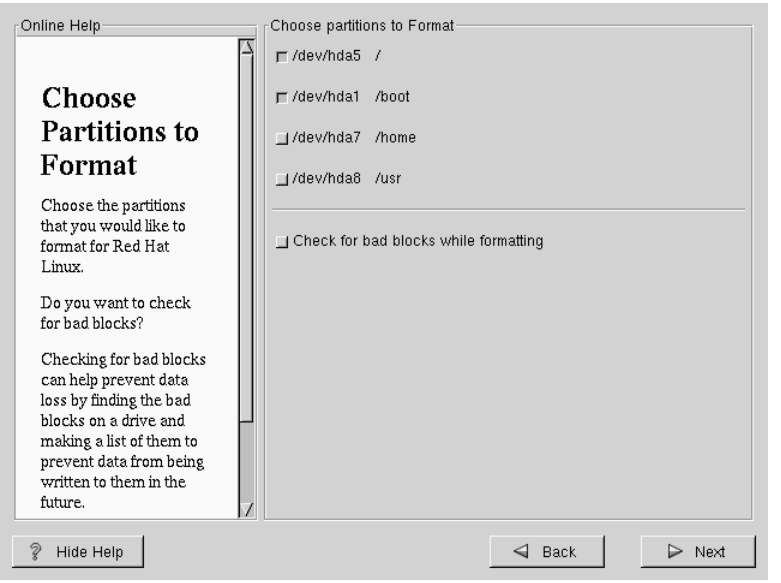
To delete a partition, highlight it in the "Partitions" section and double-click the **Delete** button. You will be asked to confirm the deletion.

4.5 Choose Partitions to Format

Choose the partitions that you would like to format. All newly created partitions should be formatted. In addition, any existing partitions that contain data you no

longer need should be formatted. However, partitions such as `/home` or `/usr/local` must not be formatted if they contain data you wish to keep (see Figure 4–8, *Choosing Partitions to Format*).

Figure 4–8 Choosing Partitions to Format



If you wish to check for bad blocks while formatting each filesystem, please make sure to select the **check for bad blocks** option.

Checking for bad blocks can help prevent data loss by locating the bad blocks on a drive and making a list of them to prevent using them in the future.

4.6 Installing LILO

If you're performing a workstation- or server-class installation, please skip ahead to Section 4.8, *Time Zone Configuration.*

In order to be able to boot your Red Hat Linux system, you usually need to install LILO (the LInux LOader). You may install LILO in one of two places:

The master boot record (MBR)

The recommended place to install LILO, unless the MBR already starts another operating system loader, such System Commander or OS/2's Boot Manager. The master boot record 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 root 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).

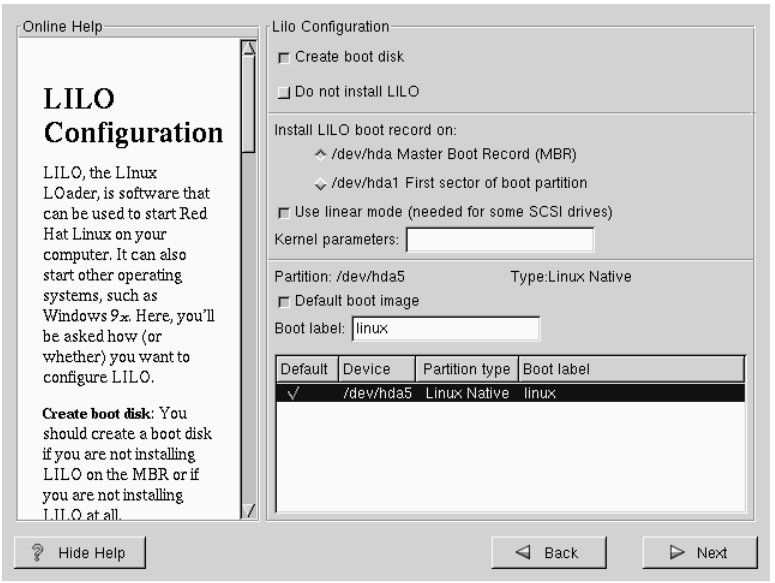
If you choose to install LILO, please select where you would like LILO to be installed on your system (see Figure 4–9, *LILO Configuration*). If your system will use only Red Hat Linux you should choose the master boot record (MBR). For systems with Win95/98, you also should install LILO to the MBR so that LILO can boot both operating systems.

If you have Windows NT (and you want to install LILO) you should choose to 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.

If you choose not to install LILO for any reason, 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 diskette). Use this option only if you are sure you have another way of booting your Red Hat Linux system!

The `Use linear mode` button is selected by default. In most cases, linear mode should be enabled; if your computer cannot use linear mode to access your hard drives, deselect this option.

Figure 4–9 LILO Configuration



If you wish to add default options to the LILO boot command, enter them into the kernel parameters field. Any options you enter will be passed to the Linux kernel every time it boots.

Bootable Partition — Every bootable partition 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. If you would like to add boot labels for other partitions (or change an existing boot label), click once on the partition to select it. Once selected, you can change the boot label.

Please Note

The "Boot label" column lists what you must enter at LILO's `boot:` prompt in order to boot the desired operating system. However, if you forget the boot labels defined on your system, you can always press [Tab] at LILO's `boot:` prompt to display a list of defined boot labels.

4.6.1 Configuring LILO

- **Create boot disk** — The **Create boot disk** option is checked by default. If you do not want to create a boot disk, you should deselect this option. However, we strongly urge you to create a boot disk. A boot disk can be handy for a number of reasons:
  - 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 If 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.
- **Do not install LILO** — if you have Windows NT installed on your system, you may not want to install LILO. If you choose not to install LILO for this reason, make sure that you have chosen to create a boot disk; otherwise you will not be able to boot Linux. You can also choose to skip LILO if you do not want to write LILO to your hard drive.

Tip

To use the boot disk with rescue mode, you have several options:

- Using the CD-ROM to boot, type `linux rescue` at the `boot:` prompt.
- Using the network boot disk, type `linux rescue` at the `boot:` prompt. You will then be prompted to pull the rescue image from the network.
- Using the boot disk included with the Red Hat Linux boxed set, type `linux rescue` at the `boot:` prompt. You then pick an installation method and choose a valid installation tree to load from.

For more information regarding rescue mode, refer to the *System Administration* chapter of the *Official Red Hat Linux Reference Guide*.

4.6.2 Alternatives to LILO

If you do not wish to use LILO to boot your Red Hat Linux system, there are several alternatives:

Boot Disk

As previously stated, you can use the boot disk created by the installation program (if you elected to create one).

LOADLIN

You 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

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.

Some commercial bootloaders

For example, System Commander and Partition Magic, which are able to boot Linux (but still require LILO to be installed in your Linux root partition).

4.6.3 SMP Motherboards and LILO

This section is specific to SMP motherboards only. If the installer detects an SMP motherboard on your system, it will automatically create two `lilo.conf` entries, rather than the usual single entry.

One entry will be called **linux** and the other will be called **linux-up**. The *linux* will boot by default. However, if you have trouble with the SMP kernel, you can elect to boot the *linux-up* entry instead. You will retain all the functionality as before, but you will only be operating with a single processor.

4.7 Network Configuration

If you have a network card and have not already configured your networking information, you now have the opportunity to configure networking (as shown in Figure 4–10, *Network Configuration*).

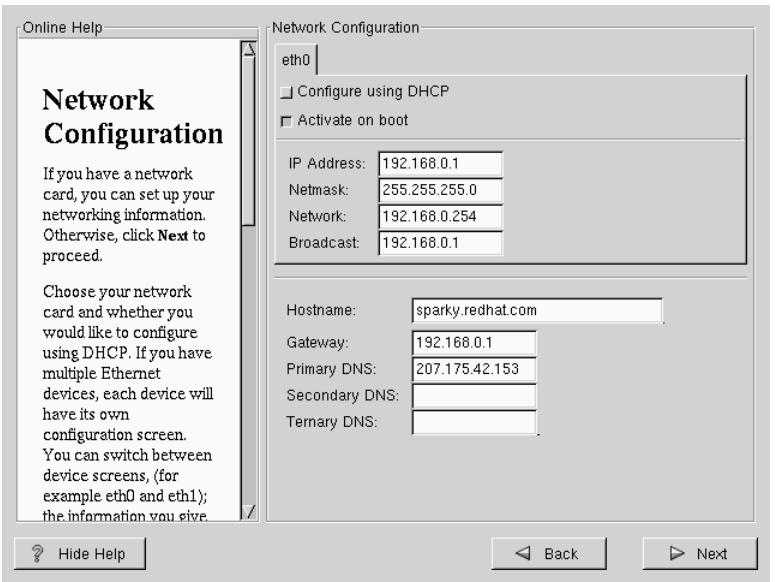
Please Note

If your system has more than one Ethernet card, please make sure you configure *both* cards. Select the card by clicking on the device name tab.

Choose your device type and whether you would like to configure using DHCP. If you have multiple Ethernet devices, each device will keep the information you have provided. You may switch between devices, for example `eth0` and `eth1`, and the information you give will be specific to each device. If you select **Activate on boot**, your network interface will be started when you boot. If you do not have DHCP client access or are unsure as to what this information is, please contact your network administrator.

Next enter, where applicable, the **IP Address**, **Netmask**, **Network**, and **Broadcast** addresses. If you are unsure about any of these, please contact your network administrator.

Figure 4–10 Network Configuration



Tip

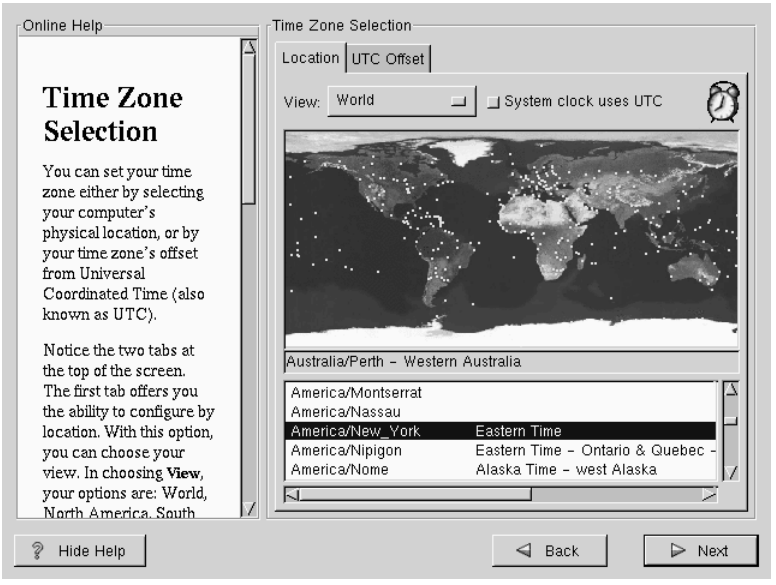
Even if your computer is not part of a network, you can enter a hostname for your system. Take this opportunity to enter in a name, if you do not, your system will be known as `localhost`.

Finally, enter the **Gateway** and **Primary DNS** (and if applicable the **Secondary DNS** and **Ternary DNS**) addresses.

4.8 Time Zone Configuration

You can set your time zone either by selecting your computer's physical location, or by your time zone's offset from Universal Coordinated Time (also known as UTC).

Figure 4–11 Configuring Time Zone



Notice the two tabs at the top of the screen (see Figure 4–11, *Configuring Time Zone*). The first tab offers you the ability to configure by location. With this option, you can choose your view. In choosing **view**, your options are: **World**, **North America**, **South America**, **Pacific Rim**, **Europe**, **Africa**, and **Asia**.

From the interactive map, you can also click on a specific city, as indicated by the yellow dots; a red **x** will appear indicating your selection. You can also scroll through a list and choose your desired time zone.

The second tab offers you the ability to use the UTC offset. UTC presents you with a list of offsets to choose from, as well as an option to set daylight saving time.

For both tabs, there is the option of selecting **System Clock uses UTC**. Please select this if you know that your system is set to UTC.

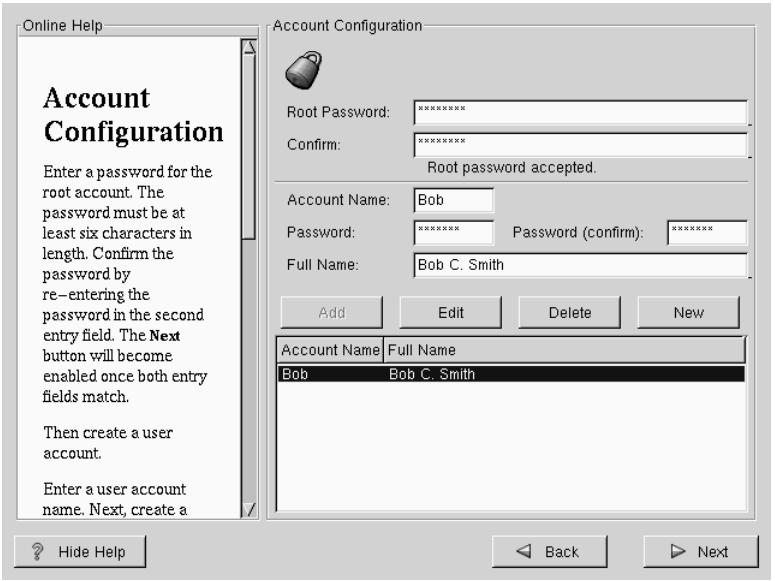
Tip

If you wish to change your time zone configuration after you have booted your Red Hat Linux system, become root and use the `/usr/sbin/timeconfig` command.

4.9 Account Configuration

The **Account Configuration** screen allows you to set your root password. Additionally, you can set up user accounts for you to log into once the installation is complete (see Figure 4–12, *Account Creation*).

Figure 4–12 Account Creation



4.9.1 Setting the Root Password

The installation program will prompt you to set a **root password** for your system.

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.

Please Note

The **root** user (also known as the **superuser**) has complete access to the entire system; for this reason, logging in as the root user is best done *only* to perform system maintenance or administration.

4.9.2 Setting Up User Accounts

If you choose to create a user account now, you will have an account to log in to once the installation has completed. This allows you to safely and easily log into your computer without having to be **root** to create other accounts.

Enter an account name. Then enter and confirm a password for that user account. Enter the full name of the account user and press [Enter]. Your account information will be added to the account list, clearing the user account fields so you can add another user.

You can also choose **New** to add a new user. Enter the user's information and use the **Add** button to add the user to the account list.

You can also **Edit** or **Delete** the user accounts you have created or no longer want.

4.10 Authentication Configuration

If you are performing a workstation-class installation, please skip ahead to Section 4.12, *GUI X Configuration Tool*.

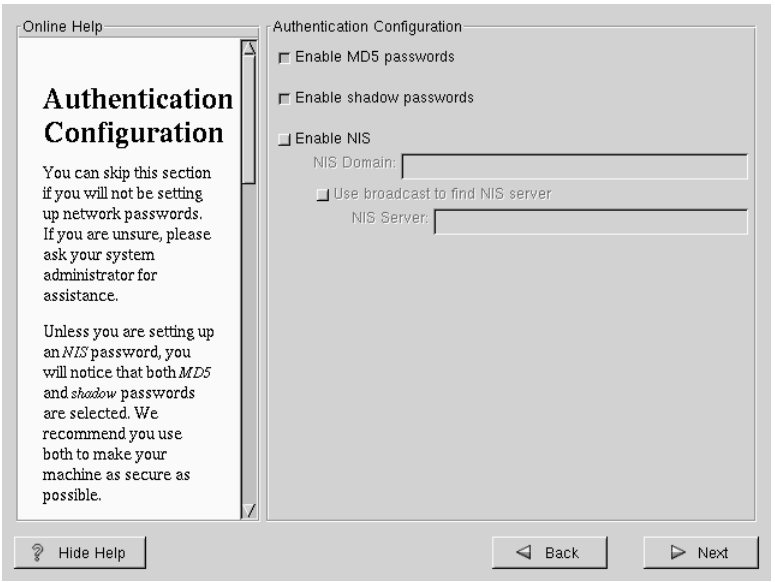
If you are performing a server-class installation, please skip ahead to Section 4.13, *Preparing to Install*.

You may skip this section if you will not be setting up network passwords. If you are unsure as to whether you should do this, please ask your system administrator for assistance.

Unless you are setting up **NIS** authentication, you will notice that both **MD5** and **shadow** passwords are selected (see Figure 4–13, *Authentication Configuration*). We recommend you use both to make your machine as secure as possible.

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 system administrator.

Figure 4–13 Authentication Configuration



- **MD5 Password** — allows a long password to be used (up to 256 characters), instead of the standard eight letters or less.
- **Shadow Password** — provides a secure method of retaining passwords. The passwords are stored in `/etc/shadow`, which is readable only by root.
- **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 to choose from here:
  - **NIS Domain** — this option allows you to specify which domain or group of computers your system belongs 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.

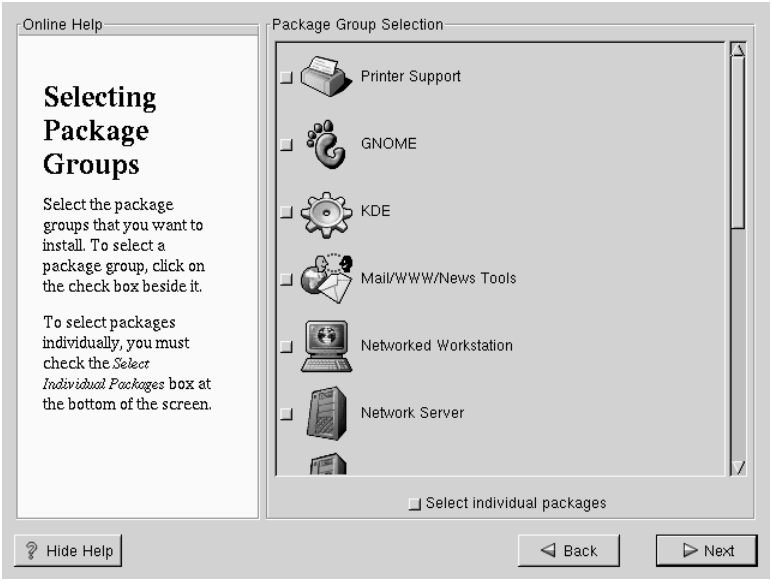
4.11 Package Group Selection

After your partitions have been selected and configured for formatting, you are ready to select packages for installation.

You can select **components**, which group packages together according to function (for example, `C Development`, `Networked Workstation`, or `Web Server`), **individual packages**, or a combination of the two.

To select a component, click on the check box beside it (see Figure 4–14, *Package Group Selection*).

Figure 4–14 Package Group Selection



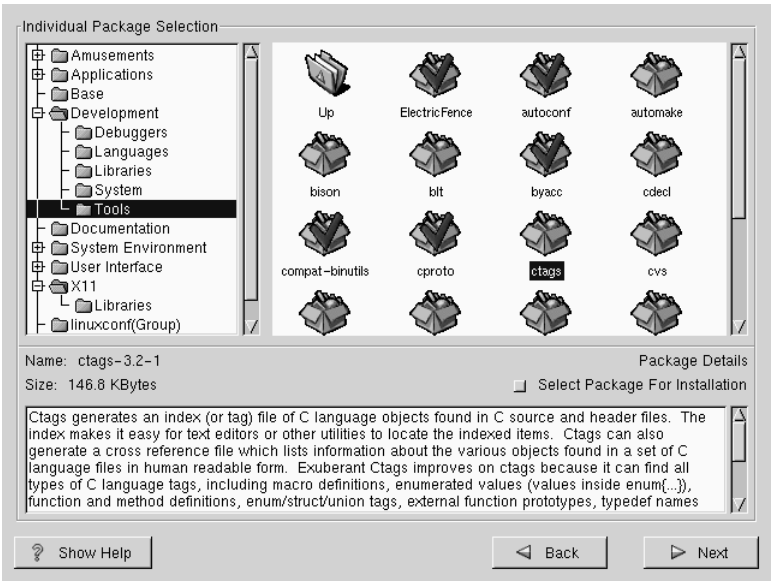
Select each component you wish to install. Selecting `Everything` (which can be found at the end of the component list) installs all packages included with Red Hat Linux. Selecting every package will require close to 1.7GB of free disk space.

To select packages individually, check the **Select Individual Packages** box at the bottom of the screen.

4.11.1 Selecting Individual Packages

After selecting the components you wish to install, you can select or deselect individual packages. The installation program presents a list of the packages in that group, which you can select or deselect using your mouse (see Figure 4–15, *Selecting Individual Packages*).

Figure 4–15 Selecting Individual Packages



On the left side of the screen you will see a directory listing of various package groups. When you expand this list (double-click to select it) and double-click on a single directory, the list of packages available for installation will appear on the right.

To select an individual package, double-click on it, or click on it once to highlight it and click on the **Select Package For Installation** button below. A red check mark will appear on any of the packages you have selected for installation.

To read information about a particular package before choosing it for installation, left-click on it once to highlight it, and the information will appear at the bottom of the screen along with the name and size of the package.

Please Note

Some packages (such as the kernel and certain libraries) are required for every Red Hat Linux system and are not available to select or deselect. These **base packages** are selected by default.

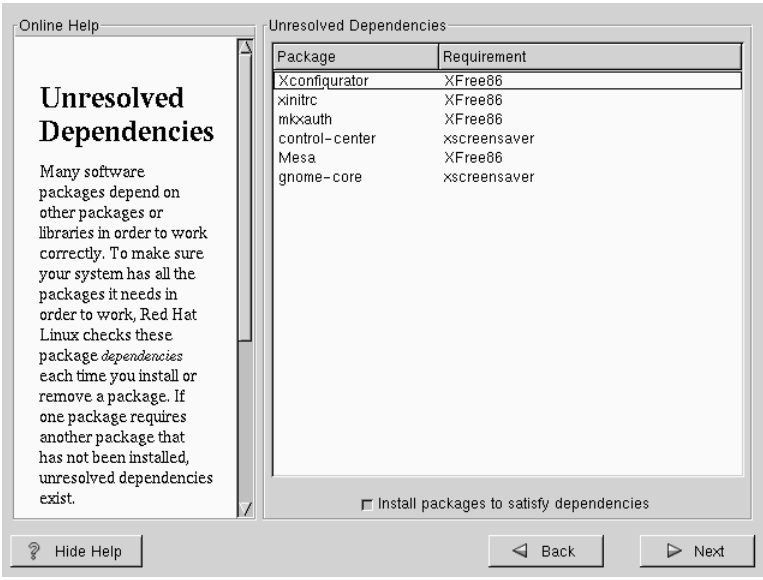
4.11.2 Unresolved Dependencies

Many software packages, in order to work correctly, depend on other software packages that must be installed on your system. For example, many of the graphical Red Hat 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.

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 4–16, *Unresolved Dependencies*).

The **Unresolved Dependencies** screen will only appear if you are missing certain packages that are needed by your selected packages. Under the list of missing packages, there is an **Install packages to satisfy dependencies** check box at the bottom of the screen which is selected by default. If you leave this checked, the installation program will resolve package dependencies automatically by adding all required packages to the list of selected packages.

Figure 4–16 Unresolved Dependencies



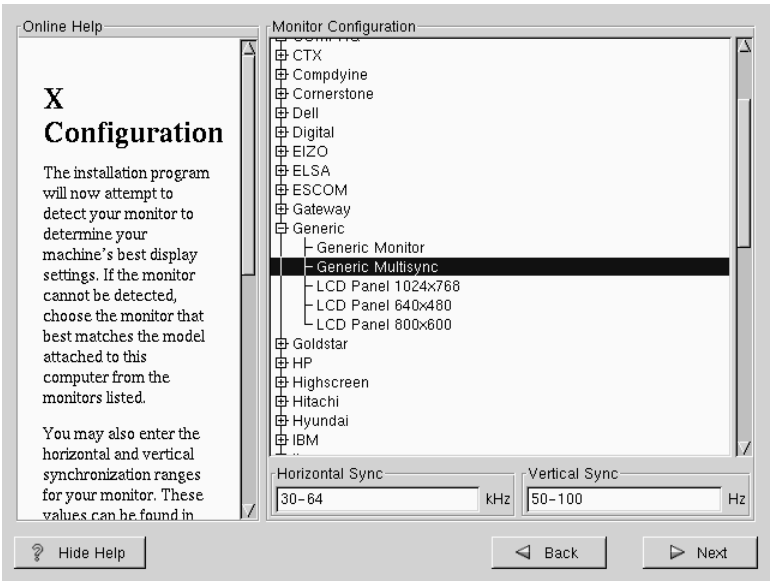
4.12 GUI X Configuration Tool

If you decided to install the X Window System packages, you now have the opportunity to configure an X server for your system. If you did not choose to install the X Window System packages, skip ahead to Section 4.14, *Installing Packages*.

4.12.1 Configuring Your Monitor

Xconfigurator, the X Window System configuration tool, first presents a list of monitors for you to choose from. In the list, you can either use the monitor that is autodetected for you, or choose another monitor.

Figure 4–17 Monitor Selection



If your monitor does not appear on the list, select the most appropriate **Generic** model available. If you do select a **Generic** monitor, Xconfigurator will suggest horizontal and vertical sync ranges. These values are generally available in the documentation which accompanies your monitor, or from your monitor's vendor or manufacturer; please check your documentation to make sure these values are set correctly.

CAUTION

Do not select a monitor *similar* to your monitor unless you are certain that the monitor you are selecting does not exceed the capabilities of your monitor. Doing so may overclock your monitor and damage or destroy it.

Also presented are the horizontal and vertical ranges that Xconfigurator suggests.

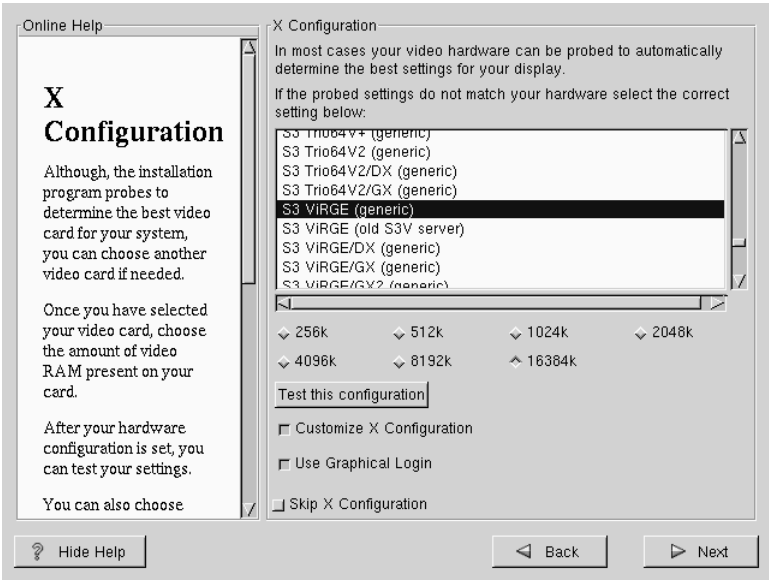
Click **Next** when you have finished configuration of your monitor.

4.12.2 Video Hardware Configuration

Next, Xconfigurator will probe for any video hardware you have (see Figure 4–18, *Videocard Setup*). Failing that, Xconfigurator will present a list of video cards and monitors for you to select from.

If your video card does not appear on the list, **XFree86** may not support it. However, if you have technical knowledge about your card, you may choose **Unlisted Card** and attempt to configure it by matching your card's video chipset with one of the available X servers.

Figure 4–18 Videocard Setup

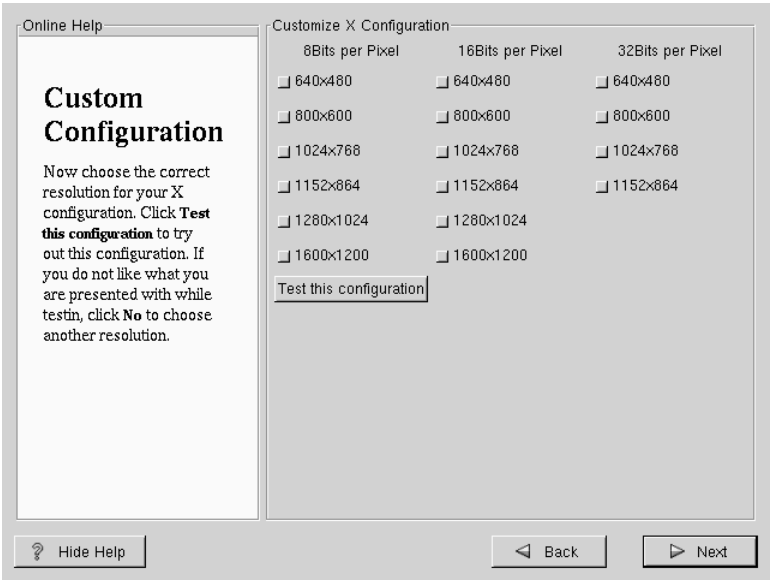


Next, Xconfigurator prompts you for the amount of video memory installed on your video card. If you are not sure, please consult the documentation accompanying your video card. You will not damage your video card by choosing more memory than is available, but the XFree86 server may not start correctly if you do.

Once your hardware has been determined, you can test the configuration settings. We recommend that you do test your configuration to make sure that the resolution and color is what you want to work with.

If you would like to customize the X configuration, please make sure the **Customize X Configuration** button is selected. If you choose to customize, you will be presented with another screen that lets you select what your resolution should be (see Figure 4–19, *X Customization*). Again, you will have the option of testing the configuration.

Figure 4–19 X Customization



You may also choose to **Skip X Configuration** if you would rather configure X after the install or not at all.

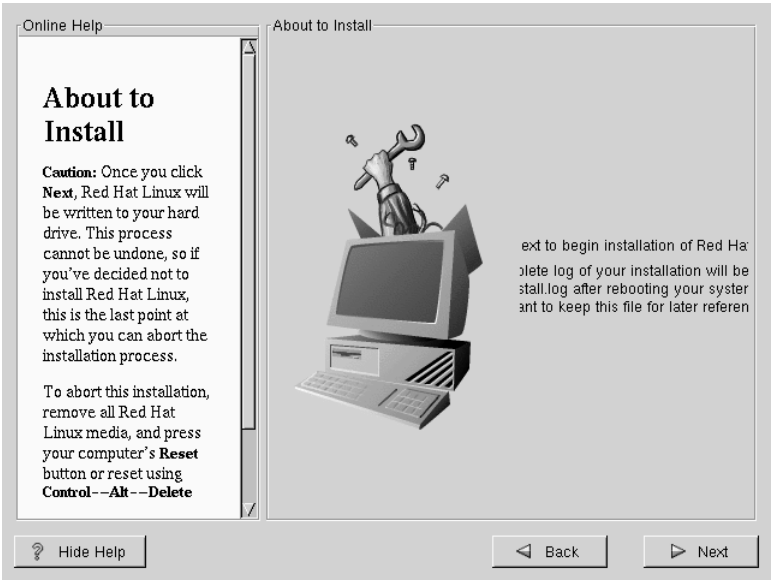
4.13 Preparing to Install

You will now see a screen preparing you for the installation of Red Hat Linux (see Figure 4–20, *Ready to Install*).

WARNING

If, for some reason, you would rather not continue with the installation process, this is your last opportunity to safely cancel the process and reboot your machine. Once you press the Next button, partitions will be written and packages will be installed. If you wish to abort the installation, you should reboot now before your hard drive(s) are rewritten.

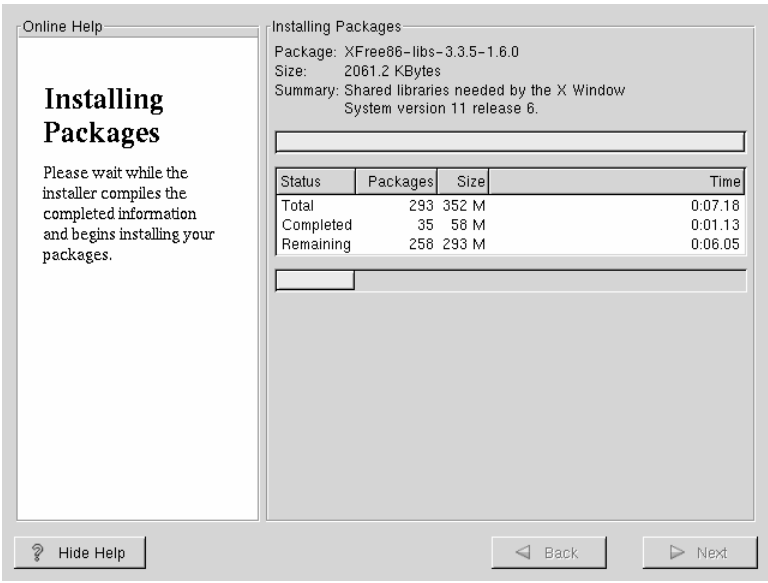
Figure 4–20 Ready to Install



4.14 Installing Packages

At this point there's nothing left for you to do until all the packages have been installed (see Figure 4–21, *Installing Packages*). How quickly this happens depends on the number of packages you've selected, and your computer's speed.

Figure 4–21 Installing Packages



4.15 Boot Disk Creation

If you chose to to create a boot disk, you should now insert a blank, formatted diskette into your floppy drive (see Figure 4–22, *Creating Your Boot Disk*).

After a short delay, your boot disk will be created; remove it from your floppy drive and label it clearly. Note that if you would like to create a boot disk after the installation, you'll be able to do so. For more information, please see the `mkbootdisk` man page, by typing `man mkbootdisk` at the shell prompt.

If you boot your system with the boot disk (instead of LILO), make sure you create a new boot disk if you make any changes to your kernel.

Figure 4–22 Creating Your Boot Disk

4.16 Installation Complete

Congratulations! Your Red Hat Linux 6.2 installation is now complete!

The installation program will prompt you to prepare your system for reboot (see Figure 4–23, *Installation Complete*). Don't forget to remove any diskette in the floppy drive or CD in the CD-ROM drive. If you did not install LILO, you'll need to use your boot disk now.

After your computer's normal power-up sequence has completed, you should see LILO's standard prompt, which is `boot:`. At the `boot:` prompt, you can do any of the following things:

- Press [Enter] — Causes LILO's default boot entry to be booted.
- Enter a Boot Label, followed by [Enter] — Causes LILO to boot the operating system corresponding to the boot label. (Press [?] at the `boot:` for a list of valid boot labels.)

- Do Nothing — After LILO's timeout period, (which, by default, is five seconds) LILO will automatically boot the default boot entry.

Figure 4–23 Installation Complete

Do whatever is appropriate to boot Red Hat Linux. You should see one or more screens of messages scroll by. Eventually, you should see a `login:` prompt or a GUI login screen (if you installed the X Window System and chose to start X automatically).

Tip

If you're not sure what to do next, we suggest you begin with the *Official Red Hat Linux Getting Started Guide* as an introduction to using Red Hat Linux. The *Official Red Hat Linux Getting Started Guide* covers topics relating to the basics of your system.

If you are a more experienced user looking for information on system configuration or administration topics, you may find the *Official Red Hat Linux Reference Guide* to be more helpful.

5 Upgrading Your Current System

This chapter explains those steps you'll see while performing an upgrade of Red Hat Linux 6.2.

5.1 What it Means to Upgrade

The installation process for Red Hat Linux 6.2 includes the ability to upgrade from prior versions of Red Hat Linux (version 2.0 and later) which are based on RPM technology.

Please Note

Performing an upgrade will *NOT* turn an existing Red Hat Linux system into a high availability server! You *must* perform a fresh Red Hat Linux installation in order to properly install Red Hat High Availability Server!

Upgrading your system installs the modular 2.2.x kernel as well as updated versions of the packages which are currently installed on your machine. The upgrade process preserves existing configuration files by renaming them using an `.rpmsave` extension (e.g., `sendmail.cf.rpmsave`) and leaves a log of the 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.

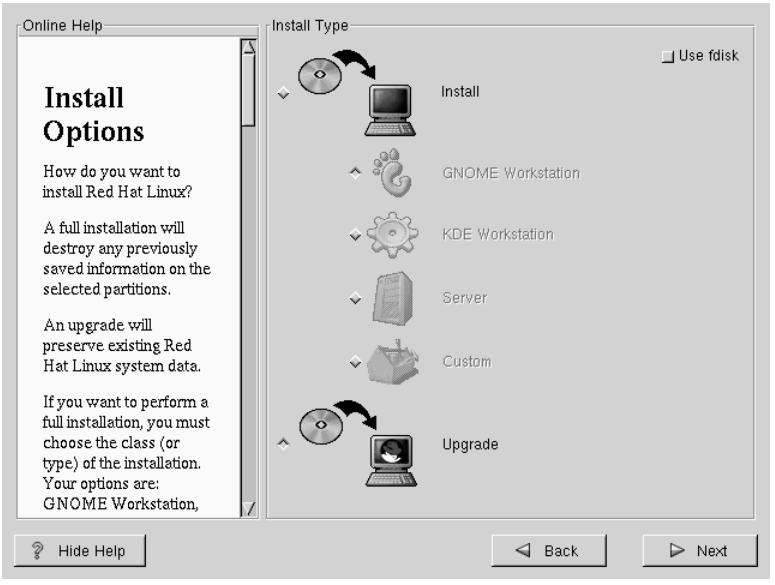
Please Note

Some upgraded packages may require that other packages are also installed for proper operation. If you choose to customize your packages to upgrade, you may be required to resolve any *dependency* problems. Otherwise, the upgrade procedure takes care of these dependencies, but it may need to install additional packages which are not on your existing system.

5.2 Upgrading Your System

At this point, you should have chosen **Upgrade** as your preferred installation type (see Figure 5–1, *Choosing to Upgrade*).

Figure 5–1 Choosing to Upgrade



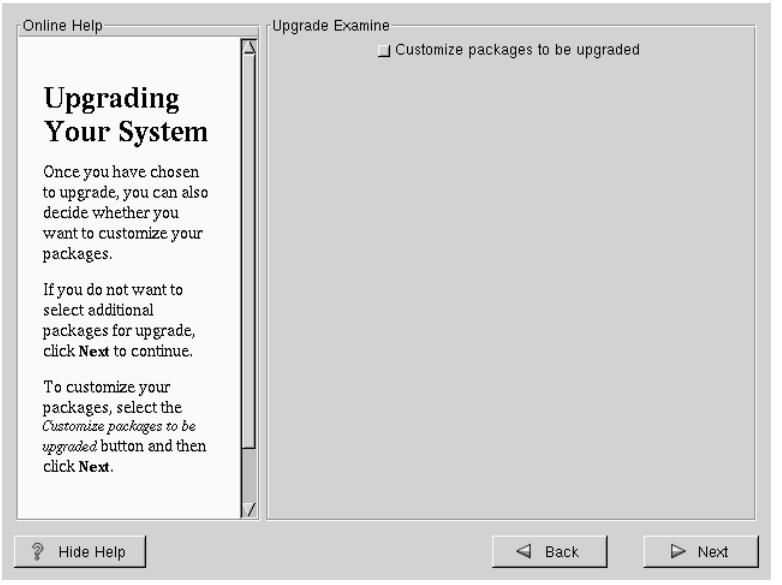
5.3 Customizing Your Upgrade

Next, you must choose whether to let the installation program upgrade your system for you or if you would like to customize your packages to be upgraded (see Figure 5–2, *Upgrade Customization*).

If you click **Next** and the **Customize packages to upgrade** button is not selected, your system will automatically begin the upgrade process (see Section 5.5, *Upgrading Packages*).

If you want to customize your upgrade packages, select this option and then click **Next.**

Figure 5–2 Upgrade Customization



5.4 Selecting Packages to Upgrade

Here, you are given the opportunity to choose which packages you would like to upgrade (see Figure 5–3, *Individual Package Selection*).

On the left side of the screen you will see a directory listing of various package groups. When you expand this list (double-click to select it) and double-click on a single directory, the list of packages available for installation will appear on the right.

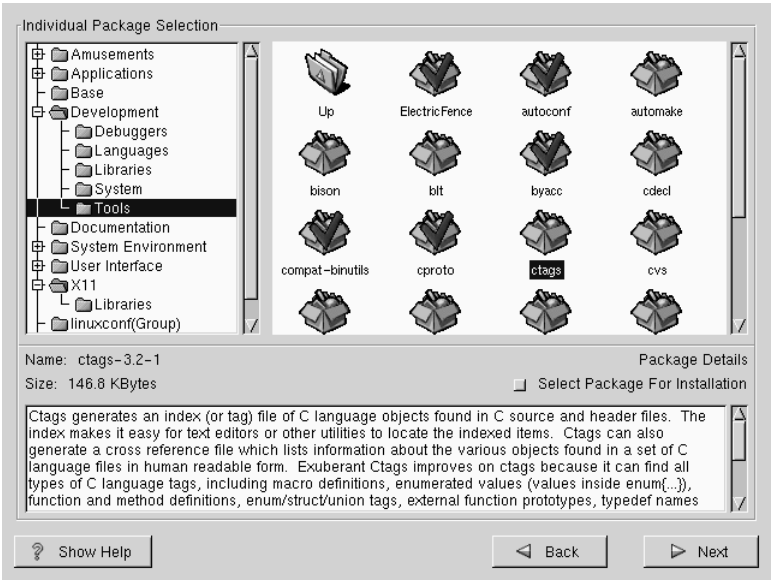
To select an individual package, double-click on it, or click on it once to highlight it and click on the **Select Package For Installation** button below. A red check mark will appear on any of the packages you have selected for installation.

To read information about a particular package before choosing it for installation, left-click on it once to highlight it, and the information will appear at the bottom of the screen along with the name and size of the package.

Please Note

Some packages (such as the kernel and certain libraries) are required for every Red Hat Linux system and are not available to select or deselect. These **base packages** are selected by default.

Figure 5–3 Individual Package Selection

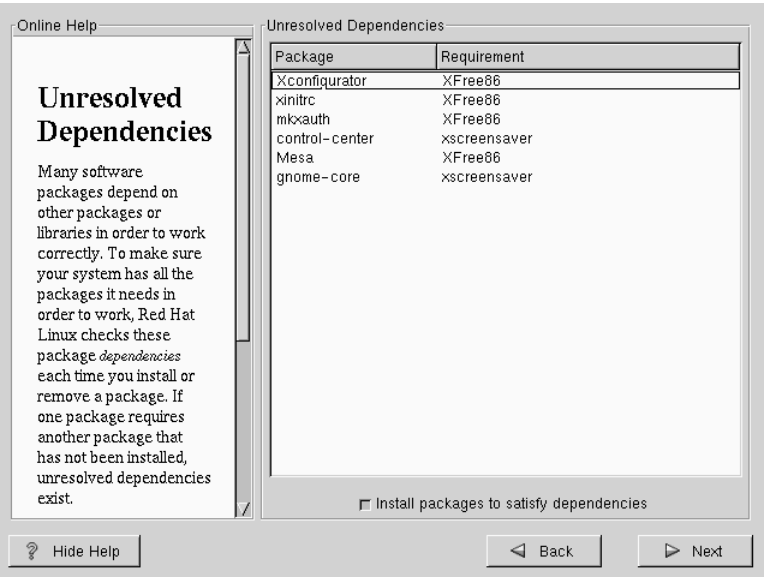


5.4.1 Unresolved Dependencies

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 5–4, *Unresolved Dependencies*).

The **Unresolved Dependencies** screen will only appear if you are missing certain packages that are needed by your customized package selection. Under the list of missing packages, there is an **Install packages to satisfy dependencies** check box at the bottom of the screen which is selected by default. If you leave this checked, the installation program will resolve package dependencies automatically by adding all required packages to the list of selected packages.

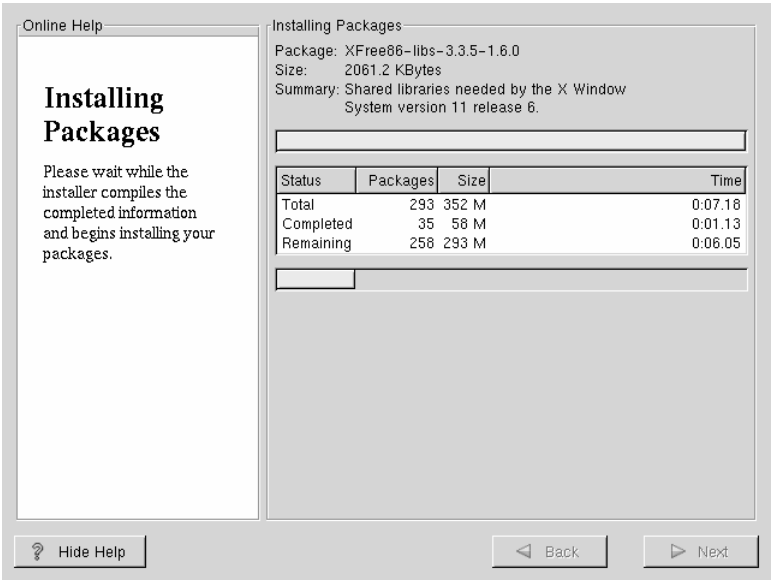
Figure 5–4 Unresolved Dependencies



5.5 Upgrading Packages

At this point there's nothing left for you to do until all the packages have been upgraded or installed (see Figure 5–5, *Installing Packages*).

Figure 5–5 Installing Packages



5.6 Upgrade Complete

Congratulations! Your Red Hat Linux 6.2 upgrade is now complete!

You will now be prompted to prepare your system for reboot. Don’t forget to remove any diskette in the floppy drive or CD in the CD-ROM drive. If you do not have LILO installed, you’ll need to use your boot disk now.

Tip

If you need a quick review of some of the basic concepts of Red Hat Linux refer to the *Official Red Hat Linux Getting Started Guide*.

For information dealing with system configuration and administration, refer to the *Official Red Hat Linux Reference Guide*.

Figure 5–6 Upgrade Complete

Part II Red Hat High Availability Server — Technology Overview

6 Post-Installation Security Overview

After installing Red Hat High Availability Server 1.0, you will need to consider the security-related aspects of deploying a highly-available server environment. As a first step in this process, the installation program makes the following changes to the network environment over that used in a standard Red Hat Linux 6.2 installation.

6.1 The /etc/hosts.deny File Set to Deny All Access

The tcp_wrappers utility is used to control access to many network services. Therefore, the default `/etc/hosts.deny` is set to `ALL: ALL` (meaning that all remote hosts are denied access to all services). In addition, `/etc/hosts.allow` contains no active entries, meaning that any service run out of inetd would fail to start and a message to that effect is recorded in the system message logs. You should edit `/etc/hosts.allow` to include only trusted systems that should be allowed access.

The following man pages are part of tcp_wrappers:

- `hosts.allow`
- `hosts.deny.5.gz`
- `hosts_access` (Section 3)
- `hosts_access` (Section 5)
- `hosts_options`
- `tcpd`
- `tcpdchk`
- `tcpdmatch`

6.2 System Logging Set to 60 Days

System logging is extended to keep up to 60 days worth of logged activity. This is the minimum legal requirement for commercial enterprises in most European countries. Normal log rotation is achieved through the configuration file `/etc/logrotate.conf` and any additional files found in the `/etc/logrotate.d/` directory.

6.3 Using `chkconfig` to Enable Services

Many system services are disabled by default, but can be reactivated through the use of `chkconfig`. Use the following command to list the services that are disabled for all runlevels:

```
chkconfig --list | grep -v :on
```

You may notice that there are some services that you wish to use; in which case `chkconfig` can be used to enable them. For example, if you'd like to enable `named`, simply issue the following command:

```
chkconfig named on
```

See the `chkconfig` man page for more information.

6.4 Password Lifetime Set to 30 Days

The maximum number of days a user password may remain unchanged is 30 days. This includes the root account! You may change this default by editing `/etc/login.defs` and changing the value for `PASS_MAX_DAYS`.

6.5 Some Services Not Installed

By default, some services that are normally installed during a traditional Red Hat Linux installation are not installed during a Red Hat High Availability Server installation. The services may be installed by performing a custom installation, and selecting the desired services. Examples of such services include `finger`, `talk`, `wall`, and `news`.

6.6 This is Only the Beginning…

We would like you to think of these measures as 'entry level' and not a complete security solution. We would also invite you to hunt down additional network resources to help improve your system's security. Your first stop should always be http://www.redhat.com; here you can find the latest information and updates to your software. Often these updates include important security-related fixes. You can go directly to the Red Hat "Updates & Errata" page by pointing your browser at http://www.redhat.com/support/updates.html.

In addition, various websites such as http://www.freshmeat.net/, http://lwn.net/ and http://slashdot.org/ can be valuable resources. Please consider allocating regular timeslots either weekly or monthly to look for updates. Always remember — security is never a task that you can consider "completed". It is an ongoing process.

7 Failover Services (FOS)

7.1 An Overview of FOS

The Piranha technology that is part of Red Hat High Availability Server consists of two types of clustering technologies:

- LVS (Linux Virtual Server)
- FOS (FailOver Services)

With FOS, a person can set up a two-node Linux cluster consisting of an active system and a standby system. The nodes will monitor each other and, if the specified IP services (ftp, http, etc.) fail on the active system, the services are switched over and provided by the standby system. No additional systems or devices are required, and except for the temporary loss of the service(s) during failure, the failover process is transparent to end users.

A Piranha FOS cluster consists of two nodes: one functioning as the **active** provider of IP services to the public client network, and the other (called the **inactive** node) monitors those services and operates as a standby system. When any service on the active node becomes unresponsive, the inactive node becomes active and provides those services instead. Services are automatically started and stopped as needed during the transition. If the failed system comes back online it will become the new inactive node and monitor for failures on the now-active system.

7.1.1 Features

FOS consists of the following features:

- Any IP service that supports a direct socket connection can be monitored and can be migrated to the inactive node during failover. This includes user-created services. Failover will occur when the service becomes unresponsive to FOS monitoring; this is independent of the number of users actually using the service. Possible services includes:
  - Web (http)

  - ftp, inetd
  - telnet
  - lpd
  - smtp/sendmail
  - ssh
  - LDAP
  - Firewall services
  - Other IP or user services

- The system administrator can specify special send/expect script strings as part of service monitoring for increased assurance that the service is functioning properly.
- FOS automatically starts and stops the monitored service as part of failover. System administrators can specify the start and stop commands or scripts (with arguments) for each monitored service. Custom scripts are also permitted.
- Although the nodes need to be identical in terms of FOS operation and configuration, the clustered systems do not need to be dedicated entirely to FOS. They can be used for additional purposes beyond the services being monitored.
- Although currently limited to clusters of two nodes, multiple independent clusters are possible [1].
- Each service can be defined as having a unique IP address, independent of the cluster's node addresses. This makes it easier to migrate existing environments where the IP services are already being provided by multiple servers, to having those services provided by a single "more fault-tolerant" cluster and keeping the changes transparent to end users.

7.1.2 Current Restrictions

FOS has the following restrictions:

[1] If clusters of more than two nodes are required, LVS must be used instead of FOS. For more information on LVS, please turn to Chapter 8, *Linux Virtual Server (LVS)*.

- Specified services are monitored and failover as a group. Services do not failover individually, nor are they load-balanced between the systems.
- Only services supporting direct socket connections can be monitored. Services requiring connections to secondary ports apart from the listening port cannot be monitored. Services not currently supported include:
  - nfs
  - ntp/daytime
- Shared data services (pop, imap, smtp) must have their data NFS-mounted from a common exported source in order to maintain data delivery.
- Send/expect strings are limited to printable text characters only (plus `\r` and `\n`). Binary data cannot be explicitly sent or tested.
- FOS must currently start and stop the monitored services as part of the failover process to ensure reliability. The services cannot already be running on the inactive node. This may reduce the usefulness of the inactive node while it is operating as a standby system.
- Only two-node clusters are supported. Both nodes must be Linux systems.
- Because several IP services are handled by the `inetd` daemon rather than individual daemons, there are situations where a non-FOS-configured service can be affected by FOS if `inetd` is involved. For example; if you choose to monitor ftp, and ftp is started and stopped by `inetd`, then when FOS shuts down `inetd` other services `inetd` provides (such as `rsh`) will also become unavailable on that system.
- The Piranha Web Interface does not, at present, provide a means to copy the changed configuration file to the other nodes in the cluster, nor an option for restarting FOS so it can use the updated file. These operations must be done manually. This restriction is expected to be removed in a future release.

7.1.3 Software Location

The FOS software is contained in the Piranha RPM files. The main RPM (called `piranha`) contains the FOS binaries, the `piranha-gui` RPM contains the Piranha Web Interface for configuring the system, and the `ipvsadm` RPM contains the `ipvsadm` program, which is used to administer the virtual-server-specific aspects of Red Hat High Availability Server. FOS documentation and source code are also in the `piranha-docs` RPM and the `piranha` source RPM, respectively.

To obtain possible updates of these (or other) packages, visit http://www.redhat.com/apps/support/updates.html on the Red Hat website.

7.1.4 Base Requirements

FOS requires two identical (or near-identical) Linux systems, both accessible from the public client network. In addition, all services being handled by FOS must be configured identically on both systems. This is because both systems must operate from identical FOS configuration files.

The source configuration file is created and maintained using the Piranha Web Interface. This interface requires that Apache and PHP be installed and configured on both cluster nodes. Details concerning the Piranha Web Interface can be found in Chapter 9, *The Piranha Web Interface — Features and Configuration.*

7.2 FOS Architecture

While the architecture of an FOS cluster consists of relatively few components, it is important to completely understand how these components work together to provide highly-available services. This section discusses FOS in more detail.

7.2.1 Primary and Backup Nodes

A node in an FOS cluster is defined as being either a **primary** or a **backup** system. The two types operate identically except in two situations:

- In cases where both nodes of a two node cluster attempt to declare themselves as actively providing services (a **quorum tie**), the primary node will always win the stalemate and force the backup system to become an inactive standby.

- An FOS cluster will always have a primary node; there are no configuration options to eliminate it. If a single-node cluster is defined (a configuration allowed by Piranha for use with other components, but of no value in an FOS setup), then that single node must be configured as a primary. Note that this is not the same situation as when a two node cluster has one node down — in that case the remaining node will be either a primary or backup, depending on which node failed, and which is still running.

7.2.2 Cluster Node States

All nodes in a running FOS cluster will be operating in one of the following states at any point in time:

Table 7–1 FOS Node States

State	Description
Active	The node is providing the configured IP services to the public users. Only one node in an FOS cluster is allowed to be the active node at any point in time.
Inactive	The node is acting as a standby system while the other node (sometimes referred to as its **partner**) is active. The inactive node monitors the services on the active node, and will become the active node if it detects one of those services failing to respond.
Dead	The node is down, or its services are non-responsive.

7.2.3 Heartbeats

Each of the cluster nodes send a periodic heartbeat message to the network, with an indication of whether that node is currently active or inactive. Each node expects to see a heartbeat message from its partner. If it is not received, this is considered a

failure of that system and may result in a failover of services. This test is independent of the IP service monitoring.

When the inactive node fails to see the heartbeat of the active node, it treats the missing heartbeat as indicating a cluster failure, and will perform a failover of all services. If the active node fails to see a heartbeat from the inactive node, the inactive system is logged as being unavailable for failover, while the services continue normal operation on the active node.

7.2.4 Virtual IP (VIP) Addresses

Failover in an FOS cluster is accomplished through the use of VIP (Virtual IP) addresses. They are virtual because they exist as additional IP addresses to the node's regular host IP address. In other words, a node can have multiple IP addresses, all on the same network interface. A node can be accessed by its VIP address(es) as well as by its regular host address.

VIP addresses are a feature of Linux and can be defined on any network interface present. For FOS, the VIP addresses and their network interfaces have to be accessible by the clients on the public network.

7.2.5 Services

Each service defined in FOS requires a VIP address, a port number, a start command (or script name), and a stop/shutdown command (or script name). Each service can be defined as having a different VIP address, or some (or all) can use the same VIP address. Services currently cannot failover individually — when one service fails, they all failover to the inactive system. This means that in most cases there is little value in specifying individual VIP addresses for services. However, there are some cases where this may be desirable:

- You have an existing environment where the IP services are already being provided by different servers, and you are using FOS to consolidate them to a single, more fault-tolerant cluster. In this case, using their previous IP addresses as VIP addresses will allow you to migrate without needing to change any client systems.

- You anticipate long-term growth in the use of each service. Using different VIP addresses now may make it easier to migrate the individual services to separate, dedicated FOS clusters in the future, while reducing the possible impact of changes on client systems.

In general, however, it is recommended that you use the same VIP address for all FOS services. Because a single VIP address means only one VIP address must be moved from the active node to the inactive node during failover, a single VIP address means faster, more reliable failovers.

Each service is also allowed two optional parameters: a **send** string and an **expect** string. If specified, these strings will be used as part of the service monitoring that will test whether the service is actually responding. If they are not specified, the service will be considered functional if a socket connection attempt to it succeeds.

7.2.6 Service Monitoring

On the inactive node, a monitoring daemon is run for each FOS service on the active node. Each monitoring daemon, called nanny, periodically tests a service on the active node. The test goes through the following steps:

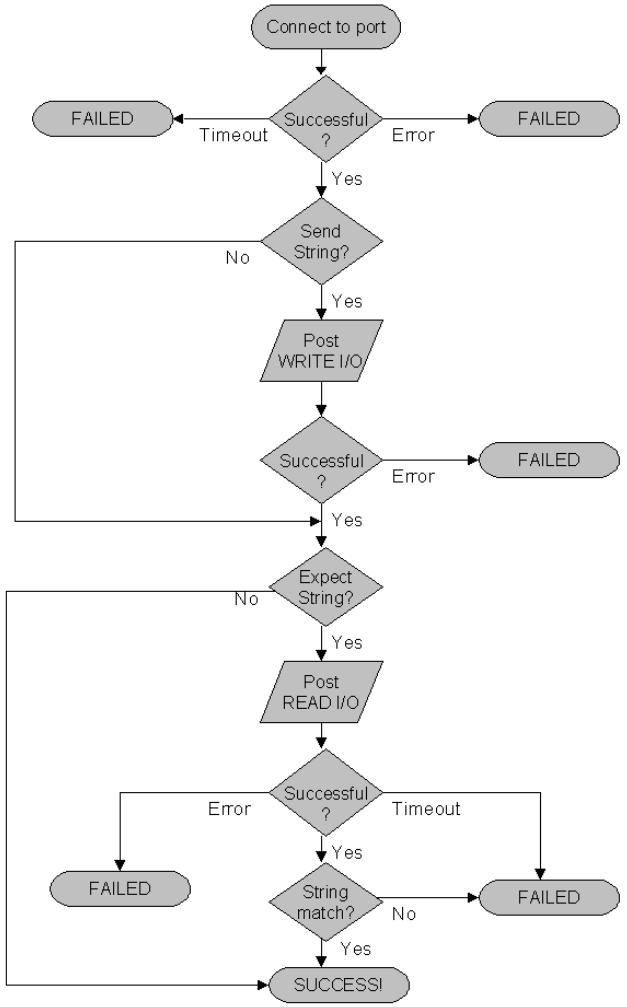
- The first test is whether a connection to that service's tcp/ip port is successful or not. If an error results, that service is considered dysfunctional and a failover occurs. Otherwise, the test continues.
- If it has been supplied, a character string (the send string) is sent to service's port. If an error occurs, the service is considered dysfunctional, and a failover occurs. Otherwise, the test continues.
- If an expect string is supplied, an attempt to receive a response takes place. If an error occurs, a response is not received, or the response does not match the expect string, the service is considered dysfunctional, and a failover occurs. Otherwise, the service is considered functional.

When nanny monitors a service, it connects using the active node's host IP address rather than the VIP address of the service. This is done to ensure cluster reliability. There are windows during service failure (and the subsequent failover) where the VIP address may exist on both cluster nodes, or be missing altogether. Using the host IP

address instead of the VIP address to monitor a service ensures that the correct system is always being examined and tested.

The following diagram illustrates the service monitoring logic used by FOS:

Figure 7–1 Service Monitoring Logic

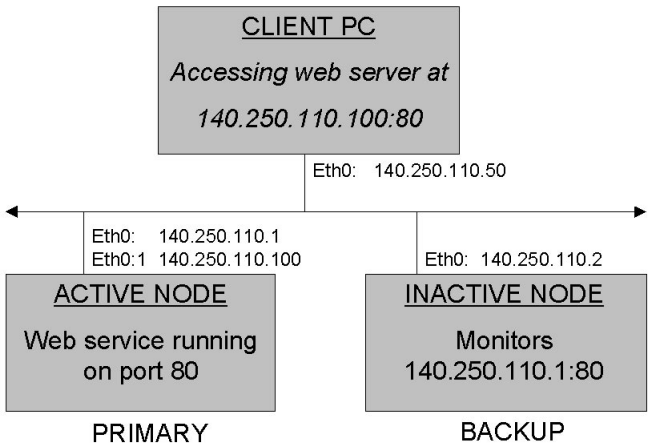


7.2.7 Failover

FOS automatically creates, deletes, or moves VIP addresses based on the information in its configuration file. Each time FOS changes a VIP address, ARP broadcasts are sent out to inform the connected network that the MAC address for the VIP address has changed. If an end-user accesses a service by referring to its VIP address and port, it will be transparent which system is actually providing that service.

In normal operation, an FOS system will have one active node with running services (and their associated VIP addresses), and an inactive node monitoring the services on the active node. This is illustrated below:

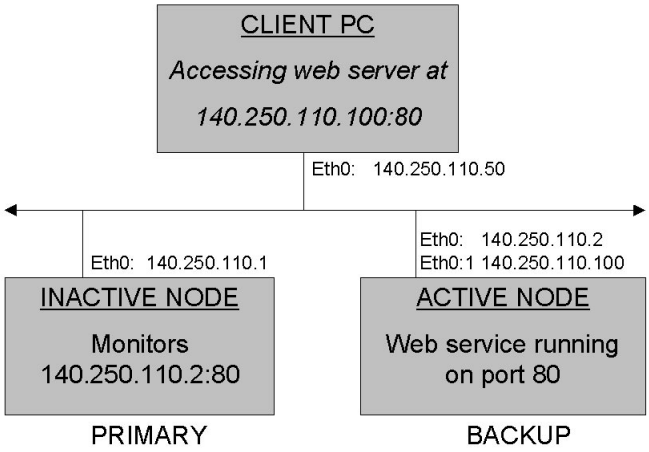
Figure 7–2 Running FOS Cluster Before Failover



When a failover occurs, the service VIP addresses are recreated on the inactive node, and the inactive node becomes active by starting the services. The originally-active system is notified (by heartbeat) that it should become inactive (if possible, depending on the failure situation). If it does go inactive, it will stop all services, start the

monitoring programs, and become eligible for a failover should the new active system suffer an outage. This is illustrated below:

Figure 7–3 Running FOS Cluster After Failover



If, for some reason, the services on the originally-active system cannot be stopped, it does not interfere with the cluster, because the VIP addresses have been moved to the new active system, directing all traffic away from the originally-active system.

7.2.8 Components

An FOS system consists of the following components:

Table 7–2 FOS Components

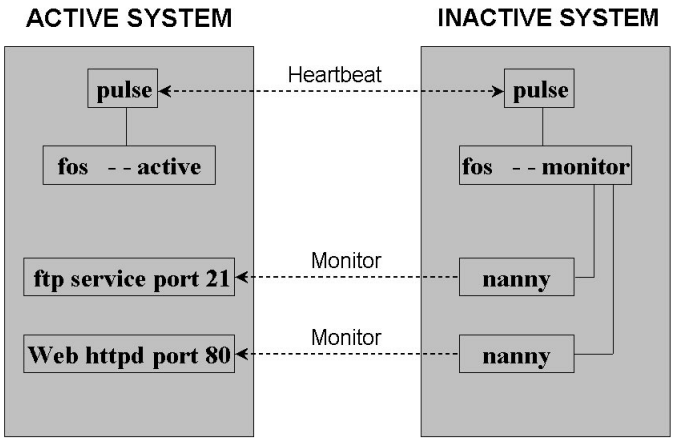
Component	Description
Piranha Web Interface	A graphical interface for creating and maintaining the cluster configuration file. (Please read Chapter 9, *The Piranha Web Interface — Features and Configuration* for more information on the Piranha Web Interface.)
`/etc/lvs.cf`	The cluster configuration file. Can be any filename desired; this is the default. The FOS-related contents of this file are detailed later in this document.
`/usr/sbin/pulse`	Main Piranha program and daemon process. Provides and tests for a heartbeat between the cluster nodes. Also starts and stops the fos daemon process as needed.
`/etc/rc.d/init.d/pulse`	Start and stop script for the pulse program.

Component	Description
`/usr/sbin/fos`	Main FOS program and daemon. Started by pulse, this program operates in two modes. On the active node, it is started using a `--active` option which causes it to automatically start and stop the IP service(s). On the inactive node, it is started with a `--monitor` option which causes it to start and stop the nanny service monitoring daemon(s). When a failure is detected by the inactive node, the fos daemon initiates a failover by exiting, which in turn causes pulse to restart it using the `--active` option, and to notify the partner cluster node that it is to go inactive.

Component	Description
/usr/sbin/nanny	Service monitoring program and daemon. Started by fos, there is one nanny daemon for each defined service to monitor. The nanny processes only run on the inactive system, and monitor the services on the active system for failure. If a failure is detected, the nanny daemon notifies the fos daemon of the failure by exiting, which in turn causes fos to terminate all other nanny processes. Then fos exits to notify the pulse daemon that a failure has occurred.
/usr/sbin/send_arp	Program used by fos to broadcast to the public network which system is currently providing the service for a VIP address.

The components on a running FOS system supporting two services looks like this:

Figure 7–4 Components of a Running FOS Cluster



7.3 Setting up an FOS Cluster

This section discusses the steps necessary to configure an FOS cluster.

7.3.1 Minimal Prerequisite Information

The information needed to set up an FOS cluster falls into two categories:

- Cluster-wide data
- Service-specific data

The following tables list the minimal information needed in both categories to set up an FOS cluster:

Table 7–3 Required Cluster-Wide Information

Item	Comment
Ability to gain root access	Required to edit and copy the configuration file, and to start/stop FOS (via the pulse daemon).
Host IP address	Each cluster node's IP address must be included in the configuration file.
rsh (or optionally, ssh)	Each time the configuration file is set up or changed, it must be copied to all nodes in the cluster. Either `rcp` (in the case of rsh) or `scp` (in the case of ssh) will be used to perform the actual copy. Note that this copy is done as root. You must decide whether to use rsh or ssh.
Proper configuration of rsh or ssh	After deciding whether to use rsh or ssh, you must then configure the primary and backup nodes such that the root account may copy files between the two nodes without being prompted for a password.

Table 7–4 Required Per-Service Information

Item	Description
Service name	A general name for the service (such as "ftp1"). This name will be used to identify the service in the configuration and system log files.

Item	Description
Port number	The TCP/IP port number used by this service. The port number will be used by FOS to test whether the service is functional. For http web services, this port is usually 80. For ftp, port 21 is normally used. The service monitoring tests are described in Section 7.2.6, *Service Monitoring*.
Start command	The command or script used to start the service. The full path must be specified, and parameters are allowed. For most Linux services, the start command will be `/etc/rc.d/init.d/`*`xxxxx`* `start` (where `xxxxx` is the name of a service script, such as `httpd`, or `inetd` (for services, such as ftp, that are controlled by inet). In some cases, you may want to create your own scripts and reference them.

Item	Description
Stop command	Similar in concept to the start command referenced above, except that this is the command or script to execute to stop the service. The full path must be specified, and parameters are allowed.
VIP address (and device)	The virtual IP address used by clients on the public network to access this service. This address must be different from the IP address of either of the cluster nodes. This address will be manipulated and broadcast on the public network as part of the failover process. Services do not failover individually, so typically you would define all of the services to use the same VIP address. Also needed is the network device name (visible to the public network) to be used by the VIP address. Typically, the cluster nodes are connected using device `eth0`, so the first VIP address on that interface would be placed on `eth0:1`, a second VIP address would use `eth0:2`, and so on.

In addition, the following optional parameters can be specified to increase the level of service testing:

Table 7–5 Optional Per-Service Information

Item	Description
Send string	A string of printable characters to send to the service's port as part of service monitoring. The string can contain spaces, `\n`, and `\r`. During service testing, if a connection attempt to the service's port is successful, the send string will then be sent to that port. If an error occurs, then the service is considered dysfunctional.

Item	Description
Expect string	A string of printable characters that are expected from the service's port [2]. The string can contain spaces, `\n`, and `\r`. The string can also be a single asterisk ("*") character, indicating that a response is mandatory, but can consist of any characters. If the number of characters received is less than the length of expect string, or if any of those characters do not match the corresponding character in the expect string, the service is considered dysfunctional.
Timeout	The maximum number of seconds to wait for a connection attempt to complete before considering the attempt a failure. Also used as the number of seconds to wait for a read attempt to complete (for comparison to the service's expect string) before considering the attempt a failure. Default value is 10 seconds.

7.3.2 The Piranha Configuration File (`lvs.cf`)

The Piranha configuration file is used in several different cluster configurations, so some portions of the file do not apply to FOS. The file is broken down into three major sections:

- Global cluster information

[2] Note that a service may send out a message in response to a connection, or in response to a specific message written to the service's port once a connection has been made.

- LVS-specific information
- FOS-specific information

Only the first and last sections are applicable to FOS. The default name and location of the configuration file is `/etc/lvs.cf`.

Each time the configuration file is modified, it must be copied to all nodes in the cluster. All nodes must use the same configuration data. Mis-matched configuration files are one of the easiest ways to cause a dysfunctional cluster.

Each line in the configuration file must follow strict formatting:

- Only one space before and after the the equal sign ("=")
- No missing mandatory information
- Numeric fields must contain only numbers
- All characters after a "#" are treated as a comment

Any violation of these formatting rules will cause the pulse program to fail (sometimes with a core dump). Using the Piranha Web Interface is highly recommended and will ensure a properly-formatted configuration file.

The `piranha-docs` RPM provides a sample configuration file. It can be found in `/usr/doc/piranha-docs*/sample.cf`. The `lvs.cf` man page provides details on every possible entry in the configuration file.

The following tables list the configuration file entries that are applicable to setting up FOS. All items are required unless otherwise noted.

Global Cluster Entries

This table lists the global settings needed to define the cluster:

Table 7–6 Piranha Configuration File Settings — GLOBAL

Entry	Description
`service = fos`	Indicates that this configuration file is for FOS rather than lvs.

Entry	Description
`primary = nnn.nnn.nnn.nnn`	Host IP address of the primary node in the cluster.
`backup_active = 1`	Indicates the backup node will be a member of the cluster.
`heartbeat = 1`	Indicates that heartbeats will be used in the cluster.
`heartbeat_port = nnnnn`	UDP port number to use for heartbeat. Need only be changed if multiple clusters exist and you need to prevent conflicts.
`keepalive = nnnnn`	Number of seconds between heartbeats.
`deadtime = nnnnn`	Number of seconds that must elapse without seeing a heartbeat from the partner system before declaring that it has failed. This number should be a multiple of the `keepalive` value.
`rsh_command = xxx`	Specified the command to use to copy files between systems. Must be either `rsh` or `ssh`.

Entry	Description
`network = nat`	This entry is not used by FOS but must be set to a valid value. Leave as `nat`.
`nat_router =` *`nnn.nnn.nnn.nnn`* `eth`*`n`*`:`*`n`* (Should be on one line)	This entry is not used by FOS, but must be set to a valid value. Place any IP address you want here, as long as it does not conflict with the host IP address of the cluster nodes, or the VIP addresses of any services. Also specify a valid device (such as `eth0`), but make sure the virtual interface number (`:`*`n`*) does not conflict with any of the device definitions for the FOS services.

Per-Service Settings

For each service to include in FOS, there must be a block of data, enclosed in braces ({}). Each service block begins with the word `failover`, followed by a user-defined service name. The name is only used to tell the services apart, and only appears in the log files.

Table 7–7 Piranha Configuration File Settings — PER-SERVICE

Entry	Description
`failover` *`xxxx`* `{`	Starts a block definition for a failover service. *`xxxx`* can be any name desired, as long as it contains no whitespace or quotes. NOTE THE OPENING BRACE ON THIS LINE!
`active = 1`	Indicates that this service is to be included in FOS. A `0` means that this definition block is to be ignored

Entry	Description
`address = nnn.nnn.nnn.nnn` `ethn:n` (Should be on one line)	Specified the VIP address and device for this service. The VIP address is usually the same for all services (and if so, the device entry must also be the same), but it can be different if desired. The device consists of two parts: the `ethn` part indicates the ethernet interface to use (the first interface would be named `eth0`, the second would be named `eth1`, and so on), while the `:n` part is a number indicating the number of the VIP address for the specified interface (`:1` for the first VIP, `:2` for the second, and so on).
`port = nnnnn`	The TCP/IP port number of this service. http is usually 80, ftp is usually 21, etc. The port number is used to test whether the service is responding or failing.
`send = "xxxxx"`	*OPTIONAL*. If specified, this string will be sent to the port as part of service monitoring. See the `lvs.cf` and nanny man pages for details.

Entry	Description
`expect = "`*xxxxx*`"`	*OPTIONAL*. If specified, this string is expected in response to connecting to the service's port and/or sending the `send` string. The `expect` string must follow the same rules as the `send` string; it may also contain a single asterisk ("*"), which means that any character(s) received will be considered as matching the `expect` string. The response from the port is compared (up to the length of the `expect` string), and a match indicates that the service is functional. See the `lvs.cf` and `nanny` man pages for details.
`timeout =` *nn*	*OPTIONAL*. If an `expect` string is specified, this entry indicates the number of seconds to await a response from the port before assuming a timeout failure. Default is 10 seconds.
`start_cmd = "`*xxxx*`"`	The script or command to perform in order for FOS to start this service. The command must include the full pathname. Parameters to the command/script may also be included, but must be separated by single spaces only.

Entry	Description
`stop_cmd = "`*xxxx*`"`	The script or command to perform in order for FOS to stop this service. The command must include the full pathname. Parameters to the command/script may also be included, but must be separated by single spaces only.
`}`	The closing brace indicates the end of this service definition.

The `start_cmd` and `stop_cmd` Entries

Both the `start_cmd` and the `stop_cmd` entries are mandatory, and must be enclosed in double quotes. The specified value is the command or script to be executed that will start or stop the service. The full path must be specified. Parameters may also be specified; each one must be separated by a single space. The command used should be repeatable, meaning that no problems (other than a possible error return value) should occur if the command is executed several times in a row.

The `send` and `expect` Strings

The `send` string (`send = "`*xxxx*`"`) is a text string to send to the service port in order to test whether the service is really functioning. The length is limited to 255 printable, quoteable text characters. Also permitted are `\n`, `\r`, `\t`, and `'`.

The `expect` string is similar to the `send` string. If it is specified, a read will be attempted on the service port, and the resulting response compared to the `expect` string. If the read attempt times out, or the number of characters read are less than the length of the `expect` string, then the service has failed. If the read is successful, then the `expect` string is compared to the response and if the characters match up to the length of `expect` string, the service is considered functional; otherwise it is considered to have failed. If an `expect` string is not specified, no read attempt will occur.

The `expect` string must follow the same rules as the `send` string, except for one additional feature. If `expect = "*"` is specified, then the service must send a response, but that response can consist of any characters and be any length.

Please Note

If the `send` string is omitted, then no attempt to send data to the service will occur *EXCEPT* if `port = 80`, and the `expect` string is also omitted. In this case, a web service is assumed and an internal http test string will be sent and expected. This condition exists for backwards compatibility with previous releases of Piranha.

If both `send` and `expect` strings are specified, the `send` string will always be sent first.

The following table lists samples of `send` and `expect` strings for some of the most common IP services.

Please Note

These strings are samples only. They may not be the best choices for all circumstances, and may not work on all systems.

Table 7–8 Sample send and expect Strings

Service	Port	send String	expect String
http/www	80	`HTTP / HEAD/1.0\n\n`	`HTTP`
ftp	21	`\n`	
telnet	23	`\n`	*

Service	Port	send String	expect String
lpd	515	\n	lpd:
ssh	22		SSH
inetd	98	\n	500
login	513	\n	
bind	952	\n	
smtp/sendmail	25	\n	220

7.3.3 File Copying with rsh and ssh

Piranha (as well as the system administrator) needs to be able to copy files between the cluster nodes (usually as root). There are also non-FOS situations where Piranha needs to execute commands on cluster nodes as part of statistic gathering. These tasks can be accomplished using either rsh or ssh.

One of these two applications must be specified in the configuration file. Each application also has specific configuration requirements (software installation, creation/modification of `.rhosts` files, etc.) that must be met in order for it to work. Consult the rsh or ssh man pages for further information.

7.3.4 Setting up an FOS Cluster Step-by-Step

Here is a sample step-by-step procedure for installing, configuring, and starting a new Piranha FOS cluster.

1. Make sure you have the basic resources for setting up the cluster readily available. This includes:
   - The TCP/IP addresses and hostnames for the network interfaces, and/or a functioning DHCP server environment.
   - A list of the TCP/IP services, and their port numbers, that will be set up to failover. Some examples might include http on port 80, ftp on port 21, etc.
   - Access to a nearby client system with a running web browser for setting up the Piranha software, and for testing the http service. You can also set

up the server as a functioning workstation with web browser, but you will need to select **CUSTOM** during the product installation in order to install those components.

- The ability to use a text editor such as vi or emacs to edit configuration files.

2. Install the product by following the on-screen displays, and by following the instructions in the installation section of this document.

3. Log into both nodes as root and perform the following basic operations:

   - Execute the `/usr/sbin/piranha-passwd` script to set an access password for the Piranha Web Interface.
   - Edit the `/etc/hosts` files and add entries for each of the cluster nodes.
   - Edit `/etc/hosts.allow`, and enable access for the appropriate service(s) for all cluster nodes. Use the commented examples in the file as a guideline.
   - Edit the `~root/.rhosts` files and add entries for each of the cluster nodes so that the root account can be used with `rsh` and `rcp`.
   - If desired, you may also want to set up `ssh` and `scp` in addition to (or instead of) using `rsh` and `rcp`. Follow the appropriate instructions for that software.

4. Make sure that each system can ping the other by IP address and name. At this point, copying files between the systems using `rcp` (or `scp` if set up) when logged in as root should also work. As an example, the following command should work (assuming you will be using `rcp`):

```
rcp myfile node2:/tmp/myfile
```

5. Configure Apache on both nodes by editing `/etc/httpd/conf/httpd.conf`, and setting the `ServerName` parameter appropriately. Start Apache by using the `/etc/rc.d/init.d/httpd` script, and passing the `start` or `restart` parameter as appropriate.

Although the following should have been done for you after you've installed Red Hat High Availability Server, the following configuration sections must be set in order for the Piranha Web Interface to work properly. First, the following entry should be present in the `/etc/httpd/conf/httpd.conf` file:

```
<Directory /home/httpd/html/piranha>
  Options All
  AllowOverride All
</Directory>
```

You should also find the following entries in the same file:

```
LoadModule php3_module        modules/libphp3.so
AddModule mod_php3.c
…
<IfModule mod_php3.c>
  AddType application/x-httpd-php3 .php3
  AddType application/x-httpd-php3-source .phps
</IfModule>
```

6. Log into the client system and start up the web browser. Access the URL `http://`*xxxxx*`/piranha/` where *xxxxxx* is the hostname or IP address of the PRIMARY node in the cluster. You should see the configuration page for the Piranha software.

   Configure the software as needed for your setup, following the information detailed in other sections of this document. Your changes should be present in the file `/etc/lvs.cf` on that cluster node.

7. Make sure the configuration files on all nodes are identical by copying the new file on the primary node to the other node by using the `rcp` or `scp` command (for example):

   ```
   rcp /etc/lvs.cf node:/etc/lvs.cf
   ```

   If this does not work, you will have to investigate the configuration changes for the `rsh` or `ssh` software you made earlier.

Changes to the configuration file will require that the file be re-copied to all the nodes, and that Piranha be stopped and restarted. (Note: A future release of Piranha may automate this process.)

8. On each node, one at a time, disconnect the node from the network and try the following tests (detailed in later sections of this chapter).

   - Start and stop the IP services you intend to use in FOS by typing the exact commands specified in the services' `start` and `stop` lines, as listed in the cluster configuration file.
   - With the service running, use `telnet` and attempt to connect to that service's TCP/IP port (for example):

     ```
     telnet localhost 80
     ```

     If this connection is refused, then that service might not be working or the port number is incorrect. Note that if `telnet` does connect to the service, the telnet session may be "hung" afterwards and you may have to terminate the `telnet` process in order to disconnect.

9. With the services down and the system connected to the network, start the `pulse` program on the primary node by executing the following command:

   ```
   /etc/rc.d/init.d/pulse start
   ```

   After a time, the Piranha software and all the services should be running. Keep watch on this by using the `ps -ax` command, and by examining the tail end of the `/var/log/messages` log file.

10. Start the `pulse` program on the other cluster node. Both nodes should now be running and monitoring each other.

11. You can test failover by disconnecting the network connection on the active node, shutting down a monitored TCP/IP service on the active node, or by terminating a `nanny` process on the inactive node by using the command `kill -s SIGTERM` *nnn* (where *nnn* is the pid of a `nanny` process).

Your FOS cluster should now be completely operational.

7.4 Testing FOS

While configuring FOS is a relatively straightforward process, unless care is taken it is easy to define a non-functioning cluster, or one that "ping-pongs" a failover back and forth between the nodes. This is partly because all the services failover as a group, so in order for FOS to operate, all services on at least one node in the cluster must come up correctly. If one service on node *A* fails to come up (due to a configuration error), and a failover occurs, but a different service on node *B* also fails to come up, then the result is a cluster where neither node comes up correctly and failover will bounce back and forth.

Before starting Piranha using a newly created or modified configuration file, some basic tests should be performed first. These tests should be carried out on both nodes in the cluster.

7.4.1 Synchronize Configuration Files

Ensure that both systems are using the same configuration by copying the piranha configuration file to the other cluster node(s). You can use either `rcp` or `scp`, depending on whether you've configured `rsh` or `ssh`:

```
# rcp /etc/lvs.cf other.cluster.node:/etc/lvs.cf
other.cluster.node: Connection refused
Trying krb4 rcp...
other.cluster.node: Connection refused
trying normal rcp (/usr/bin/rcp)
#
```

7.4.2 Starting and Stopping Services Manually

Without Piranha running, bring up the IP services by manually typing the same command that is defined in the `start_cmd` line for each service's entry in the configuration file. Then use the `ps` command and make sure the services are running as expected.

Next, type the same command that is defined in the `stop_cmd` line for each service's entry in the configuration file. Then use the `ps` command to make sure the services have been shut down properly.

To continue testing, bring up the IP services again by manually typing the same command that is defined in the `start_cmd` line for each service's entry in the configuration file.

7.4.3 Using telnet to Test send and expect Strings

This test ensures that the port used by each service is correct, and that the strings used in service monitoring reflect the actual behavior of each service. One of the most common configuration errors is that a service's port number (as defined in the cluster configuration file) does not match the port number actually used by the service.

With the services already running (see Section 7.4.2, *Starting and Stopping Services Manually*), use the `telnet` command to attempt to connect to that service's TCP/IP port number (as defined in the configuration file). For example, if your cluster is providing http service on port 4040, use the following command to confirm that port 4040 is, in fact, in use:

```
# telnet localhost 4040
Trying 127.0.0.1...
Connected to localhost (127.0.0.1).
Escape character is '^]'.
```

Because we didn't get a Connection refused error message, we can be confident that a service is using port 4040 [3]. At this point, we can disconnect from the port in this manner:

```
^]

telnet> quit
Connection closed.
#
```

[3] Of course, this does not mean that the service *we expected* is using the port. We'll get to that in a moment.

Please Note

When telnet is used to connect to some services, you may find you cannot disconnect. In this case, you will have to use the `kill` command on telnet.

If you *did* get a Connection refused message, then the service you started is not using that port and Piranha will fail to connect also. On the other hand, if telnet did connect sucessfully, then FOS will be able to connect as well.

Note, however, that not every service can be considered "telnet-friendly" — some services may drop the connection after a period of time, or there might be no response at all (try pressing [Enter] several times). This does not necessarily indicate that the service is not functioning. Some services simply cannot be accessed in an interactive telnet session.

As a final test, try using telnet to connect to the service from the *other* cluster node. If this fails, then FOS will also fail. Resolve any problems preventing telnet from connecting and it is likely that the nanny daemon will also be able to connect.

If you have defined `send` strings for a service, you can try typing the contents of the `send` string and see if it the service responds appropriately. However, in some cases you will not be able to type it fast enough to prevent the connection from timing out and failing to process what you've typed. This does not necessarily indicate that there is any problem; just that you cannot use telnet to test. The only way to be sure is to see if the nanny daemon can use the `send` and `expect` strings sucessfully.

7.4.4 Using ps and Log Files

Start Piranha by starting the pulse daemon on one cluster node (have that node unplugged from the network if necessary). Depending on the node you are using, FOS will either attempt to start the IP services right away, or will start the nanny daemon(s), which will then timeout, initiate a failover, and start the IP services. In either case, you should end up with a running environment where pulse is running, fos is running in `--active` mode, and the IP service(s) are up. You should be able to see this using the `ps -axw` command.

You should also examine the system log file (`/var/log/messages`) and read the Piranha-related entries. Make sure they do not indicate any problem (see Section 7.5.2, *Error Messages* for a discussion of possible error messages). If a problem *is* logged, make sure you resolve it before proceeding.

Repeat this process for the other cluster node.

7.4.5 Starting FOS For the First Time

If you disconnect both cluster nodes from the network, and start Piranha on each, you should end up with each node believing that the other has failed and both nodes should be running with active services. If you then connect the nodes to the network while they are still running, the two systems will detect each other's heartbeat. Since both nodes are claiming to be active, the node that is defined as the backup node should become inactive by shutting down its IP services and starting the nanny daemon(s).

7.4.6 Forcing a Node Failover

The easiest way to test failover is to unplug the active node from the network. The inactive node should detect the loss of a heartbeat message and become active. If you then shut down Piranha on the unplugged system, reconnect it to the network, and restart Piranha, the reconnected system should detect that FOS is *already* active on the other node and no failover should occur to interrupt the running service.

If you reconnect the disconnected node with Piranha already running, then Piranha will detect that two active systems are running; The one defined by the `backup` entry of the configuration file should become inactive (even if that results in another failover).

7.4.7 Forcing a Service Failover

The easiest way to cause a failover due to the loss of a single service is to stop the service by manually executing the command defined by that service's `stop` entry in the Piranha configuration file. The command should be issued on the active system. This will stop the service, which should cause the nanny daemon on the inactive node to log a service failure and trigger a failover.

7.5 Trouble-shooting FOS

This section describes the most common problems and causes/solutions. This section, along with the earlier section describing how to test FOS, should help resolve most situations.

7.5.1 Common Problems / Questions

Please Note

Most of the components that make up Piranha can be run manually (provided you supply the same option switches that they require when run as daemons). In addition, most components also support a `-v` and/or a `--norun` option to aid in the debugging process.

The nanny daemon keeps reporting that a service is not working

If you are using `send`/`expect` strings as part of service testing, try using FOS without them. Without these strings, FOS will test the service by connecting to the service's port only. If this resolves the problem, it is likely that the service is not passing your `send`/`expect` string testing.

FOS keeps performing failovers even though the services are running

Even though the services are running, it does not necessarily mean that FOS can connect to them. The best test for this is to bring up FOS on just one cluster node (which will cause all the services to be started), and use telnet from the other cluster node to attempt to connect to the TCP/IP port defined for each service. If telnet returns a Connection refused message, then it is likely that the nanny daemon will also fail to connect to the service. Resolve the situation preventing telnet from connecting and the nanny daemon should stop initiating failovers.

The most common causes for this failure are either that the service has been configured to use a different TCP/IP port number than the one specified in the Piranha configuration file, or the Piranha configuration files are not identical on the cluster nodes.

Next, shut down Piranha on the active node and bring it up on the inactive node, then perform the telnet tests in the opposite direction.

Also check the system log files (on both nodes). For any failover condition, Piranha will log the reason.

If Apache is one of the services, make sure that the `LISTEN` port number matches the port number in the Piranha configuration file.

FOS keeps "ping-ponging" the failover back and forth

First see the previous section on "FOS keeps performing failover even though the services are running", and perform the telnet testing described.

If you still experience this problem, make sure that your `keepalive`, `deadtime`, and the service's `timeout` values are not too short. Try increasing the values to ensure that Piranha has sufficient time to correctly determine a failure. Also note that `deadtime` should have a value that is a multiple of the `keepalive` value.

Piranha did not shut down correctly; how do I kill it without causing it to restart?

If you must kill the Piranha daemons manually, then you must kill them "from the top down". In other words, you must first kill pulse, then fos, and then the nanny daemons. Killing fos first (for example) will cause pulse to think a failover should occur.

Use the command `kill -s SIGTERM <pid>` first, before using SIGKILL (also known as `-9`). In most cases, killing pulse or fos with `SIGTERM` will cause it to automatically kill all of its own children.

When I unplug the active node, a failover occurs, but when I plug it back in, *another* failover occurs back to the first node

This occurs because the system you unplugged was the primary node, which caused the backup node to become active. The unplugged node was still active (even if it had lost network connectivity). Plugging it back in created the situation where both nodes were declaring themselves active; in this case, the primary node always wins the stalemate. Therefore, even though the backup node was also the currently active node, it failed over and became inactive.

There are two ways to prevent this:

- If possible, make sure the system you unplug is the backup system. If the backup system is also the active system (meaning that the primary system is inactive), then unplugging the backup system will cause a failover to the primary, but plugging the backup system back in will not cause a second failover (because it will lose the "both nodes active" stalemate).
- The second method will always work no matter which node is unplugged. Just make sure that before the node is plugged back in, you first shut down Piranha. Then plug the node back into the network, and restart Piranha. The node will detect that there is already an active system and will start up as the inactive node.

Piranha is causing the private log files for each service to fill up with connect attempt messages

ftp/inetd and httpd have individual log files. Each connection attempt (by any software, for any reason) may cause a one or two line entry in these log files. Because the nanny daemons, being part of Piranha's service monitoring, must connect to each service on a regular basis, there is no way to configure Piranha to prevent this, short of disabling service monitoring.

The only work-arounds are to set up a cron or backup entry to "roll over" the file(s), preventing them from filling the disk, and/or increase the timeout parameters in Piranha's configuration file so the services are tested less often and therefore log fewer entries. Other possibilities, such as using symbolic links to the null device, could result in a loss of important security information and are not recommended.

Can I set up web services that are independent of FOS?

Apache can be configured to start multiple httpd daemons that listen on different TCP/IP ports. This means that you can configure one port for use with Piranha, and another that acts independently for other purposes.

Because this is more of an Apache configuration issue than a Piranha configuration issue, information on configuring Apache in this manner is beyond the scope of this document. Please see the Apache Software Foundation's website (http://www.apache.org/) for additional information on Apache configuration.

How can I set Piranha up so that I can use the Piranha Web Interface on the inactive node?

The Piranha Web Interface is designed to be used on the active node, with the resulting configuration file copied to the inactive system. It is possible to use the Piranha Web Interface on the inactive node if you start a second httpd daemon that uses a different TCP/IP port from the http service defined in the Piranha configuration file. Note that this is a similar situation to the previous question.

Can I have the services already running on the standby system, instead of having Piranha starting them?

Yes, you can replace the `start_cmd` and `stop_cmd` lines in the `/etc/lvs.cf` file with commands that do not affect the running services. However, this will result in a cluster configuration that can only survive one failover; in this case, a second failure will not cause the services to failover to the original node.

7.5.2 Error Messages

Most of the error messages are self-explanatory. Here are descriptions for some of the less obvious or more critical ones.

"Service type is not 'fos'"

You are attempting to run the fos program manually, but the Piranha configuration file does not have `service = fos` set.

"gratuitous *xxx* arps finished"

Each time a VIP address is created or removed, Piranha sends out ARP broadcasts to notify the network of the change in MAC address for that IP address. This message indicates that those broadcasts have completed.

"Incompatible heartbeat received – other system not using identical services"

An attempt is being made (either due to mis-matched configuration files or manual startup of a Piranha component) to start one cluster node using lvs services and the other node in fos services. All nodes in a cluster must use the same Piranha cluster service.

"Notifying partner WE are taking control!"

A situation has occurred where the backup system needs to become (or already is) the active cluster node, and it is telling the primary node (which is trying to become active, or already is) that it must switch to inactive mode.

"PARTNER HAS TOLD US TO GO INACTIVE!"

This message is the partner to the one listed above. A situation has occurred where the backup system needs to become (or already is) the active cluster node, and it is telling the primary node (which is trying to become active, or already has) that it must switch to inactive mode.

"Undefined backup node marked as active? – clearing that..."

The Piranha configuration file has set `backup_active = 1`, but there is no backup node IP address defined in the file. This message appears if Piranha is then started with its configuration file containing this error. As the message implies, Piranha is treating the situation as if `backup_active = 0` was set in the configuration file instead.

"pulse: cannot create heartbeat socket – running as root?"

The pulse daemon cannot start because it cannot create the TCP/IP socket needed for its heartbeat. The most common reasons for this is either pulse is being started by someone with a UID other than 0 (a non-root account), or pulse (or another Piranha daemon) is already running.

"no service active & available..."

The most common cause for this message occurs when no services in the Piranha configuration file are set as `active = 1`. If they are all set as inactive, then FOS has no services to control or monitor.

"fos: no failover services defined"

Piranha has `service = fos` enabled, but there are no FOS services defined in the configuration file.

7.6 Additional Information

7.6.1 Multiple Clusters

While a Piranha FOS cluster can currently only be composed of two nodes, you can create multiple, independent clusters. When doing this, make sure that unique `heartbeat_port` and VIP addresses are used for each cluster.

7.6.2 Further reading

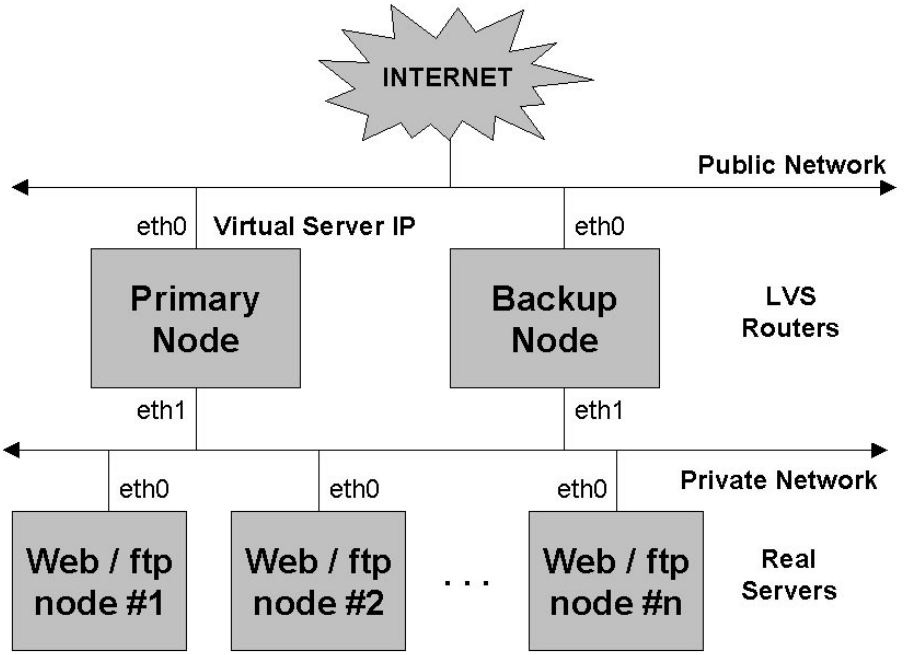
More information can be found in the fos, pulse, nanny, and `lvs.cf` man pages.

8 Linux Virtual Server (LVS)

8.1 Introduction

A Linux Virtual Server (LVS) cluster is a collection of servers that have been specially configured to provide highly-available services. The diagram below illustrates how an LVS cluster works.

Figure 8–1 A Basic LVS Cluster Configuration



An LVS cluster consists of one or two router nodes (top of figure) and a variable number of application servers (bottom). We will refer to the LVS router nodes as

LVS routers, and to the pool of application servers as **real servers**. Note that these terms designate *roles* rather than machines. For example, the LVS routers shown in Figure 8–1, *A Basic LVS Cluster Configuration* might be configured to perform both LVS router and real server roles.

Service requests arriving at an LVS cluster are addressed to a **virtual server** IP address (sometimes refered to as a VIP address). This is a publicly-advertised address that an administrator at the site associates with a fully-qualified domain name (for example, lvs.ajax.com). The illustration shows only one virtual server address, but there may be more than one. The important thing to remember is that a VIP address will migrate from one LVS router to the other during a failover, thus maintaining a presence at that IP address. As such, they can be considered **floating** addresses. VIP addresses may be aliased to the device (for example, `eth0:1`) that connects the LVS routers to the public network, or each could be associated with a separate device.

Only one LVS router is active at a time. The role of the **active router** is to redirect service requests from a virtual server address to the real servers. The redirection is based on one of four supported load-balancing algorithms (described in Table 8–1, *Load-balancing Methods*). The active router dynamically monitors the health of the real servers, and the workload on each, via one of three supported methods (described in Table 8–2, *Enabling Synchronization and Monitoring Tools*). If a real server becomes disabled, the active router stops sending jobs to that server until it returns to normal operation.

The backup router performs the role of a **standy system** . Periodically, the LVS routers exchange "I’m alive" heartbeat messages. Should the backup node fail to receive a heartbeat message within an expected interval, it initiates a **failover** and assumes the role of the active router. During failover, the backup router takes over the VIP address(es) serviced by the failed router and uses a technique known as **ARP spoofing** — It starts announcing itself as the destination for IP packets addressed to the failed node. When the failed node returns to service, it assumes the hot backup role.

8.1.1 Routing Methods

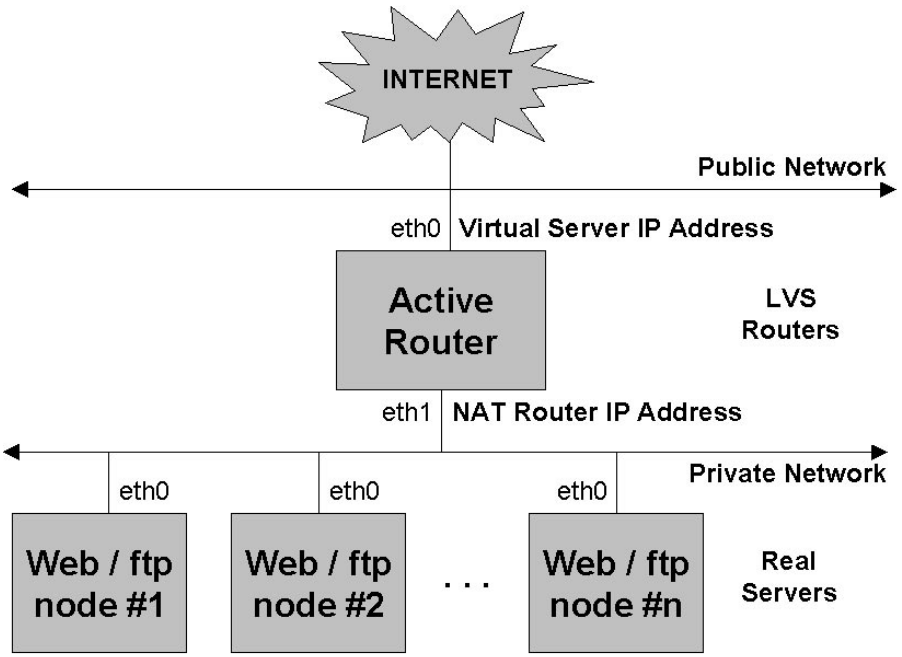
Three routing methods are supported:

- Network Address Translation (NAT)
- IP encapsulation (tunneling)
- Direct routing

Each LVS router is configured to use a single routing method. If you set up multiple LVS clusters, you can use different routing methods for each cluster.

The diagram below illustrates NAT routing works.

Figure 8–2 An LVS Cluster Implemented with NAT Routing



The NAT router IP address is the default route used by each real server on the private network to communicate with the active router. Like the virtual server address, it

is also a floating IP address. The NAT router address may be aliased to the device (for example, `eth1:1`) connecting the LVS routers to the network of real servers, or associated with a separate device on each router.

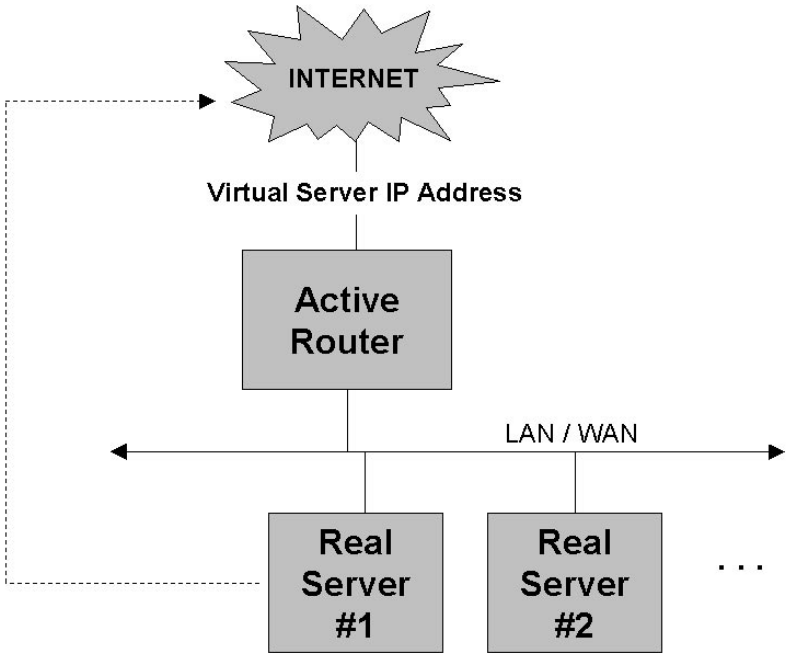
Also like virtual server addresses, NAT addresses are enabled only on the active router. Therefore, should the active router fail, the backup router enables the virtual server and NAT addresses during take-over of the floating IP addresses. In the topology shown in Figure 8–2, *An LVS Cluster Implemented with NAT Routing*, virtual server addresses are enabled on device `eth0` and the NAT router address on `eth1`.

As a real server processes a request, it returns packets to the active router, where the address of the real server in the packets is replaced by the virtual server address. In this manner, the private network of real servers is masqueraded from the requesting clients.

NAT routing is easy to set up and flexible. The real servers may be any kind of machine running any operating system or Web server, or any combination. The chief disadvantage is that, after about twenty real servers, the LVS router may become a bottleneck because it must process outgoing as well as incoming requests.

Figure 8–3, *An LVS Cluster Implemented with IP Encapsulation (tunneling)* illustrates how a tunneling virtual server works. With this method, IP encapsulation is enabled on the real servers. Prior to assigning a job to a real server, the active router encapsulates the IP address of the requesting client inside the real server's address.

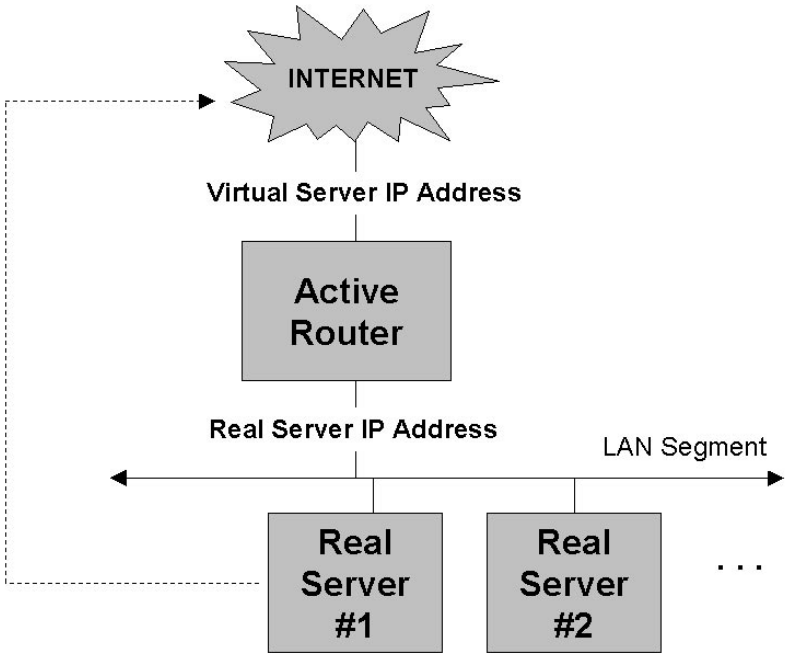
Figure 8–3 An LVS Cluster Implemented with IP Encapsulation (tunneling)



When a real server has processed a job, it returns the response directly to the requesting client rather than to the active router. Thus, the active router processes incoming requests only. An additional advantage of tunneling is that the real servers may be geographically distributed. (While IP encapsulation using non-Linux real servers may be possible, only Linux real servers have been tested in this configuration.)

Figure 8–4, *An LVS Cluster Implemented with Direct Routing* illustrates direct routing. With this method, the LVS routers and real servers exist on the same LAN segment. After selecting the real server for a job, the active router changes the address of the request to that of the real server.

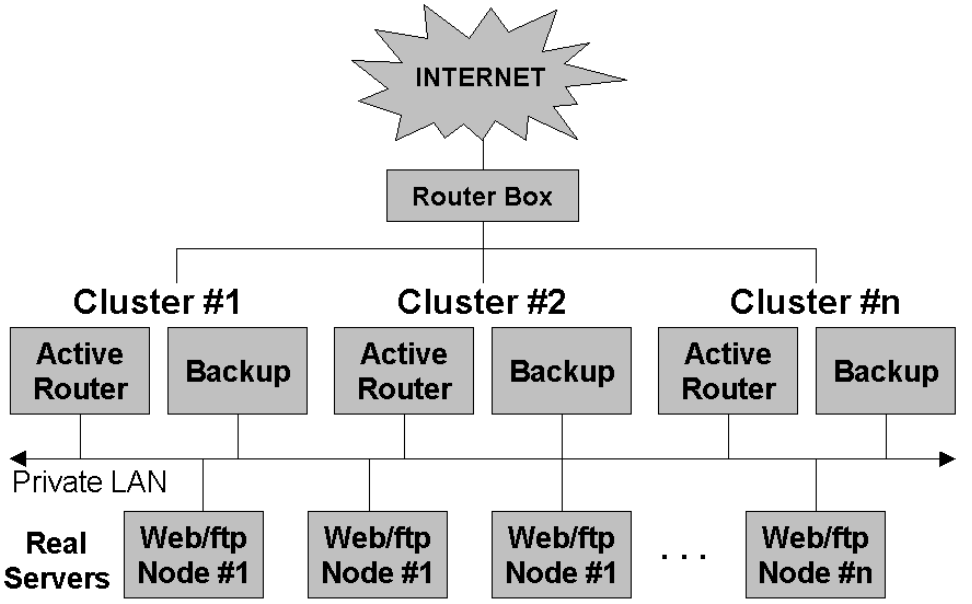
Figure 8–4 An LVS Cluster Implemented with Direct Routing



When a real server has processed a job, it returns the response directly to the requesting client rather than to the active router. This method avoids the (admittedly slight) overhead of tunneling.

Figure 8–5, *Combining LVS Cluster Routing with DNS Round Robin* suggests how multiple clusters could be deployed in combination with DNS round robin to satisfy the needs of the busiest sites. In this scenario, a DNS server resolves service requests to multiple virtual server addresses, which a router distributes equally among multiple LVS routers. These routers, in turn, distribute requests to a shared bank of real servers.

Figure 8–5 Combining LVS Cluster Routing with DNS Round Robin



8.1.2 Job Scheduling Methods

Regardless of the routing method, the **IPVS table** in the active router's kernel redirects requests from a virtual server address to a real server. For example, a TCP request addressed to port 80 on virtual server 1.2.3.1 might be routed to port 80 on real server 192.168.1.2. The actual mapping of jobs to real servers in the IPVS table is based on the load-balancing algorithm in use. Table 8–1, *Load-balancing Methods* describes the supported load-balancing methods.

Table 8–1 Load-balancing Methods

Name	Description
Round robin	Distribute jobs equally among the real servers.
Least-connections	Distribute more jobs to real servers with fewer active connections. (The IPVS table stores active connections.)
Weighted round robin	Distribute more jobs to servers with greater capacity. Capacity is indicated by a user-assigned weight, which is then adjusted upward or downward by dynamic load information.
Weighted least-connections	Distribute more jobs to servers with fewer active connections relative to their capacity. Capacity is indicated by a user-assigned weight, which is then adjusted upward or downward by dynamic load information.

8.2 Components of an LVS Cluster

The components of an LVS cluster are described below.

8.2.1 `pulse`

This is the controlling process that starts the other daemons as needed. It is started on the LVS routers by the `/etc/rc.d/init.d/pulse` script, normally at boot time . Through `pulse`, which implements a simple heartbeat, the inactive LVS router determines the health of the active router and whether to initiate failover.

8.2.2 `lvs`

The `lvs` daemon runs on the LVS routers. It reads the configuration file and calls `ipvsadm` to build and maintain the IPVS routing table.

8.2.3 nanny

The nanny monitoring daemon runs on the active LVS router. Through this daemon, the active router determines the health of each real server and monitors its workload. A separate process runs for each service defined on each real server.

8.2.4 /etc/lvs.cf

This is the LVS cluster configuration file. Directly or indirectly, all daemons get their configuration information from this file.

8.2.5 Piranha Web Interface

The Web-based tool for monitoring, configuring, and administering an LVS cluster. Normally this is the tool you will use to maintain /etc/lvs.cf, restart the running daemons, and monitor an LVS cluster.

8.2.6 ipvsadm

This tool updates the IPVS routing table in the kernel. The lvs daemon sets up and administers an LVS cluster by calling ipvsadm to add, change or delete entries in the IPVS routing table.

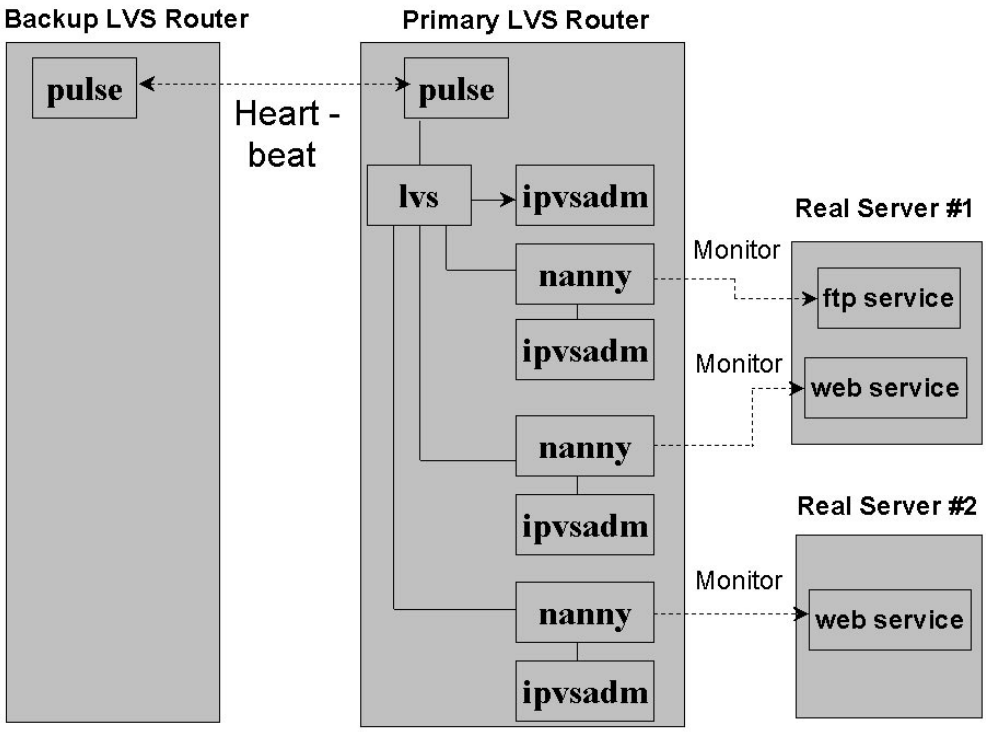
8.2.7 send_arp

This program sends out ARP broadcasts when the virtual server address changes from one node to another.

8.2.8 LVS Cluster — A Block Diagram

This diagram shows the how the various components work together in an LVS cluster configuration:

Figure 8–6 Components of a Running LVS Cluster



8.3 Background of the LVS Cluster

The Red Hat LVS cluster is based either directly on contributions from the Linux community, or on components that were inspired or enriched by various Linux community projects.

The primary source of the LVS cluster is Wensong Zhang's Linux Virtual Server (LVS) kernel routing algorithm (see http://www.linuxvirtualserver.org). The capabilities of the LVS project that the Red Hat LVS cluster currently supports are:

- Building virtual servers: floating IP addresses where requests for service arrive from the public internet.
- Routing service requests from virtual servers to a pool of real servers.
- Load-balancing (see Table 8–1, *Load-balancing Methods*).
- Packet-forwarding (see Section 8.1.1, *Routing Methods*).
- Persistent connections.

The LVS innovations supported by the Red Hat LVS cluster are based on a number of technologies. For a good general discussion and index of the relevant HOWTOs on these and related topics, see http://www.linas.org/linux/load.html.

8.4 Hardware/Network Requirements

An LVS cluster consists of one or two LVS routers and a collection of real servers providing Web, FTP, or other services. The connection and hardware requirements are described below.

A Linux server that will be the primary LVS router is required. If NAT routing is used, this machine needs two network adapters, one connecting it to the public network and the other to a private network (where the real servers are located). The private network is masqueraded from requesting clients. The servers on the private network may be any kind of computing platform running any operating system or Web server.

With IP encapsulation, the primary LVS router and real servers may be on the same LAN, or the real servers may be geographically dispersed (on a WAN). Real servers process requests directed to them and return responses directly to requesting clients.

With direct routing, the primary LVS router and real servers are on the same LAN segment. As with IP encapsulation, real servers return processed requests directly to requesting clients.

If you want failover capability, a second Linux server that will be the backup LVS router is required. This machine, a hot standby, needs to be configured exactly like the primary LVS router. For example, with NAT routing, the backup LVS router also needs two network adapters connecting it to the public network and to the private network of real servers. The adapter devices must match on the two LVS routers.

Thus, if devices `eth0` and `eth1` on the primary LVS router connect it to public and private network, respectively, then these same devices on the backup LVS router must connect it to the public and private network.

During configuration, you assign a **weight** to each real server. This is an integer reflecting each server's processing capacity (based on memory, processor speed, number of processors, etc.) relative to that of the others. It is the ratio between each server's weights that are significant. For example, if the weights of two servers are 2/1, 20/10, or 200/100, the result is the same — the first server has twice the capacity of the second. However, note that finer adjustments are possible if larger weights are used. The assigned weight, which may be dynamically adjusted based on load information, is used by two of the available job scheduling algorithms (described in Table 8–1, *Load-balancing Methods*). You should be prepared to assign accurate weights.

8.5 Routing Prerequisites

The LVS routers require Red Hat High Availability Server 1.0 or greater, and the type of routing you choose must be supported by your kernel/module configuration.

8.5.1 Enabling NAT

With NAT routing, packet forwarding, packet defragmenting, and IP masquerading must be enabled on the LVS routers.

Enable packet forwarding. To do this at system boot, make sure the file `/etc/sysctl.conf` contains the line `net.ipv4.ip_forward = 1`. To enable packet forwarding without rebooting, as root issue this command:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Enable packet defragmenting. To do this at system boot, make sure the file `/etc/sysctl.conf` contains the line `net.ipv4.ip_always_defrag = 1`. To enable packet defragmenting without rebooting, as root issue this command:

```
echo 1 > /proc/sys/net/ipv4/ip_always_defrag
```

To enable IP masquerading, issue this command:

```
ipchains -A forward -j MASQ -s n.n.n.n/type -d 0.0.0.0/0
```

where:

- *n.n.n.n* is the address of the private subnet to which the real servers are connected.
- *type* is 8, 16, 24, or 32, indicating the address type and mask:

```
netmask          | type | Subnet
~~~~~~~~~~~~~~~~~|~~~~~~|~~~~~~~~~~~~~~~
255.0.0.0        | 8    | Class A
255.255.0.0      | 16   | Class B
255.255.255.0    | 24   | Class C
255.255.255.255  | 32   | Point-to-point
```

You will probably want to put the `ipchains` command in an init script (e.g., `/etc/rc.d/rc.local`), so that masquerading is configured on the LVS routers at system startup.

`ipchains` is the tool used to create and manage firewalling rules set in the kernel's TCP stack. Masquerading is a small subset of these rules that allow machines making use of private IP networks to communicate with the outside world. Using `ipchains` can have an impact on system security. If you have security concerns, read the `ipchains` HOWTO (http://www.linux-doc.org/HOWTO/IPCHAINS-HOWTO.html).

8.5.2 Enabling IP Encapsulation

On each real server, establish a tunnel between it and each virtual server address. For example, these commands establish two tunnels (`tunl0` and `tunl1`) to two virtual server addresses (1.2.3.1 and 1.2.3.2):

```
ifconfig tunl0 1.2.3.1 up
ifconfig tunl1 1.2.3.2 up
```

To prevent real servers, rather than the active router, from intercepting ARP broadcasts, you also need to hide tunnels from ARP broadcasts. For example, these commands hide tunnels tunl0 and tunl1:

```
echo 1 > /proc/sys/net/ipv4/conf/all/hidden
echo 1 > /proc/sys/net/ipv4/conf/tunl0/hidden
echo 1 > /proc/sys/net/ipv4/conf/tunl1/hidden
```

8.5.3 Enabling Direct Routing

On each real server, enable a route to each virtual server address. For example, the following command aliases virtual server 1.2.3.1 to adapter `eth0`:

```
ifconfig eth0:0 1.2.3.1 up
```

You also need to hide virtual server routes from ARP broadcasts. For example, these commands hide any virtual server addresses on device eth0:

```
echo 1 > /proc/sys/net/ipv4/conf/all/hidden
echo 1 > /proc/sys/net/ipv4/conf/eth0/hidden
```

8.6 Persistence

In certain situations, it may be desirable (for a time) for a client to reconnect repeatedly to the same real server, rather than have an LVS load balancing algorithm send that request to the best available server at that momement. Examples of such situations include multi-screen web forms, cookies, SSL, etc. In these cases, a client may not work properly unless the transactions are being handled by the same server in order to retain context. LVS provides a feature to handle this called **persistence**.

When enabled, persistence acts like a timer. When a client connects to a service, LVS remembers that last connection for a specified period of time. If that same client IP address connects again within that period, it will be sent to the same server that it used previously — bypassing the load balancing mechanisms. When a connecton occurs outside the time window, it is handled according to the scheduling rules in place.

Persistence also allows you to specify a subnet mask to apply to the client IP address test as a tool for controlling what addresses qualify.

8.7 Cluster Node Interconnection Prerequisites

During configuration, you select the tool family (either `rsh` or `ssh`) that will be used to synchronize the `/etc/lvs.cf` configuration files on the LVS routers. This tool will also be used for parts of the data gathering used in determining proper load balancing. The selected tool must be enabled on the LVS routers, such that the root account on each router can log in to the other router without administrator intervention.

Also during configuration, you select the tool (`uptime`, `ruptime`, or `rup`) that the active router will use to monitor the workload on the real servers. Enable the selected tool on the real servers. If this cannot be done (for example, one of your real servers is a Windows/NT Web server), the cluster will still provide highly available services. However, the weighted round robin and weighted least-connections algorithms (described in Table 8–1, *Load-balancing Methods* will be affected. Namely, since load information will not be available, the user-assigned weights will be applied statically, rather than being dynamically adjusted based on server workload.

Table 8–2, *Enabling Synchronization and Monitoring Tools* describes in general terms the steps required to enable these tools on the source and destination hosts. For more detailed information, see the appropriate man page(s). Note that, with `rsh` and `ssh`, the root account must be able to log in over the network. To enable remote root login to a Red Hat Linux system, remove the following line from the file `/etc/pam.d/login`:

```
auth required /lib/security/pam_security.so
```

This is a security hole, albeit small. Make sure you have the LVS nodes properly firewalled so that logins are allowed only from trusted sources.

Table 8–2 Enabling Synchronization and Monitoring Tools

Tool	Do This
`rsh`	Create a `.rhosts` file with permission 600 in the root account's home directory (`/root`) on the destination host. There should be a line in the file naming the source host and user (for example, `foo.host1.com root`).
`ssh`	Obtain/install the tool (which for legal reasons cannot be released with international Linux distributions). On the source and destination hosts, disable remote login via all other methods, set up RSA-based authentication using `.ssh/authorized_keys`, and start `sshd`.
`uptime`	On each real server, enable either `rsh` or `ssh`, as described above.
`ruptime`	Set up each LVS router and real server to start `rwhod` whenever it boots.
`rup`	Set up each real server to start `rpc.rstatd` whenever it boots.

Please Note

The `rup` and `ruptime` programs require that the `rstatd` and `rwhod` daemons run on the system. Use one of the several available tools (such as `chkconfig` to enable these daemons.

8.8 Installing the Software

The software providing LVS functionality is released in the following RPM packages:

- `piranha` (programs)
- `ipvsadm` (virtual server administration tool)
- `piranha-gui` (HTML configuration tool)
- `piranha-docs` (documentation)

These packages are automatically installed during the Red Hat High Availability Server installation, which is described in Part I, *Installing Red Hat High Availability Server* .

To obtain possible updates of these (or other) packages for your hardware platform, visit http://www.redhat.com/apps/support/updates.html on the Red Hat website.

8.9 Configuring an LVS Cluster

You set up and maintain an LVS cluster from the LVS routers, by editing the configuration file and starting or restarting the `pulse` daemon. Specifically, the steps are:

1. On the primary router, edit the configuration file `/etc/lvs.cf`. This is best done using the Piranha Web Interface, which is described in Chapter 9, *The Piranha Web Interface — Features and Configuration.*
2. Copy the edited configuration file to the backup router.
3. Start (or restart) the `pulse` daemon; first on the active router, then on the backup router.

You can perform these steps from the shell, by editing the configuration file with the editor of your choice. The shell commands for starting, restarting, and stopping the `pulse` daemon are:

- `/etc/rc.d/init.d/pulse start`
- `/etc/rc.d/init.d/pulse restart`
- `/etc/rc.d/init.d/pulse stop`

The `pulse` daemon starts or restarts the other LVS cluster daemons as needed, which obtain their configuration information, directly or indirectly, from the current configuration file.

If you stop `pulse` (in order to shut down the cluster), stop it on the backup router first. This will prevent the backup router from initiating a failover when the active router goes down.

Alternatively, you can use the Piranha Web Interface to configure, monitor, and administer your LVS cluster. The entry fields on its windows set or change lines in `/etc/lvs.cf`.

The next section describes the LVS cluster configuration file. Read this section if you want to edit this file manually. If you chose to use the Piranha Web Interface, please read Chapter 9, *The Piranha Web Interface — Features and Configuration* instead.

8.9.1 Editing the Configuration File

The `/etc/lvs.cf` file has three sections. The global section, described in Table 8–3, *Setting Global Parameters*, sets up the LVS routers and specifies networking and heartbeat parameters. There is one set of parameters for the cluster. The per-virtual-server section, described in Table 8–4, *Setting Per-Virtual-Server Parameters*, defines virtual server addresses, sets up the associations between virtual servers and real servers, and specifies job-scheduling parameters. There is a separate set of parameters for each defined virtual server. The per-real-server section, described in Table 8–5, *Setting Per-Real-Server Parameters*, defines the real servers that will be load-balanced by each virtual server. There is one set of these parameters for each virtual server.

Let’s start with the parameters that are global.

Table 8–3 Setting Global Parameters

Parameter	Description
`primary = n.n.n.n`	Enter the IP address of the adapter connecting the primary LVS router to the public network.
`backup = n.n.n.n`	Enter the IP address of the adapter connecting the backup backup LVS router to the public network.
`heartbeat_port = n`	Enter the port number used for the heartbeat on the primary and backup LVS routers.
`keepalive = n`	Enter the number of seconds between heartbeats.
`deadtime = n`	Enter the number of seconds to wait before declaring a non-responding router dead and initiating failover.
`rsh_command = [rsh|ssh]`	Enter the command family to use for synchronizing the configuration files on the primary and backup routers. Important: as described in Table 8–2, *Enabling Synchronization and Monitoring Tools*, you *must* enable the selected command on the primary and backup routers.
`network = [nat|direct|tunnel]`	Enter the routing method to use when the LVS router contacts the real servers. Note that you will need to make the appropriate configuration changes to support the routing method you choose.

Parameter	Description
`nat_router = n.n.n.n ethn:n`	Enter the floating IP address and device of the NAT router. This IP address must be the default route used by each real server to communicate with the active LVS router. The IP address is aliased to the device (for example, `eth1:1`) connecting the LVS routers to the private network of real servers. The device must be the same (i.e., `eth1`) on both LVS routers.
`service = [lvs|fos]`	Enter the type of cluster service you wish to use. To take advantage of the features described in this chapter, you must choose `lvs`.

The next set of parameters are repeated for each virtual server.

Table 8–4 Setting Per-Virtual-Server Parameters

Parameter	Description
`virtual xxx {`	Enter a unique identifier for the virtual server. Note the brace ({) that starts this per-virtual-server block.
`address = n.n.n.n`	Enter the virtual server's IP address: a floating IP address that has been associated with a fully-qualified domain name.
`active = [0|1]`	Enable (1) or disable (0) this server.

Parameter	Description
`load_monitor = [`*`uptime`*`|`*`ruptime`*`|`*`rup`*`]`	Select the tool (default `uptime`) that will be used by the active router to monitor the workload of real servers. Important: as described in Table 8–2, *Enabling Synchronization and Monitoring Tools*, unless you enable the selected command on the real servers, the scheduling algorithms that use dynamic load information will apply the assigned weight statically rather than adjust the weight from load information. If you select the default (`uptime`), the tool you specified in the `rsh_command` parameter is used to log in to the real servers. This tool must be enabled on the real servers.
`timeout =` *`n`*	Enter the number of seconds (default `10`) that must lapse before a real server determined to be dead is removed from the routing table.
`reentry =` *`n`*	Enter the number of seconds (default `180`) that a restored real server must remain alive before being re-added to the routing table.
`port =` *`nnn`*	Enter the listening port for this virtual server: Typically, port 80 is used for http, and port 21 for ftp. However, any valid port can be used.

Parameter	Description
`persistent = nnn`	If specified, this enables persistence and defines the timeout (in seconds) of persistent connections. Persistence causes multiple requests from a client to be redirected to the same server each time (the server selected for the first request). The timeout value of persistent sessions is specified in seconds. The default is 300 seconds. Use this option to solve problems with cookies, SSL, or FTP with tunneling or direct routing.
`pmask = nnn.nnn.nnn.nnn`	If persistence is enabled, this parameter allows a granularity at which the clients are grouped. The source address of the request is masked with this netmask to, for example, direct all clients from a /24 network to the same real server. The default is 255.255.255.255, which means that the persistence granularity is per client host. Use this option to solve problems with non-persistent cache clusters on the client side.
`scheduler = [wlc|lc|wrr|rr]`	Select the scheduling algorithm (default `wlc`) for distributing jobs from this virtual server to the real servers. The choices are described in Table 8–1, *Load-balancing Methods*.
`}`	Ends the per-virtual-server-block.

The following parameters are repeated on a per-real-server basis.

Table 8–5 Setting Per-Real-Server Parameters

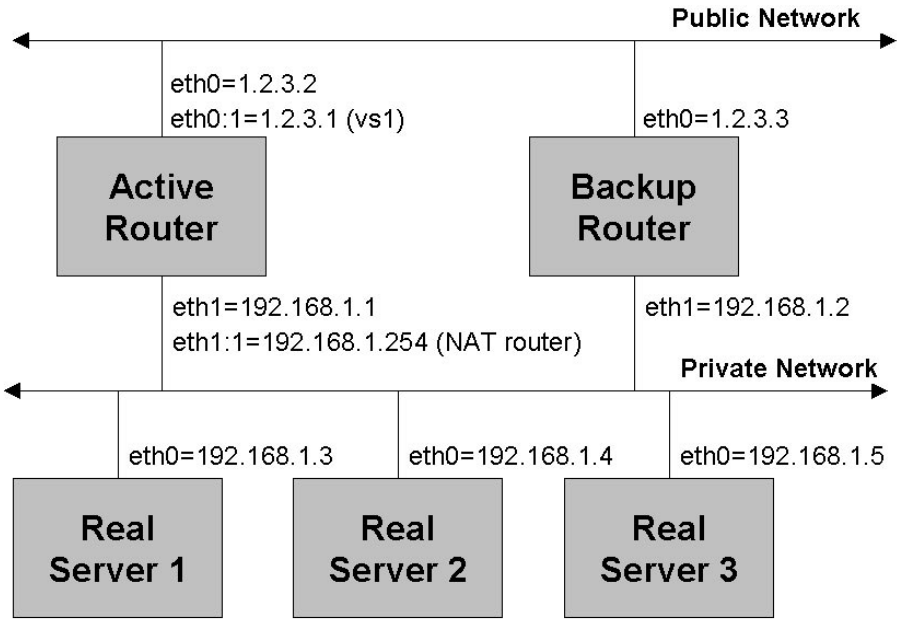
Parameter	Description
`server` *xxx* {	Enter a unique name for the real server. Note the brace ({) that starts this per-real-server block.
`address =` *n.n.n.n*	Enter the IP address of the real server on the private network.
`active = [`*0*`|`*1*`]`	Enable (`1`) or disable (`0`) the real server.
`weight =` *n*	Enter an integer (default is `1`) specifying this server's processing capacity relative to that of other real servers. For example, a server assigned a weight of 2000 has twice the capacity of a server assigned 1000. The weighted scheduling algorithms adjust this number dynamically based on workload.
}	Ends the per-real-server-block.

8.10 Example — Setting Up a Five-node Cluster

This section describes step by step how to create a cluster of two LVS routers and three Web/FTP servers. First, collect the necessary information and set up the five systems as explained in the next section. Then, implement the example either by editing the `lvs.cf` using a text editor or by using the Piranha Web Interface.

Figure 8–7, *Layout of the Example Network* shows the network that will exist after you've set up the LVS routers and real servers. All network addresses shown are for purposes of illustration only; you'll need to use addresses allocated to you from your network administrator.

Figure 8–7 Layout of the Example Network



8.10.1 Preliminary Setup

1. If you've not done so already, obtain a virtual server IP address. In our example this will be 1.2.3.1. Requests for service at the LVS cluster will be addressed to a fully-qualified domain name associated with this address.
2. Locate five servers and designate their roles:
   - One primary LVS router
   - One backup LVS router

- Three real servers

(Note that you'll also need a client system with a web browser, for testing purposes.)

The LVS routers must be Linux boxes running Red Hat High Availability Server 1.0 or later. The real servers may be any platform running any operating system and Web server.

Steps 3 through 14 set up the LVS routers.

3. On each LVS router, install two ethernet adapter cards, `eth0` and `eth1`.

4. Install the product by following the on-screen displays, and by following the instructions in the installation section of this document.

5. Log into both nodes as root and perform the following basic operations:

   - Execute the `/usr/sbin/piranha-passwd` script to set an access password for the Piranha Web Interface.
   - Edit the `/etc/hosts` files and add entries for each of the cluster nodes.
   - Edit `/etc/hosts.allow`, and enable access for the appropriate service(s) for all cluster nodes. Use the commented examples in the file as a guideline.
   - Edit the `~root/.rhosts` files and add entries for each of the cluster nodes so that the root account can be used with `rsh` and `rcp`.
   - If desired, you may also want to set up `ssh` and `scp` in addition to (or instead of) using `rsh` and `rcp`. Follow the appropriate instructions for that software.

6. Make sure that each system can ping the other by IP address and name. At this point, copying files between the systems using `rcp` (or `scp` if set up) when logged in as root should also work. As an example, the following command should work (assuming you will be using `rcp`):

```
rcp myfile node2:/tmp/myfile
```

7. Configure Apache on both nodes by editing `/etc/httpd/conf/httpd.conf`, and setting the `ServerName` parameter appropriately. Start Apache by using the `/etc/rc.d/init.d/httpd` script, and passing the `start` or `restart` parameter as appropriate.

   Although the following should have been done for you after you've installed Red Hat High Availability Server, the following configuration sections must be set in order for the Piranha Web Interface to work properly. First, the following entry should be present in the `/etc/httpd/conf/httpd.conf` file:

   ```
   <Directory /home/httpd/html/piranha>
     Options All
     AllowOverride All
   </Directory>
   ```

   You should also find the following entries in the same file:

   ```
   LoadModule php3_module        modules/libphp3.so
   AddModule mod_php3.c
   …
   <IfModule mod_php3.c>
     AddType application/x-httpd-php3 .php3
     AddType application/x-httpd-php3-source .phps
   </IfModule>
   ```

8. Log into the client system and start up the web browser. Access the URL `http://`*`xxxxx`*`/piranha/` where *`xxxxxx`* is the hostname or IP address of the PRIMARY node in the cluster. You should see the configuration page for the Piranha software.

   Configure the software as needed for your setup, following the information detailed in other sections of this document. Your changes should be present in the file `/etc/lvs.cf` on that cluster node.

9. Make sure the configuration files on all nodes are identical by copying the new file on the primary node to the other node by using the `rcp` or `scp` command (for example):

   ```
   rcp /etc/lvs.cf node:/etc/lvs.cf
   ```

If this does not work, you will have to investigate the configuration changes for the `rsh` or `ssh` software you made earlier.

Changes to the configuration file will require that the file be re-copied to all the nodes, and that Piranha be stopped and restarted. (Note: A future release of Piranha may automate this process.)

10. Create a public IP interface on `eth0` and a private IP interface on `eth1`. The public interface device (`eth0`) is the heartbeat device. The virtual server address is aliased to this device.

Interface	Primary node	Backup node
`eth0`	1.2.3.2	1.2.3.3
`eth1`	192.168.1.1	192.168.1.2

11. Designate an IP address (192.168.1.254) for the router device (`eth1`) connecting the active LVS router to the private network. This floating IP address will be aliased to the router device as `eth1:1`, and will be the gateway to the private network and the default route used by each real server to communicate with the active router.

12. On each LVS router:

    a. Enable packet forwarding. To do this at system boot, make sure the file `/etc/sysctl.conf` contains the line `net.ipv4.ip_forward = 1`. To enable packet forwarding without rebooting, as root issue this command:

    ```
    echo 1 > /proc/sys/net/ipv4/ip_forward
    ```

    b. Enable packet defragmenting. To do this at system boot, make sure the file `/etc/sysctl.conf` contains the line `net.ipv4.ip_always_defrag = 1`. To enable packet defragmenting without rebooting, as root issue this command:

    ```
    echo 1 > /proc/sys/net/ipv4/ip_always_defrag
    ```

c. Masquerade the private network. Issue this command and put it in `/etc/rc.d/rc.local`:

```
ipchains -A forward -j MASQ -s 192.168.1.0/24 -d 0.0.0.0
```

13. Decide whether to use `rsh` or `ssh` for synchronizing LVS cluster files. Verify that your choice is installed such that the LVS routers can log in to one another as root without administrator intervention (see Table 8–2, *Enabling Synchronization and Monitoring Tools*). In this example, we will choose `rsh`.

14. On each LVS router, make sure that Web service is configured and running, so the Piranha Web Interface can be used.

Please Note

Note: It is recommended that the http service on the routers be configured to use a different port than that used by http on the real servers. This will help prevent accidental conflicts or use of the wrong http service by client browsers.

Steps 15 through 18 set up the real servers.

15. On each real server, install an ethernet network card, `eth0`, and create an IP address on the same private subnet as in Step 3. Next, assign a weight to each server indicating its processing capacity relative to that of the others. In this example, `rs1` has twice the capacity (two processors) of `rs2` and `rs3`.

	rs1	rs2	rs3
`eth0`	192.168.1.3	192.168.1.4	192.168.1.5
weight	2000	1000	1000

16. On each real server, verify that the address named in Step 11 (192.168.1.254) is the real server's default route for communicating with the active LVS router.

17. Decide which program (`uptime`, `ruptime`, `rup`) will be used by the active router to monitor the workload on the real servers. If you choose `uptime`, each LVS router must be able to connect with each real server without administrator intervention, using the program you selected in Step 13. See Table 8–2, *Enabling Synchronization and Monitoring Tools* for general instructions. If the selected program cannot be enabled (for example, one of the real servers is an NT box), the scheduling algorithms that use dynamic load information will still work but the user-assigned weights will be statically applied rather than dynamically adjusted based on load.

18. Verify that each real server runs an installed and configured httpd server. Note that the real servers must listen on the same port (80 in the example) as the corresponding virtual server.

19. Verify (for example, using `telnet` or `ping`) that each real server can reach hosts on the public LAN. If a real server on the private network cannot reach a host on your LAN, this probably indicates a communication failure between the server and the active router. See Steps 15 and 16.

20. Determine the runtime parameters. For some of these, you may need to experiment over time to obtain optimal values. In this example, we will use the values listed.

Parameter	Value	Parameter Description
`heart-beat_port`	539	Number of the heartbeat listening port on the primary and backup routers.
`keepalive`	6	Number of seconds between heartbeats.

Parameter	Value	Parameter Description
deadtime	18	Number of seconds to wait for a non-responding router to respond before initiating failover. Should be a multiple of keepalive.
timeout	10	Number of seconds to wait for a non-responding real server to respond before removing it from the routing table.
reentry	180	When a real server that has been removed from the routing table starts responding again, wait this number of seconds before re-adding the server to the routing table.
scheduler	wlc	Use the Weighted least-connections load-balancing algorithm (assign more jobs to servers that are least busy relative to their load-adjusted weight). See Table 8–1, *Load-balancing Methods* for a description of the choices.
port	80	Virtual server port number. The listening port selected for the virtual server is used on the real servers as well.

21. Now we are ready to implement the example. You can do this by creating the configuration file with a text editor, or you can use the Piranha Web Interface configuration tool as explained in Chapter 9, *The Piranha Web Interface — Features and Configuration*. Using the Piranha Web Interface is highly recommended.

    Here is the resulting file. The number to the right of most lines represents the step in Section 8.10.1, *Preliminary Setup* discussing this setting.

```
primary = 1.2.3.2                    3
service = lvs
rsh_command = rsh                    13
backup_active = 1
```

```
backup =  1.2.3.3                    3
heartbeat = 1
heartbeat_port = 539                 20
keepalive = 6                        20
deadtime = 18                        20
network = nat
nat_router = 192.168.1.254 eth1:1    11
virtual vs1 {                        15
     active = 1
     address = 1.2.3.1 eth0:1        1
     port = 80                       20
     send = "GET / HTTP/1.0\r\n\r\n"
     expect = "HTTP"
     load_monitor = ruptime          17
     scheduler = wlc                 20
     timeout = 6                     20
     reentry = 15                    20
     server rs1 {                    15
         address = 192.168.1.3       15
         active = 1
         weight = 2000               15
     }
     server rs2 {                    15
         address = 192.168.1.4       15
         active = 1
         weight = 1000               15
     }
     server rs3 {                    15
         address = 192.168.1.5       15
         active = 1
         weight = 1000               15
     }
}
```

22. Copy the edited configuration file to the backup router:

```
rcp /etc/lvs.cf bkuprtr:/etc/lvs.cf
```

23. On the primary router, start the `pulse` daemon with this command:

```
/etc/rc.d/init.d/pulse start
```

24. Start `pulse` on the backup router, using the same command.

25. Use the `ps ax` command (or run the Piranha Web Interface) to monitor the running system.

8.11 Testing LVS

Once an LVS cluster is running, any attempt to connect to a service's VIP address (via browser or telnet for example), should result in that connection being answered by one of the real servers. Which real server will depend on the scheduling tool being applied.

The following commands are useful for testing or debugging an LVS setup:

- `ping` — A real server should be able to ping any system on the public network, while only the active LVS router should be able to ping a real server.
- `telnet <ipaddress> <port>` — Using `telnet` to connect to a virtual server VIP address should result in a real server responding to that attempt.
- `ipchains -l` — Lists the currently defined rules for forwarding ip messages.
- `ps ax` — When done on the active LVS Router, the `pulse`, `lvs`, and `nanny` daemons should all appear in the list. When done on the backup router, only `pulse` will be shown.
- `ipvsadm -l` — when done on the active LVS router, will list all the active services running on real servers, along with their scheduling method and weight.
- `lynx` — `lynx` is a text-based web browser and can be useful for testing the load balancing of the real server services. For example:

    1. Create or modify a web page on each real server that identifies which real server the page resides on. The best type of page is one that just displays a message like "This is real server #1", with no other content. The page should also include a very large comment section (a few KB worth of comments) that will add bulk to the page loading without affecting the display.
    2. On the active LVS router, execute the following command:

        ```
        while true; do lynx -dump nnn.nnn.nnn.nnn; done
        ```

where *nnn.nnn.nnn.nnn* is the VIP address for the real server's web service.

3. If load balancing is functioning, you should see the display alternate between "This is real server #1", "This is real server #2", and so on.

   While `lynx` is running, use the `ipvsadm -l` command to examine the current status of the services and their scheduling weight.

9 The Piranha Web Interface — Features and Configuration

The Piranha Web Interface provides a structured approach to creating the necessary configuration file for a Piranha cluster. This chapter describes the basic operation of the Piranha Web Interface.

9.1 Recommendations

Red Hat recommends that the Piranha Web Interface be used for any configuration file changes. The configuration file must follow strict formatting rules; otherwise, the software may experience failures. Using the Piranha Web Interface is the best method of preventing these kinds of problems.

Once a configuration file has been created or updated, the administrator must be sure to copy this file (usually via `rcp` or `scp`) to all nodes in the cluster. All nodes must use identical configuration files. Mismatched files are one of the most common cause of cluster problems.

9.2 Piranha Web Interface Requirements

As the name implies, the Piranha Web Interface requires a Web server capable of supporting CGI. Apache 1.3.*x* is a good choice. In addition, PHP version 3 (or later) is required. If your system uses RPM, you can simply install the appropriate Apache and PHP RPMs.

Otherwise, please consult the Apache and PHP homepages (http://www.apache.org/, and http://www.php.net/ respectively) for downloading and installation instructions. Once Apache and PHP are installed, you should install the `piranha`, `ipvsadm`, and `piranha-gui` RPMs [1].

The Piranha RPMs will create a new user on your system (user piranha) which is a system account with no login properties (ie, you cannot log into a machine with this

[1] Non-RPM systems should download the latest RPM tar file from http://www.rpm.org/, and use the rpm2cpio utility to extract the contents from the Piranha RPMs.

user ID). In future releases, the piranha user ID will be used to implement additional functionality.

9.2.1 Security Implications

One major security-related implication to the use of a Web-based interface is that the design promotes potentially untrusted data (which could originate from anywhere on the Internet) as input to a root-owned process. The password protection scheme is the front line of defense; though not a 100% perfect solution, it should fend off most unwanted visitors.

If you are still concerned about who has access to the Piranha Web Interface, you can further restrict access by using Apache's acl facility. This will allow only trusted hosts or networks access to the Piranha Web Interface [2].

[2] Note that this approach will only work if the systems you are clustering will not require Web services with different access restrictions from those required by the Piranha Web Interface.

9.3 A Tour of the Piranha Web Interface

Please Note

Before using the Piranha Web Interface for the first time, you must restrict access to it by creating a password. This is done by using `/usr/sbin/piranha-passwd`. First, login as root, and issue the following command:

```
/usr/sbin/piranha-passwd <password>
```

Note that the password *must* be passed as an argument to `piranha-passwd`. If you have previously installed Piranha, the previous password is preserved, and this step will not be necessary.

If you change the password for user piranha while you have an active Piranha Web Interface session in progress, you will be prompted to provide the password for again. This is due to your browser providing the old password to the Web server. Simply type in the new password when prompted.

To use the Piranha Web Interface you will minimally need a text-only Web browser. Point your browser to this URL:

```
http://localhost/piranha/
```

When your browser connects to the Piranha Web Interface, you will notice that you must login to access the cluster configuration services. Enter "piranha" in the **User ID** field, and the password you set with `piranha-passwd` in the **Password** field.

9.3.1 Hints on Using the Piranha Web Interface

While the user interface provided by the Piranha Web Interface is similar to that found on many websites, there are two hints that you should keep in mind as you use the interface to configure your cluster:

- Selecting a specific service using the radio buttons
- Saving modified information using the **ACCEPT** button

Let's look at each hint in more detail.

Selecting a Specific Service Using the Radio Buttons

Some of the screens (which are known as **panels**) have the ability to list more than one service. [3]On these panels, each service will be represented by a one-line summary, and preceded by a radio button. In order to perform an action (such as editing or deleting) on a specific service, you must *first* press that service's radio button, and *then* perform the desired action.

On those panels where this section method is used, there will be a note reminding you of this behavior.

Saving Modified Information Using the ACCEPT Button

On most panels, there is a button labeled **ACCEPT**. This button used to confirm any changes you have made to the data displayed on that panel. Note that if you change any displayed information, and then move to another panel without first pressing **ACCEPT**, your changes will be lost.

Always press the **ACCEPT** button after changing any information!

Keeping these hints in mind, let's begin our tour of the Piranha Web Interface.

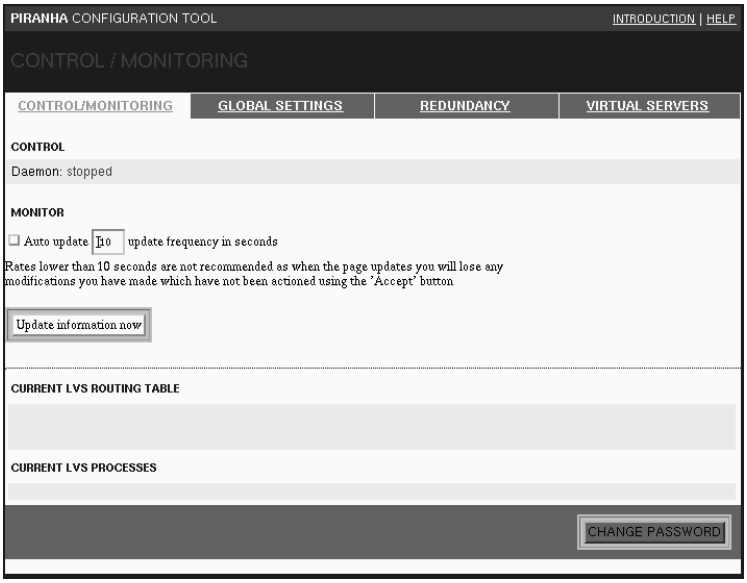
9.3.2 The CONTROL/MONITORING Panel

The first screen you will be presented with is the **CONTROL / MONITORING** panel. From this panel, it is possible to obtain information about the state of the cluster. It also has the ability to automatically update the table listings. This is useful when you're

[3] Specifically, these are the **FAILOVER** and **VIRTUAL SERVERS** panels.

behind the back of a machine tearing out cables trying to test your cluster and do not have easy access to the mouse and keyboard. To enable the auto-update feature, simply press the **Auto update** button, enter the desired update frequency in the provided text box, and press [Enter]:

Figure 9–1 The CONTROL/MONITORING Panel

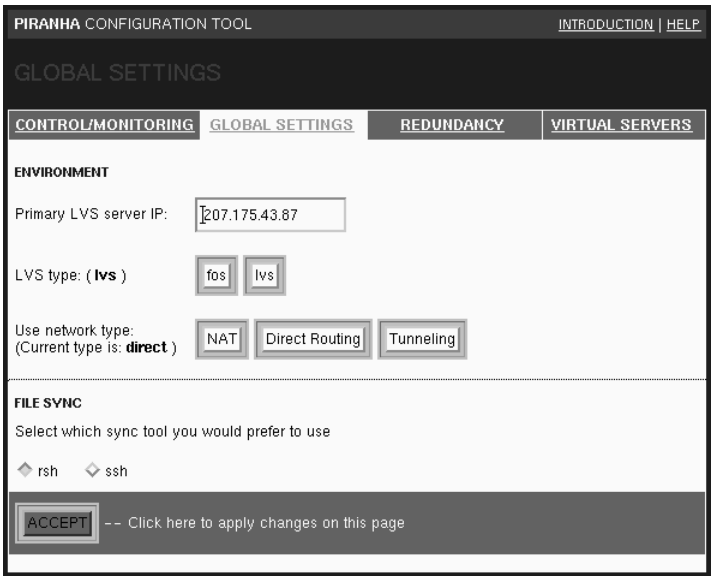


You can also manually update the contents of the **CONTROL/MONITORING** panel by pressing the **Update information now** button.

9.3.3 The GLOBAL SETTINGS Panel

If you now click on the **GLOBAL SETTINGS** tab, you will be presented with the following page:

Figure 9–2 Global Settings Panel



From this panel you can start defining the type of cluster you wish to build. Start by entering the IP address of your primary server in the **Primary LVS server IP** text field. The **LVS type** section defines the type of cluster service, and gives two choices:

- **lvs** (Linux Virtual Services)
- **fos** (Fail Over Services)

Please see Chapter 8, *Linux Virtual Server (LVS)* and Chapter 7, *Failover Services (FOS)* for more details on the differences between LVS and FOS.

If you've selected LVS, you will be prompted to select from the three types of packet forwarding offered:

- NAT (Network Address Translation)
- Direct Routing
- Tunneling

If you choose NAT, you will be presented with two more fields to complete:

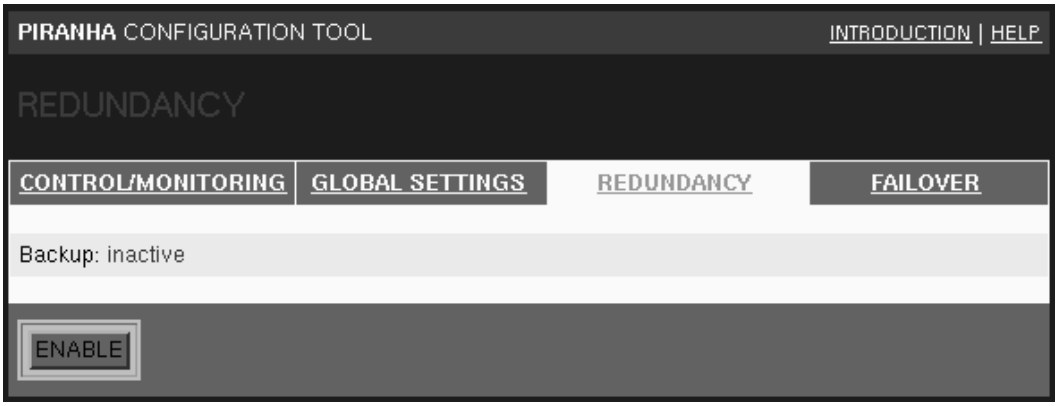
- The NAT router IP address
- The router device (eth0:2, for example)

Below the packet forwarding selection, you can select the method of file replication used to synchronize the cluster configuration files between the clustered systems. The default is to synchronize using rsh, although ssh may be used if you have it installed.

9.3.4 The REDUNDANCY Panel

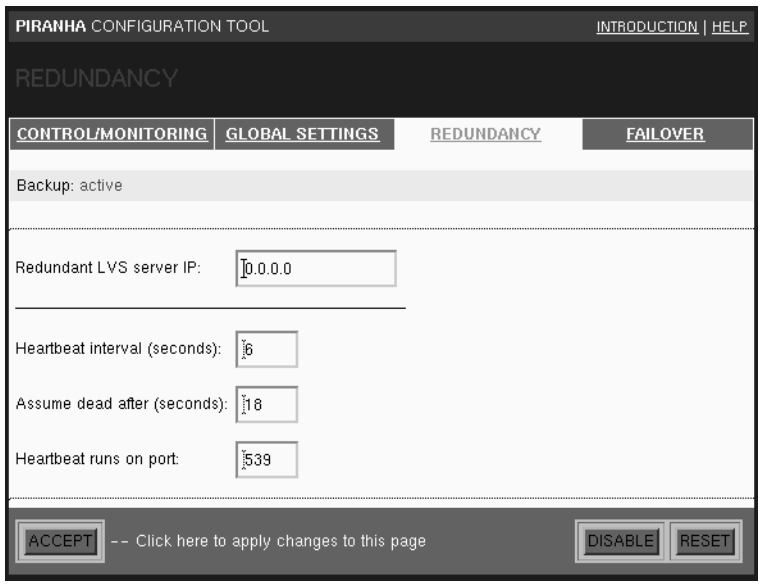
Next, click on the **REDUNDANCY** link; you will be presented with a screen displaying information on the redundant LVS server. Initially, this page will be blank because no redundant LVS server has been defined:

Figure 9–3 Redundancy Panel (initial)



To define a redundancy server, click on the **ENABLE** button and the screen will change to the following:

Figure 9–4 Redundany Panel (expanded)



Edit the data on this panel as appropriate to your setup. At a minimum, you'll need to enter the IP address of your redundant server in the **Redundant LVS server IP** text box. The remaining text boxes contain default values that are appropriate for most configurations.

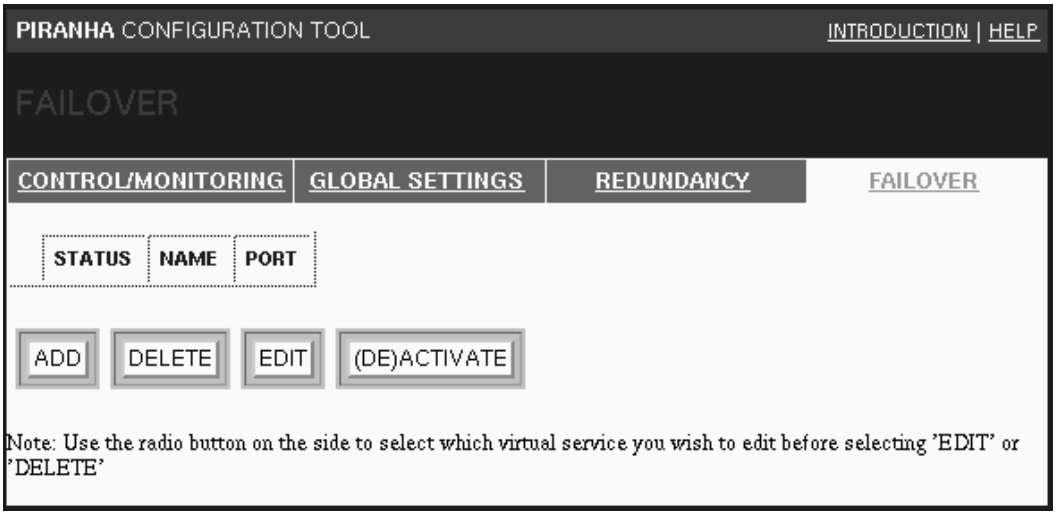
9.3.5 The Fourth Panel

The title of the fourth tab header depends on your choice of LVS service in the **LVS type** section on the **GLOBAL SETTINGS** panel. If you selected **FOS**, the fourth tab will be entitled **FAILOVER**. If, on the other hand, you selected **LVS**, the fourth tab will be entitled **VIRTUAL SERVERS.**

The FAILOVER Panel

Let's start with the **FAILOVER** panel. (If you selected LVS, please skip ahead to *The* **VIRTUAL SERVERS** *Panel* in Section 9.3.5.) Click on the **FAILOVER** tab. You will be presented with the following panel:

Figure 9–5 Failover Panel (Initial)

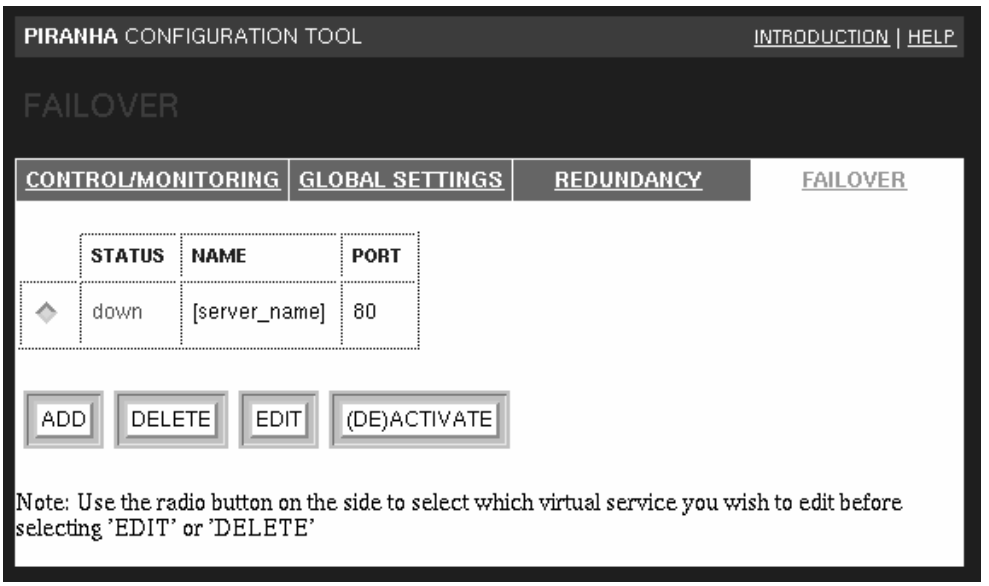


Here you'll be presented with the following buttons:

- **ADD** (Adds a blank service template)
- **DELETE** (Removes the selected service)
- **EDIT** (Makes changes to the selected service)
- **(DE)ACTIVATE** (Toggles the selected service's status)

Click on the **ADD** button to start defining a new server. The **FAILOVER** panel will now include a new server entry:

Figure 9–6 The FAILOVER Panel (Server Added)

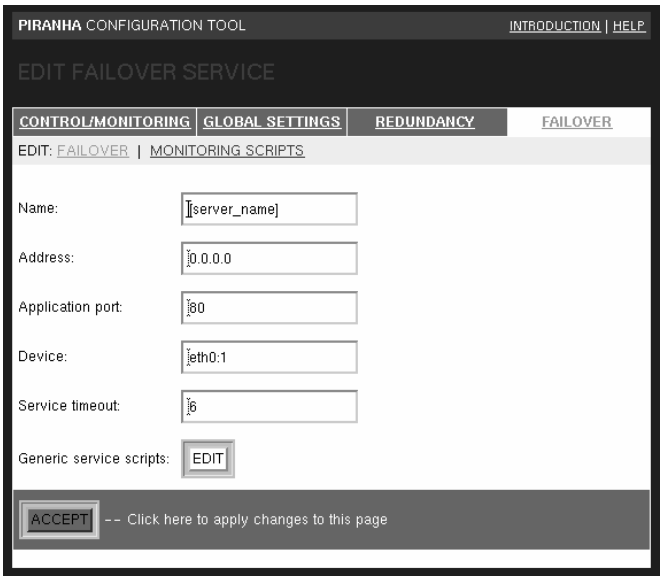


Please Note

If you add more than one server, use the radio button in front of each line to indicate which server you wish to delete, edit, or (de)activate.

By clicking **EDIT**, you will be presented with the following panel:

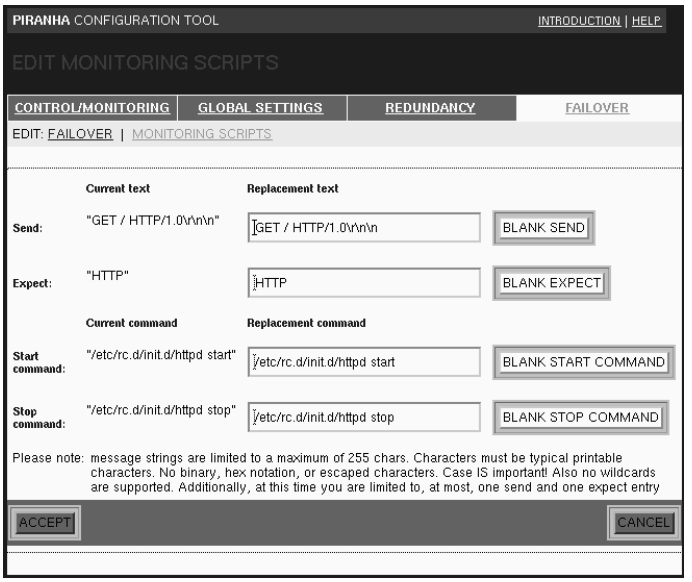
Figure 9–7 Failover Panel (Editing)



The fields will be filled in with defaults; the **Name** and **Address** fields must be modified. Modify the remaining fields as appropriate.

By pressing the **EDIT** button (after first saving any data with the **ACCEPT** button) you will be shown the following panel:

Figure 9–8 Failover Panel (script editing)

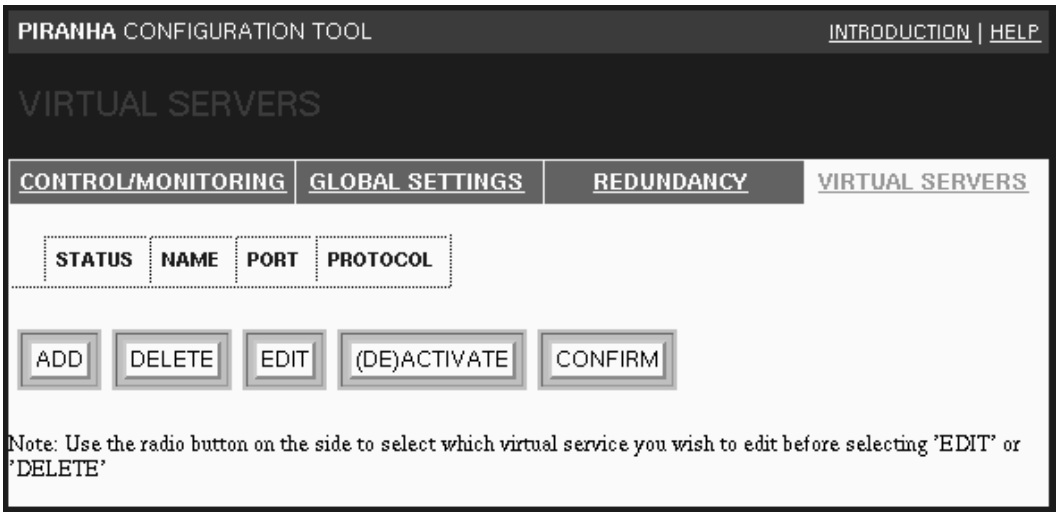


Note that the fields have a default example inserted. which you can use as-is, modify, or replace entirely.

The VIRTUAL SERVERS Panel

If, in your initial setup, you chose to use LVS services from the **GLOBAL SETTINGS** panel, the fourth tab will read **VIRTUAL SERVERS** instead of **FAILOVER**. The panel will look similar to the FOS page, except that you will notice an extra field labeled **PROTOCOL**:

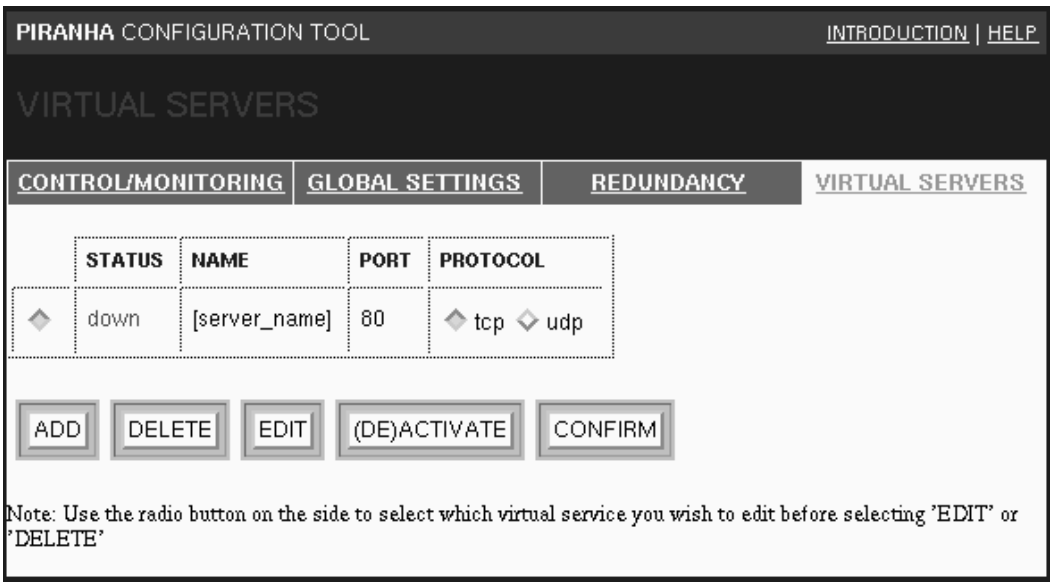
Figure 9–9 Virtual Servers Panel (Initial)



The **PROTOCOL** field makes it possible to define the server as using either the TCP or UDP protocols. A **CONFIRM** button has also been added to permit confirmation of any changes to the **PROTOCOL** field.

A new server can now be added using the **ADD** button:

Figure 9–10 Virtual Servers Panel (Server Added)

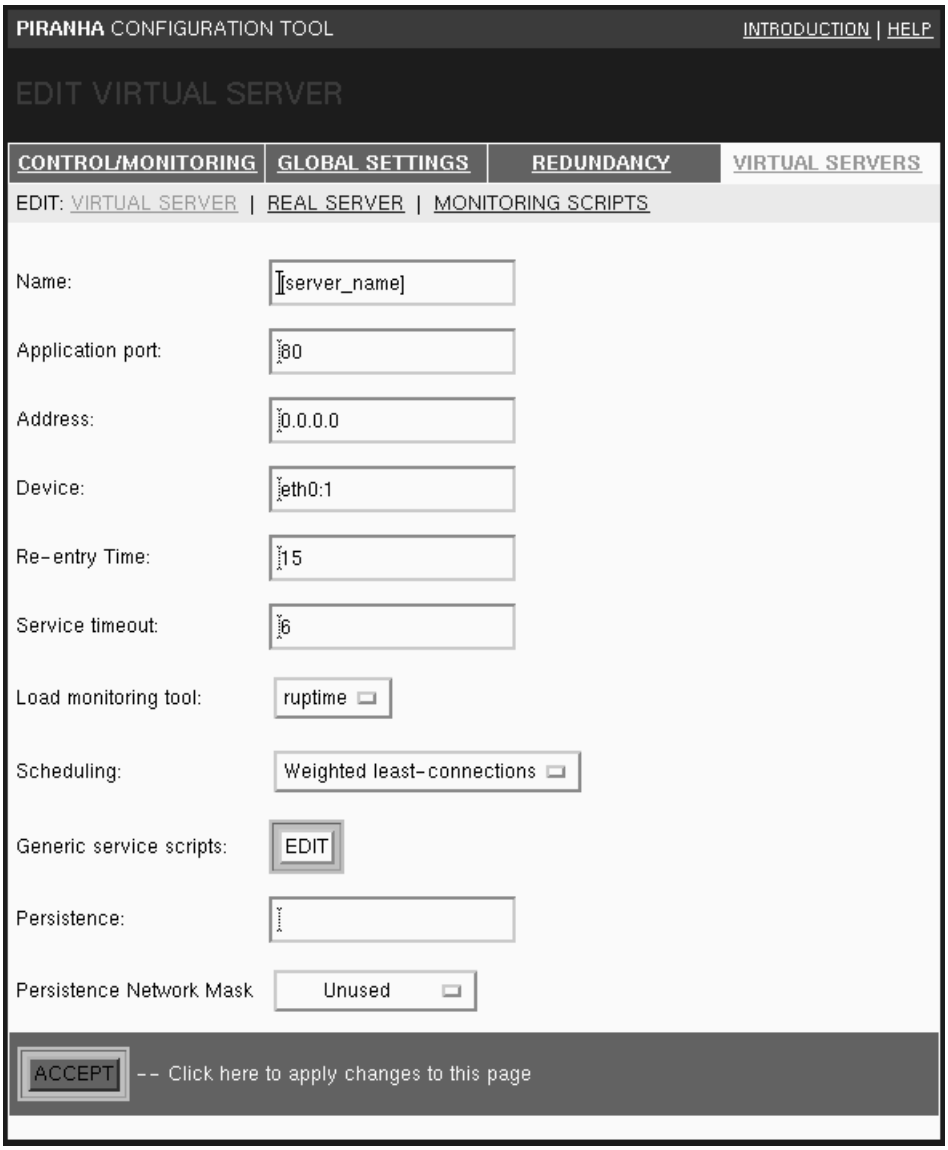


Please Note

If you add more than one server, use the radio button in front of each line to indicate which server you wish to delete, edit, (de)activate, or confirm.

Once a new server has been added and selected with the radio button, the **EDIT** button will bring you to a panel similar to the one used in FOS:

Figure 9–11 Virtual Servers Panel (Server Editing)



The fields will be filled in with defaults; the **Name** and **Address** fields must be modified. Modify the remaining fields as appropriate.

Part III Appendixes

A Additional Resources

The following are additional sources of information and assistance for Linux High Availability, LVS, and Red Hat Piranha.

A.1 Documentation

If the `piranha-docs` RPM package has been installed on your system, there are additional help and information files located in:

```
/usr /doc/piranha-docs-*/*
```

This directory includes text, html, and Postscript documents providing information on LVS configuration, architectures, and more.

Further information can also be found in the man pages for `pulse`, `nanny`, `lvs`, `fos`, `ipvsadm`, and `lvs.cf`.

A.2 Websites

http://www.redhat.com/ — The Red Hat home page. Contains links to product information, announcements, downloads, training, on-line purchasing, etc.

http://www.redhat.com/apps/community/ — The Red Hat community page. Contains online documentation, whitepapers, and links to the Piranha dedicated Web pages.

http://bugzilla.redhat.com/bugzilla — The Red Hat Bugzilla page. For searching or logging bug reports on Red Hat Linux products.

http://www.linuxvirtualserver.org/ — The community LVS project page. The central source for LVS information, documentation, and patches. This site also has a links page, with links to many high availability and clustering-related pages for Linux.

http://www.linuxdoc.org/ — The Linux Documentation Project site.

A.3 Mailing Lists

In most cases, the authors of the software components involved with high aailability Linux are active participants in the followingmailing lists:

- Linux Virtual Server (LVS) mailing list — To subscribe to the list, send a message to: lvs-users-subscribe@LinuxVirtualServer.org
- Linux High Availability Developers mailing list — To subscribe to the list, go to http://lists.tummy.com/mailman/listinfo/linux-ha-dev. The project's homepage is http://linux-ha.org/.

B A Sample /etc/lvs.cf File

The following file is a heavily-commented sample of an lvs.cf file. A copy of this file is present in the piranha and piranha-docs RPMs; you can find them in /usr/doc/<pkgname>*/sample.cf, where <pkgname> is either piranha or piranha-docs.

```
# This file is generated by the piranha GUI.  Do not hand edit.  All
# modifications created by any means other than the use of piranha
# will not be supported.
#
# This file has 3 sections. Section 1 is always required, then EITHER
# section 2 or section 3 is to be used.
#       1. LVS node/router definitions needed by the LVS system.
#       2. Virtual server definitions, including lists of real
#          servers.

#       3. Failover service definitions (for any services running on
#          the LVS primary or backup node instead of on virtual
#          servers). NOTICE: Failover services are an upcoming feature
#          of piranha and are not provided in this release.

# SECTION 1 - GLOBAL SETTINGS
#
# The LVS is a single point of failure (which is bad).  To protect
# against this machine breaking things, we should have a
# redundant/backup LVS node.
#       service:        Either "lvs" for Virtual Servers  or "fos" for
#                       Failover Services (defaults to "lvs" if
#                       missing)
#       primary:        The IP of the main LVS node/router
#       backup:         The IP of the backup LVS node/router
#       backup_active:  Set this to 1 if using a backup LVS
#                       node/router
#       heartbeat:      Use heartbeat between LVS nodes
#       keepalive:      Time between heartbeats between LVS machines.
#       deadtime:       Time w/ out response before node failure is
#                       assumed.

service = lvs
primary = 207.175.44.150
backup = 207.175.44.196
backup_active = 1
```

```
heartbeat = 1
heartbeat_port = 1050
keepalive = 6
deadtime = 18

# All nodes must have either appropriate .rhost files set up for all
# nodes in the cluster, us use some equivalent mechanism. Default it
# rsh, but you may set an alternate command (which must be equivalent
# to rsh) here (ssh is the most common).

rsh_command = rsh

# lvs server configuration environments: NAT, Direct Routing, and
# Tunneling.  NAT (Network Address Translation) is the simplist to set
# up and works well in most situations.
#
# network = direct
# network = tunnel

network = nat
nat_router = 192.168.10.100 eth1:1

# SECTION 2 - VIRTUAL SERVERS
#
# Information we need to keep track of for each virtual server is:
# scheduler:    pcc, rr, wlc, wrr (default is wlc)
# persistent:   time (in seconds) to allow a persistent service
#               connection to remain active.  If missing or set to 0,
#               persistence is turned off.
# pmask:        If persistence is enabled, this is the netmask to
#               apply.  Default is 255.255.255.255
# address:      IP address of the virtual server (required)
# active:       Simple switch if node is on or off
# port:         port number to be handled by this virtual server
#               (default is 80)
# load_monitor: Tool to check load average on real server machines.
#               Possible tools include rup, ruptime, uptime.
# timeout:      Time (in seconds) between service activity queries
# reentry:      Time (in seconds) a service must be alive before it is
#               allowed back into the virtual server's routing table
#               after leaving the table via failure.
# send:         [optional] test string to send to port
# expect:       [optional] test string to receive from port
# protocol:     tcp or udp (defaults to tcp)
#
```

```
# This is the needed information for each real server for each Virtual
# Server:
# address:      IP address of the real server.
# active:       Simple switch if node is on or off
# weight:       relative measure of server capacity

virtual server1 {
 address = 207.175.44.252 eth0:1
 active = 1
 load_monitor = uptime
 timeout = 5
 reentry = 10
 port = http
     send = "GET / HTTP/1.0\r\n\r\n"
     expect = "HTTP"
 scheduler = wlc
 persistent = 60
 pmask = 255.255.255.255
     protocol = tcp

 server Real1 {
  address = 192.168.10.2
  active = 1
  weight = 1
 }

 server Real2 {
  address = 192.168.10.3
  active = 1
  weight = 1
 }
}

virtual server2 {
 address = 207.175.44.253 eth0:1
 active = 0
 load_monitor = uptime
 timeout = 5
 reentry = 10
 port = 21
     send = "\n"

 server Real1 {
  address = 192.168.10.2
  active = 1
```

```
 }

 server Real2 {
  address = 192.168.10.3
  active = 1
 }
}

# SECTION 3 - FAILOVER SERVICES
#
# LVS node Service failover. This section applies only to services
# running on the primary and backup LVS nodes (instead of being part
# of a virtual server setup). You cannot currently use these services
# and virtual servers in the same setup, and you must have at least a
# 2 node cluster (a primary and backup) in order to use these failover
# services. All nodes must be identically configured Linux systems.
#

# Failover services provide the most basic form of fault recovery. If
# any of the services on the active node fail, all of the services
# will be shutdown and restarted on a backup node. Services defined
# here will automatically be started & stopped by LVS, so a backup
# node is considered a "warm" standby. This is due to a technical
# restriction that a service can only be operational on one node at a
# time, otherwise it may fail to bind to a virtual IP address that
# does not yet exist on that system or cause a networking conflict
# with the active service. The commands provided for "start_cmd" and
# "stop_cmd" must work the same for all nodes. Multiple services can
# be defined.
#
# Information here is similar in meaning and format to the virtual
# server section. Failover Services and Virtual Servers cannot both be
# used on a running system, so the "service = xxx" setting in the
# first section of this file indicates which to use when starting the
# cluster.

failover web1 {
     active = 1
     address = 207.175.44.242 eth0:1
     port = 1010
     send = "GET / HTTP/1.0\r\n\r\n"
     expect = "HTTP"
     timeout = 10
     start_cmd = "/etc/rc.d/init.d/httpd start"
     stop_cmd = "/etc/rc.d/init.d/httpd stop"
```

```
}

failover ftp {
     active = 0
     address = 207.175.44.252 eth0:1
     port = 21
     send = "\n"
     timeout = 10
     start_cmd = "/etc/rc.d/init.d/inet start"
     stop_cmd = "/etc/rc.d/init.d/inet stop"
}
```

C Getting Technical Support

Please Note

Red Hat High Availability Server 1.0 includes additional support offerings. Please refer to the terms and conditions document present in your Red Hat High Availability Server box.

C.1 Remember to Sign Up

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 about only those subjects that interest you.
- 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.

C.2 An Overview of Red Hat Support

Red Hat provides installation assistance for Official Red Hat Linux boxed set products and covers installation on a single computer. This assistance is intended to help customers successfully install Red Hat Linux. Assistance with installation is offered via telephone and the Web.

Red Hat Support will attempt to answer any questions you may have before the installation process is initiated. This includes the following:

- Hardware compatibility questions
- Basic hard drive partitioning strategies

Red Hat, Inc. Support can also provide assistance during the installation process:

- Getting any supported hardware recognized by the Red Hat Linux operating system
- Assistance with drive partitioning
- Configuring Red Hat Linux and up to one other operating system (on Intel platforms only) to dual-boot using the Linux boot loader LILO. Please note that third party boot loaders and partitioning software are not supported.

We can also help you with basic post-installation tasks, such as:

- Successfully configuring the X Window System using XF86Setup or Xconfigurator
- Configuring a local parallel port printer to print text
- Configuring a mouse

Our installation assistance service is designed to get you up and running with Red Hat Linux as quickly and as easily as possible. However, there are many other things that you may want to do with your Red Hat Linux system (from compiling a custom kernel to setting up a Web server) which are not covered.

For assistance with these tasks, there is a wealth of on-line information available in the form of HOWTO documents, Linux-related websites, and commercial publications. The Red Hat Linux operating system includes the various Linux HOWTO documents

on the installation CD in the `/doc/HOWTO` directory as plain text files that can easily be read from within Red Hat Linux and other operating systems.

A large number of Linux-related websites are available. The best starting point for finding information on Red Hat Linux is the Red Hat, Inc. website at:

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

Many Linux-related books are available. If you're new to Linux, a book that covers Linux basics will be invaluable. We can recommend several titles: *Using Linux*, by Bill Ball; *Linux Clearly Explained*, by Bryan Pfaffenberger; *Linux for Dummies*, by Jon "maddog" Hall; and *A Practical Guide to Linux*, by Mark G. Sobell.

Red Hat also offers various incident-based support plans to assist with configuration issues and tasks that are not covered by installation assistance. Please see the Red Hat Support website for more information. The Red Hat technical support website is located at the following URL:

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

C.3 Scope of Red Hat Support

Red Hat, Inc. can only provide installation assistance to customers who have purchased an Official Red Hat Linux boxed set. If you have obtained Linux from any other company, you must contact that company for support. Examples of such companies are as follows:

- Macmillan
- Sams/Que
- Linux Systems Labs (LSL)
- Mandrake
- CheapBytes

Additionally, Red Hat Linux obtained via any of the following methods does not qualify for support from Red Hat:

- Red Hat Linux PowerTools Archive

- Downloaded via FTP on the Internet
- Included in a package such as Motif or Applixware
- Copied or installed from another user's CD

C.4 The Red Hat Support System

As of October 1999, Red Hat, Inc. has implemented a new technical support system. If you signed up for technical support in the past with Red Hat, it may be necessary for you to sign up again. The new system will implement a unified login and password that will work across the entire Red Hat website. The support system will also automatically route and track service requests.

If you haven't signed up yet, then you should. Instructions for how to sign up are provided next, in Section C.5, *How to Get Technical Support*.

C.5 How to Get Technical Support

In order to receive technical support for your Official Red Hat product, you first have to sign up.

Every Official Red Hat product comes with a Personal Product Identification code: a 16-character alphanumeric string. The Personal Product ID for Red Hat Linux 6.2 is located on a red and white card that can be found inside the box. Your Personal Product ID is on a perforated card that you can punch out and keep in a safe place. You need this code, so don't lose the card!

Please Note

Do not throw away the card with your Personal Product ID. You need the Personal Product ID to get technical support. If you lose the certificate, you may not be able to receive support.

The Personal Product ID is the code that will enable your technical support and any other benefits or services that you purchased from Red Hat, depending upon which

Red Hat product you purchased. The Personal Product ID may also enable priority FTP access, depending on the product that you purchased, for a limited amount of time.

C.5.1 Signing up for Technical Support

You'll need to:

1. Create a customer profile at http://www.redhat.com/now. You may have already completed this step; if you have, continue to the next step. If you do not already have a customer profile on the Red Hat website, please create a new one.
2. With your login name and password, please login at the Red Hat Support website at http://www.redhat.com/support.
3. Update your contact information if necessary.

Please Note

If your e-mail address is not correct, communications regarding your technical support requests CANNOT be delivered to you, and you will not be able to retrieve your login and password by e-mail. Be sure that you give us your correct e-mail address.

If you're worried about your privacy, please see Red Hat's privacy statement at http://www.redhat.com/legal/privacy_statement.html.

4. Add a product to your profile. Please enter the following information:
   - The Personal Product ID for the boxed set product
   - A description of the hardware on which the Red Hat Linux product will be installed
   - The Support Certificate Number or Entitlement Number if the product is a contract
5. Set your customer preferences.

6. Answer the optional customer questionnaire.
7. Submit the form.

If the previous steps were completed successfully, you can now login at http://www.redhat.com/support and open a new technical service request. However, you must still use your Personal Product ID in order to obtain technical support via telephone (if the product you purchased came with phone support). Please do not lose your Personal Product ID, or you might not be able to receive support.

C.6 Questions for Technical Support

Technical support is both a science and a mystical art form. In most cases, support technicians must rely on customer observations and communications with the customer in order to diagnose and solve the problem. Therefore, it is extremely important that you are as detailed and clear as possible when you state your questions and report your problems. Examples of what you should include are:

- Symptoms of the problem (for example: "Linux is not able to access my CD-ROM drive. When it tries, I get timeout errors.")
- When the problem began (for example: "My system was working fine until yesterday, when a lightning storm hit my area.")
- Any changes you made to your system (for example: "I added a new hard drive and used ‘Partition Wizzo’ to add Linux partitions.")
- Other information that may be relevant to your situation, such as the installation method (CD-ROM, NFS, HTTP)

C.6.1 How to Send Support Questions

Please login at http://www.redhat.com/support and open a new service request, or call the phone number for support. If your product came with phone support, or you’ve purchased a phone support contract, the phone number you’ll need to call will be provided to you during the sign up process.

C.7 Support Frequently Asked Questions (FAQ)

C.7.1 Q: E-Mail Messages to support@redhat.com Bounce

I send e-mail to support@redhat.com but my messages bounce back to me. What is the problem?

C.7.2 A: support@redhat.com Is Not Used at This Time

To better serve our customers, Red Hat is re-engineering our e-mail support process. At this time, the support@redhat.com address is not functional. In the meantime, please use support via the Web or by telephone.

C.7.3 Q: System Won't Allow Login

I know that I have already signed up, but the system will not let me log in.

C.7.4 A: Old Logins and Passwords Won't Work

You could be trying to use an old login and password, or simply mistyping your login or password.

D Installing Without Partitioning

This chapter explains how to install Red Hat Linux 6.2 without creating Linux partitions on your system.

Please Note

Although this is a great way to explore the world of Red Hat Linux without having to put Linux partitions on your system, please note that you will still have to perform a full Red Hat Linux installation as outlined in this manual.

Please Note

You must currently have a formatted DOS (FAT) filesystem in order to perform this type of installation. Users who have Win95/98 should have no problems with this type of installation. Users who have NTFS partitions (such as those using Windows NT) will have to create and format a DOS (FAT) filesystem before this installation can be performed. This installation will not work unless the DOS (FAT) filesystem has been formatted prior starting the Red Hat Linux installation.

D.1 The Ups and Downs of a Partitionless Installation

There may be reasons why you might want to perform a partitionless installation, but there also are some drawbacks (depending on how you look at them).

Here we will cover the basics of what will happen, both during an installation and as a result of this type of installation, and how your system will be affected.

Basic Installation

You will perform a basic Red Hat Linux installation. However, instead of adding Linux partitions to your system, you will edit an existing, formatted DOS (FAT) partition (that must have enough free space) to be named root (/).

Unlike a typical Red Hat Linux installation, you will not need to format any partitions, since you will not be adding any partitions to your system.

LILO (*LI*nux *LO*ader) and Boot Disk

In a partitionless installation, you will not configure LILO (the LInux LOader). In a typical installation, you are able to choose where you would like LILO to be installed — either on the master boot record (MBR) or on the first sector of your root partition — or you can choose not to install LILO at all.

You must create a boot disk in order to access Red Hat Linux with a partitionless installation, and you will be prompted to create a boot disk at the end of the installation.

Performance Implications

Red Hat Linux will perform slower than it would if it had its own dedicated partitions. However, for those of you unconcerned with speed, a partitionless installation is a great way of seeing what Red Hat Linux has to offer without having to deal with partitioning your system.

D.2 Performing a Partitionless Installation

If you have a DOS (FAT) filesystem you must first make sure you have a DOS (FAT) partition with enough free space to dedicate to this installation.

D.2.1 How Much Space Do I Need?

Like a typical installation, you will need to have enough available space in order to install Red Hat Linux on your system. To give you an idea, below is a list of installation methods and their *minimum* space requirements.

For more information about these installation classes, see Section 2.1.7, *Step 7 - Which Installation Type is Best For You?*.

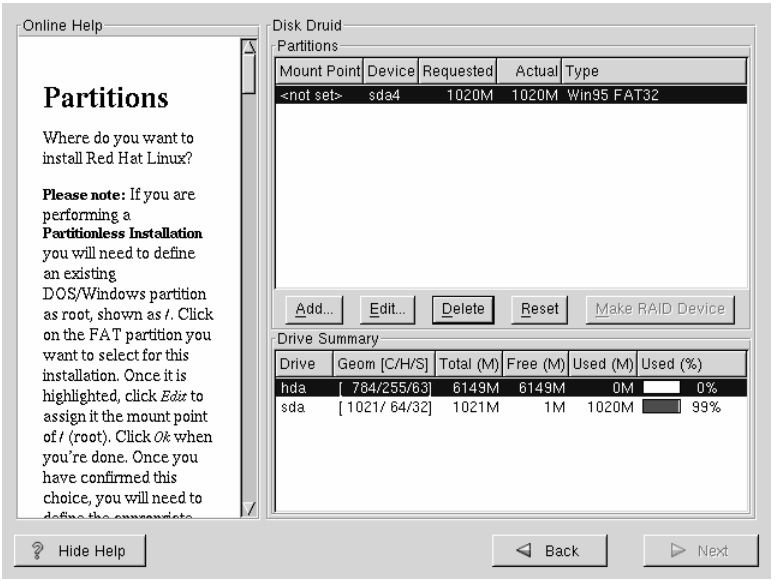
- Cluster Server - 1.7GB

- Custom (choosing *Everything*) - 1.7GB

D.2.2 Using Disk Druid

Since you will not be adding partitions or creating new partitions, there is relatively little that you actually need to do with Disk Druid (a GUI partitioning tool).

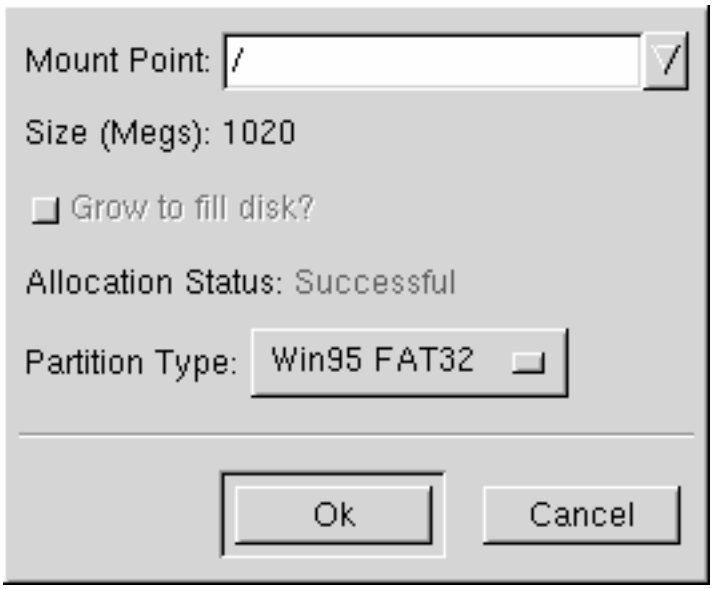
Figure D–1 Choosing DOS (FAT) Partition to Define as /



What you should see when Disk Druid's main screen appears is a list of your DOS (FAT) partitions (see Figure D–1, *Choosing DOS (FAT) Partition to Define as /*). Choose a DOS (FAT) partition with enough available free space to install your choice of installation classes. Highlight the partition by clicking on it with your mouse or by using the [Tab], [Up] and [Down] keys.

Once the desired partition is highlighted, choose **Edit**. A new window will appear allowing you to name this partition (see Figure D–2, *Editing a DOS (FAT) Partition*). In the **mount point** field, label this partition as / (known as root) and click **Enter**.

Figure D–2 Editing a DOS (FAT) Partition



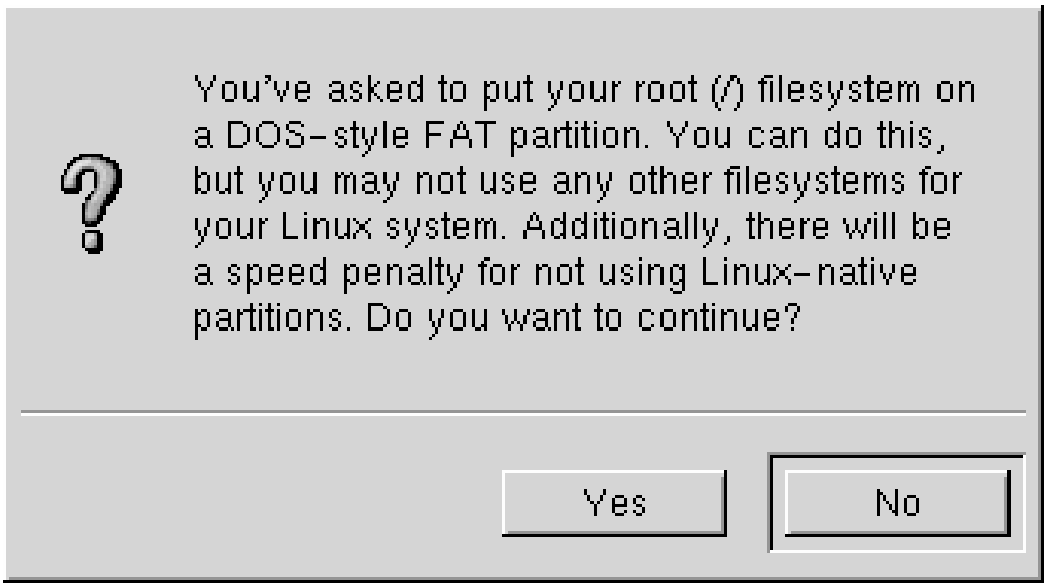
A confirmation window (see Figure D–3, *Confirmation*) will appear next asking you to confirm that you do want to continue with this installation. It also explains that you cannot have any Linux partitions on your system other than the / labeled DOS partition that you have just created. Click **Yes** to continue.

Next, you will be able to determine the root filesystem size and the swap size of this / partition.

The installation program will determine the maximum size for the root filesystem (Figure D–4, *Configuration of Filesystem*). You can make the root filesystem anything you would like, as long as it does not exceed the maximum size recommendation.

The size you create for the root filesystem is the amount of disk space available for the entire filesystem (this means that you need to keep in mind the size of the installation class as well as allow you space to write and save data to).

Figure D–3 Confirmation

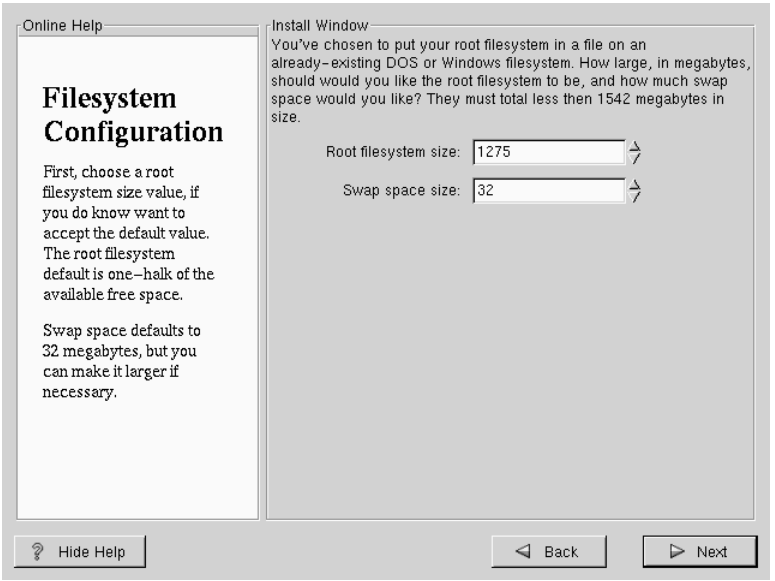


Swap space acts like virtual memory. In other words, data is written to a swap partition when there is not enough RAM to store the data your system is processing. The installation program will set swap to 32MB as a default. You can choose to increase the swap size if desired, but there is no need to create a swap space larger than 256MB.

From here, you can continue following the main installation chapter (see Section 4.7, *Network Configuration*) for further installation instructions. The only difference you will see from this point is a screen prompting you to create a boot disk. Once you make the boot disk and follow the other instructions, your installation will be complete.

To access Red Hat Linux, make sure the boot disk that you created during the installation is in your floppy drive. When you reboot your system it will enter into Red Hat Linux rather than your other OS. To access your other OS, remove the boot disk and reboot your system.

Figure D–4 Configuration of Filesystem



D.2.3 How to Remove a Partitionless Installation From Your System

To remove this partitionless installation, you will need to delete the following files:

```
redhat.img
rh-swap.img
```

These files can be found in the partition's root directory (known as \ under Dos/Windows.)

Once these files have been removed, Red Hat Linux will no longer boot on your system. Your system will return to its previous state and you will be able to access the space used by Red Hat Linux as you normally would.

E Removing Red Hat Linux

To uninstall Red Hat Linux from your system, you will need to remove the LILO information from your Master Boot Record (MBR).

There are several methods to removing LILO from the master boot record of the machine. Inside of Linux, you can replace the MBR with an earlier saved version of the MBR using the /sbin/lilo command:

```
/sbin/lilo -u
```

In DOS, NT, and Windows 95 you can use fdisk to create a new MBR with the "undocumented" flag `/mbr`. This will ONLY rewrite the MBR to boot the primary DOS partition. The command should look like:

```
fdisk /mbr
```

If you need to remove Linux from a hard drive, and have attempted to do this with the default DOS fdisk, you will experience the "Partitions exist but they don't exist" problem. The best way to remove non-DOS partitions is with a tool that understands partitions other than DOS.

You can perform this with the installation floppy by typing "linux expert" (without the quotes) at the `boot:` prompt.

```
boot:linux expert
```

Select install (versus upgrade) and when it comes to partitioning the drive, choose fdisk. In fdisk type [p] to print out the partition numbers, and remove the Linux partitions with the [d] command. When you're satisfied with the changes you have made, you can quit with a [w] and the changes will be saved to disk. If you deleted too much, type [q] and no changes will occur.

Once you have removed the Linux partitions, you can reboot your computer by pressing [Control-]-[Alt-]-[Delete] instead of continuing with the install.

Index

D

E

F

G

H

J

K

L

M

N

O

P

R

S

T

U

V

W

X