Fedora 12

Deployment Guide

Deployment, configuration and administration of Fedora 12

Douglas Silas

John Ha

David O'Brien

Michael Hideo

Don Domingo

Michael Behm

Fedora 12 Deployment Guide
Deployment, configuration and administration of Fedora 12
Edition 0

Author	Douglas Silas	dhensley@redhat.com
Author	John Ha	
Author	David O'Brien	
Author	Michael Hideo	
Author	Don Domingo	
Author	Michael Behm	
	Jeffrey Fearn	
	Garrett LeSage	
	Andrew Fitzsimon	
	Michael Behm	
	Sandra Moore	
	Edward Bailey	
	Karsten Wade	
	Mark Johnson	
	Andrius Benokraitis	
	Lucy Ringland	

The Deployment Guide documents relevant information regarding the deployment, configuration and administration of Fedora 12.

Preface

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the *Liberation Fonts*[1] set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later includes the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

`Mono-spaced Bold`

Used to highlight system input, including shell commands, file names and paths. Also used to highlight key caps and key-combinations. For example:

> To see the contents of the file **`my_next_bestselling_novel`** in your current working directory, enter the **`cat my_next_bestselling_novel`** command at the shell prompt and press **`Enter`** to execute the command.

The above includes a file name, a shell command and a key cap, all presented in Mono-spaced Bold and all distinguishable thanks to context.

Key-combinations can be distinguished from key caps by the hyphen connecting each part of a key-combination. For example:

> Press **`Enter`** to execute the command.
>
> Press **`Ctrl`**+**`Alt`**+**`F1`** to switch to the first virtual terminal. Press **`Ctrl`**+**`Alt`**+**`F7`** to return to your X-Windows session.

The first sentence highlights the particular key cap to press. The second highlights two sets of three key caps, each set pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **`Mono-spaced Bold`**. For example:

> File-related classes include **`filesystem`** for file systems, **`file`** for files, and **`dir`** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialogue box text; labelled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

[1] https://fedorahosted.org/liberation-fonts/

> Choose **System > Preferences > Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, click the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).
>
> To insert a special character into a **gedit** file, choose **Applications > Accessories > Character Map** from the main menu bar. Next, choose **Search > Find…** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit > Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in Proportional Bold and all distinguishable by context.

Note the **>** shorthand used to indicate traversal through a menu and its sub-menus. This is to avoid the difficult-to-follow 'Select **Mouse** from the **Preferences** sub-menu in the **System** menu of the main menu bar' approach.

`Mono-spaced Bold Italic` or ***`Proportional Bold Italic`***

Whether Mono-spaced Bold or Proportional Bold, the addition of Italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

> To connect to a remote machine using ssh, type **`ssh`** ***`username@domain.name`*** at a shell prompt. If the remote machine is **`example.com`** and your username on that machine is john, type **`ssh john@example.com`**.
>
> The **`mount -o remount`** ***`file-system`*** command remounts the named file system. For example, to remount the **`/home`** file system, the command is **`mount -o remount /home`**.
>
> To see the version of a currently installed package, use the **`rpm -q`** ***`package`*** command. It will return a result as follows: ***`package-version-release`***.

Note the words in bold italics above — username, domain.name, file-system, package, version and release. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

> When the Apache HTTP Server accepts requests, it dispatches child processes or threads to handle them. This group of child processes or threads is known as a *server-pool*. Under Apache HTTP Server 2.0, the responsibility for creating and maintaining these server-pools has been abstracted to a group of modules called *Multi-Processing Modules* (*MPMs*). Unlike other modules, only one module from the MPM group can be loaded by the Apache HTTP Server.

1.2. Pull-quote Conventions

Two, commonly multi-line, data types are set off visually from the surrounding text.

Output sent to a terminal is set in `Mono-spaced Roman` and presented thus:

```
books         Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests   Desktop1  downloads      images  notes  scripts  svgs
```

Source-code listings are also set in `Mono-spaced Roman` but are presented and highlighted as follows:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }

}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.

Note

A note is a tip or shortcut or alternative approach to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.

Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring Important boxes won't cause data loss but may cause irritation and frustration.

Warning

A Warning should not be ignored. Ignoring warnings will most likely cause data loss.

2. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla: *http://bugzilla.redhat.com/bugzilla/* against the product **Fedora Documentation.**

When submitting a bug report, be sure to mention the manual's identifier: *Deployment_Guide*

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

3. Acknowledgements

Certain portions of this text first appeared in the *Deployment Guide*, copyright © 2007 Red Hat, Inc., available at *http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.4/html/Deployment_Guide/index.html*.

Introduction

Welcome to the *Fedora Deployment Guide*.

The Fedora Deployment Guide contains information on how to customize your Fedora system to fit your needs. If you are looking for a comprehensive, task-oriented guide for configuring and customizing your system, this is the manual for you.

This manual discusses many intermediate topics such as the following:

- Setting up a network interface card (NIC)
- Configuring a Virtual Private Network (VPN)
- Configuring Samba shares
- Managing your software with RPM
- Determining information about your system
- Upgrading your kernel

This manual is divided into the following main categories:

- File systems
- Package management
- Network-related configuration
- System configuration
- System monitoring
- Kernel and Driver Configuration
- Security and Authentication
- Red Hat Training and Certification

This guide assumes you have a basic understanding of your Fedora system. If you need help installing Fedora, refer to the *Fedora Installation Guide*.

Part I. Package Management

All software on a Fedora system is divided into RPM packages, which can be installed, upgraded, or removed. This part describes how to manage packages on Fedora using the **Yum** and **RPM** package managers and the **PackageKit** suite of graphical package management tools.

Chapter 1.

Yum

Yum is the Fedora package manager that is able to query for information about packages, fetch packages from repositories, install and uninstall packages using automatic dependency resolution, and update an entire system to the latest available packages. **Yum** performs automatic dependency resolution on packages you are updating, installing or removing, and thus is able to automatically determine, fetch and install all available dependent packages. **Yum** can be configured with new, additional repositories, or *package sources*, and also provides many plugins which enhance and extend its capabilities. **Yum** is able to perform many of the same tasks that **RPM** can; additionally, many of the command line options are similar. **Yum** enables easy and simple package management on a single machine or on groups of them.

Secure Package Management with GPG-Signed Packages

Yum provides secure package management by enabling GPG (Gnu Privacy Guard; also known as GnuPG) signature verification on GPG-signed packages to be turned on for all package repositories (i.e. package sources), or for individual repositories. When signature verification is enabled, **Yum** will refuse to install any packages not GPG-signed with the correct key for that repository. This means that you can trust that the **RPM** packages you download and install on your system are from a trusted source, such as the Fedora Project, and were not modified during transfer. Refer to *Section 1.3, “Configuring Yum and Yum Repositories”* for details on enabling signature-checking with **Yum**, or *Section 3.3, “Checking a Package's Signature”* for information on working with and verifying GPG-signed **RPM** packages in general.

Yum also enables you to easily set up your own repositories of **RPM** packages for download and installation on other machines.

Learning **Yum** is a worthwhile investment because it is often the fastest way to perform system administration tasks, and it provides capabilities beyond those provided by the **PackageKit** graphical package management tools. Refer to *Chapter 2, PackageKit* for details on using **PackageKit**.

1.1. Checking For and Updating Packages

1.1.1. Checking For Updates

You can use the **`yum check-update`** command to see which installed packages on your system have updates available.

Note: Yum and Superuser Privileges

You must have superuser privileges in order to use **`yum`** to install, update or remove packages on your system. All examples in this chapter assume that you have already obtained superuser privileges by using either the **su** or **sudo** command.

```
~]# yum check-update
Loaded plugins: presto, refresh-packagekit, security
PackageKit.x86_64                        0.5.3-0.1.20090915git.fc12  fedora
```

```
PackageKit-glib.x86_64             0.5.3-0.1.20090915git.fc12  fedora
PackageKit-yum.x86_64              0.5.3-0.1.20090915git.fc12  fedora
PackageKit-yum-plugin.x86_64       0.5.3-0.1.20090915git.fc12  fedora
glibc.x86_64                       2.10.90-22                  fedora
glibc-common.x86_64                2.10.90-22                  fedora
kernel.x86_64                      2.6.31-14.fc12              fedora
kernel-firmware.noarch             2.6.31-14.fc12              fedora
rpm.x86_64                         4.7.1-5.fc12                fedora
rpm-libs.x86_64                    4.7.1-5.fc12                fedora
rpm-python.x86_64                  4.7.1-5.fc12                fedora
yum.noarch                         3.2.24-4.fc12               fedora
```

Twelve packages are listed as having updates available. The first package in the list is **PackageKit**, the graphical package manager. The first line of the above output tells us:

- `PackageKit` — the name of the package
- `x86_64` — the CPU architecture the package was built for
- `0.5.3-0.1.20090915git.fc12` — the version of the updated package to be installed
- `fedora` — the repository in which the updated package is located

The output also shows us that we can update the kernel (the kernel package), **Yum** and **RPM** themselves (the **`yum`** and **`rpm`** packages), as well as their dependencies (such as the `kernel-firmware`, `rpm-libs` and `rpm-python` packages), all using **`yum`**.

1.1.2. Updating Packages

You can choose to update a single package, multiple packages, or all packages at once. If any dependencies of the package (or packages) you update have updates available themselves, then they are updated too. To update a single package, enter **`yum update <package_name>`**:

```

~]# yum update glibc
Loaded plugins: presto, refresh-packagekit, security
Setting up Install Process
Resolving Dependencies
--> Running transaction check
--> Processing Dependency: glibc = 2.10.90-21 for package: glibc-
common-2.10.90-21.x86_64
---> Package glibc.x86_64 0:2.10.90-22 set to be updated
--> Running transaction check
---> Package glibc-common.x86_64 0:2.10.90-22 set to be updated
--> Finished Dependency Resolution
Dependencies Resolved
======================================================================
 Package            Arch         Version          Repository     Size
======================================================================
Updating:
 glibc              x86_64       2.10.90-22       fedora       2.7 M
Updating for dependencies:
 glibc-common       x86_64       2.10.90-22       fedora       6.0 M
Transaction Summary
======================================================================
Install       0 Package(s)
Upgrade       2 Package(s)
Total download size: 8.7 M
Is this ok [y/N]:
```

This output contains two further items of interest:

1. `Loaded plugins: presto, refresh-packagekit, security` — **yum** always informs you which **Yum** plugins are installed and enabled. Here, **yum** is using the **presto**, **refresh-packagekit** and **security** plugins. Refer to *Section 1.4, “Yum Plugins”* for general information on **Yum** plugins, or to *Section 1.4.3, “Plugin Descriptions”* for descriptions of specific plugins.

2. `kernel.x86_64` — you can download and install new kernels safely with **yum**.

   > **Important: Updating and Installing Kernels with Yum**
   >
   > **Yum** always **install**s a new kernel in the same sense that **RPM***installs* a new kernel when you use the command **`rpm -i kernel`**. Therefore, you do not need to worry about the distinction between *installing* and *upgrading* a kernel package when you use **yum**: it will do the right thing, regardless of whether you are using the **`yum update`** or **`yum install`** command.
   >
   > When using **RPM**, on the other hand, it is important to use the **`rpm -i kernel`** command (which installs a new kernel) instead of **`rpm -u kernel`** (which *replaces* the current kernel). Refer to *Section 3.2.2, “Installing”* for more information on installing/updating kernels with **RPM**.

3. **yum** presents the update information and then prompts you as to whether you want it to perform the update; **yum** runs interactively by default. If you already know which transactions **yum** plans to perform, you can use the `-y` option to automatically answer **yes** to any questions **yum** may ask (in which case it runs non-interactively). However, you should always examine which changes **yum** plans to make to the system so that you can easily troubleshoot any problems that might arise.

   If a transaction does go awry, you can view **Yum**'s log of transactions by entering **`cat /var/log/yum.log`** at the shell prompt. The most recent transactions are listed at the end of the log file.

Updating All Packages and Their Dependencies

To update all packages and their dependencies, simply enter **`yum update`** (without any arguments):

```
~]# yum update
```

Example 1.1. Updating all packages at once

1.1.3. Updating Security-Related Packages

Discovering which packages have security updates available and then updating those packages quickly and easily is important. **Yum** provides the **security** plugin for this purpose. The **security** plugin extends the **yum** command with a set of highly-useful security-centric commands, subcommands and options. Refer to *Section 1.4.3, “security (yum-plugin-security)”* for specific information.

1.1.4. Preserving Configuration File Changes

You will inevitably make changes to the configuration files installed by packages as you use your Fedora system. **RPM**, which **Yum** uses to perform changes to the system, provides a mechanism for ensuring their integrity. Refer to *Section 3.2.4, “Upgrading”* for details on how to manage changes to configuration files across package upgrades.

1.2. Packages and Package Groups

1.2.1. Searching, Listing and Displaying Package Information

You can search all **RPM** package names, descriptions and summaries by using the **yum search *<term>* [more_terms]** command. **yum** displays the list of matches for each term:

```
~]# yum search meld kompare
Loaded plugins: presto, refresh-packagekit, security
=============================== Matched: kompare
 ===============================
kdesdk.x86_64 : The KDE Software Development Kit (SDK)
komparator.x86_64 : Kompare and merge two folders
================================ Matched: meld
 =================================
meld.noarch : Visual diff and merge tool
python-meld3.x86_64 : An HTML/XML templating system for Python
```

yum search is useful for searching for packages you do not know the name of, but for which you know a related term.

Listing Packages

yum list and related commands provide information about packages, package groups, and repositories.

Tip: Filtering Results with Glob Expressions

All of **Yum**'s various list commands allow you to filter the results by appending one or more *glob expressions* as arguments. Glob expressions consist of the wildcard characters **`*`** (which expands to match any character multiple times) and **?** (which expands to match any one character). Be careful to escape both of these glob characters when passing them as arguments to a **yum** command. If you do not, the bash shell will interpret the glob expressions as *pathname expansions*, and potentially pass all files in the current directory that match the globs to **yum**, which is not what you want. Instead, you want to pass the glob expressions themselves to **yum**, which you can do by either:

- escaping the wildcard characters
- double-quoting or single-quoting the entire glob expression.

The following examples show both methods:

```
~]# yum list available gimp\*plugin\*
Loaded plugins: presto, refresh-packagekit, security
Available Packages
gimp-fourier-plugin.x86_64       0.3.2-3.fc11        fedora
gimp-lqr-plugin.x86_64           0.6.1-2.fc11        updates
```

Example 1.2. Filtering results using a single glob expression with two escaped wildcard characters

```
~]# yum list installed "krb?-*"
Loaded plugins: presto, refresh-packagekit, security
Installed Packages
krb5-auth-dialog.x86_64         0.12-2.fc12         @fedora
krb5-libs.x86_64                1.7-8.fc12          @fedora
krb5-workstation.x86_64         1.7-8.fc12          @fedora
```

Example 1.3. Filtering results using a double-quoted glob expression:

- **`yum list <`*`glob_expr`*`> [more_glob_exprs]`** — List information on installed and available packages matching all glob expressions.

```
~]# yum list abrt-addon\* abrt-plugin\*
Loaded plugins: presto, refresh-packagekit, security
Installed Packages
abrt-addon-ccpp.x86_64                          0.0.9-2.fc12
 @fedora
abrt-addon-kerneloops.x86_64                    0.0.9-2.fc12
 @fedora
abrt-addon-python.x86_64                        0.0.9-2.fc12
 @fedora
abrt-plugin-bugzilla.x86_64                     0.0.9-2.fc12
 @fedora
abrt-plugin-kerneloopsreporter.x86_64           0.0.9-2.fc12
 @fedora
abrt-plugin-sqlite3.x86_64                      0.0.9-2.fc12
 @fedora
Available Packages
abrt-plugin-filetransfer.x86_64                 0.0.9-2.fc12
 fedora
abrt-plugin-logger.x86_64                       0.0.9-2.fc12
 fedora
abrt-plugin-mailx.x86_64                        0.0.9-2.fc12
 fedora
abrt-plugin-runapp.x86_64                       0.0.9-2.fc12
 fedora
abrt-plugin-sosreport.x86_64                    0.0.9-2.fc12
 fedora
abrt-plugin-ticketuploader.x86_64               0.0.9-2.fc12
 fedora
```

Example 1.4. Listing all ABRT addons and plugins using glob expressions

- **`yum list all`** — List all installed *and* available packages.
- **`yum list installed`** — List all packages installed on your system. The rightmost column in the output lists the repository the package was retrieved from, where **`installed`** indicates the package came pre-installed as a component of the base system.
- **`yum list available`** — List all available packages in all enabled repositories.
- **`yum grouplist`** — List all package groups.
- **`yum repolist`** — List the repository ID, name, and number of packages it provides for each *enabled* repository.

Displaying Package Info

`yum info <`*`package_name`*`>` `[more_names]` displays information about one or more packages (glob expressions are valid here as well):

```
~]# yum info abrt
```

```
Loaded plugins: presto, refresh-packagekit, security
Installed Packages
Name       : abrt
Arch       : x86_64
Version    : 0.0.9
Release    : 2.fc12
Size       : 525 k
Repo       : installed
From repo  : fedora
Summary    : Automatic bug detection and reporting tool
URL        : https://fedorahosted.org/abrt/
License    : GPLv2+
Description: abrt is a tool to help users to detect defects in applications
 and
           : to create bug reports that include all information required by
 the
           : maintainer to hopefully resolve it. It uses a plugin system to
 extend
           : its functionality.
```

`yum info <package_name>` is similar to the **`rpm -q --info <package_name>`** command, but provides as additional information the ID of the **Yum** repository the RPM package is found in (look for the *From repo:* line in the output).

1.2.2. Installing

You can install a package and all of its non-installed dependencies by entering:

```
#]$ yum install <package_name>
```

You can install multiple packages simultaneously by appending their names as arguments: **`yum install <package_name> [more_names]`**.

If you are installing packages on a *multilib* system, such as an AMD64 or Intel64 machine, you can specify the architecture of the package (as long as it's available in an enabled repository) by appending *`.arch`* to the package name:

```
~]# yum install sqlite2.i586
```

You can use glob expressions to quickly install multiple similarly-named packages:

```
~]# yum install audacious-plugins-\*
```

In addition to package names and glob expressions, you can also provide file names to **`yum install`**. If you know the name of the binary you want to install, but not its package name, you can give **`yum install`** the path name:

```
#]$ yum install /usr/sbin/named
```

yum then searches through its package lists, finds the package which provides /usr/sbin/named, if any, and prompts you as to whether you want to install it.

What if you know you want to install the package that contains the **named** binary, but don't know in which bin or sbin directory that file lives? In that situation, you can give **yum provides** a glob expression:

```
~]# yum provides "*bin/named"
Loaded plugins: presto, refresh-packagekit, security
32:bind-9.6.1-0.3.b1.fc11.x86_64 : The Berkeley Internet Name Domain (BIND)
 DNS (Domain Name System) server
Repo        : fedora
Matched from:
Filename    : /usr/sbin/named
~]# yum install bind
```

Example 1.5. Finding which package owns a file and installing it

Note

yum provides is the same as **yum whatprovides**.

Tip: yum provides/whatprovides and Glob Expressions

yum provides "*/<file_name>" is a common and useful trick to quickly find the package(s) that contain <file_name>.

Installing a Package Group

A package group is similar to a package: it is not useful itself, but installing one also pulls in a group of dependent packages that serve a common purpose. A package group has a name and a groupid. The **yum grouplist -v** command lists the names of all package groups, and, next to each of them, their *groupid* in parentheses. The groupid is always the term in the last pair of parentheses, such as **kde-desktop** and **kde-software-development** in this example:

```
~]# yum -v grouplist kde\*
KDE (K Desktop Environment) (kde-desktop)
KDE Software Development (kde-software-development)
```

You can install a package group by passing its full group name (without the groupid part) to **groupinstall**:

```
~]# yum groupinstall "KDE (K Desktop Environment)"
```

You can also install by groupid:

```
~]# yum groupinstall kde-desktop
```

You can even pass the groupid (or quoted name) to the **install** command if you prepend it with an @-symbol (which tells **yum** that you want to perform a **groupinstall**):

```
~]# yum install @kde-desktop
```

1.2.3. Removing

yum remove <package_name> uninstalls (removes in **RPM** and **Yum** terminology) the package, as well as any packages that depend on it. As when you install multiple packages, you can remove several at once by adding more package names to the command:

```
~]# yum remove foo bar baz
```

Similar to the **install** command, remove can take, as arguments, package names, glob expressions, file lists or package provides.

Warning: Removing a Package when Other Packages Depend On It

Yum is not able to remove a package without also removing packages which depend on it. This type of operation can only be performed by **RPM**, is not advised, and can potentially leave your system in a non-functioning state or cause applications to misbehave and/or crash. For further information, refer to *Section 3.2.3, "Uninstalling"* in the **RPM** chapter.

Removing a Package Group

You can remove a package group using syntax congruent with the **install** syntax:

```
~]# yum groupremove "KDE (K Desktop Environment)"
~]# yum groupremove kde-desktop
~]# yum remove @kde-desktop
```

Example 1.6. Alternative but equivalent ways of removing a package group

1.3. Configuring Yum and Yum Repositories

This section shows you how to:

- set global **Yum** options by editing the **[main]** section of the **/etc/yum.conf** configuration file;

- set options for individual repositories by editing the [*repository*] sections in **/etc/yum.conf** and *.repo* files in the **/etc/yum.repos.d/** directory;
- use **Yum** variables in **/etc/yum.conf** and files in **/etc/yum.repos.d/**so that dynamic version and architecture values are handled correctly; and,
- set up your own custom **Yum** repository.

The **/etc/yum.conf** configuration file contains one mandatory **[main]** section under which you can set **Yum** options. The values that you define in the **[main]** section of **yum.conf** have global effect, and may override values set any individual [*repository*] sections. You can also add [*repository*] sections to **/etc/yum.conf**; however, best practice is to define individual repositories in new or existing **.repo** files in the **/etc/yum.repos.d/**directory. Refer to *Section 1.3.2, “Setting [repository] Options”* if you need to add or edit repository-specific information.

1.3.1. Setting [main] Options

The **/etc/yum.conf** configuration file contains exactly one **[main]** section. You can add many additional options under the **[main]** section heading in **/etc/yum.conf**. Some of the key-value pairs in the **[main]** section affect how **yum** operates; others affect how **Yum** treats repositories. The best source of information for all **Yum** options is in the **[main] OPTIONS** and **[repository] OPTIONS** sections of **man yum.conf**.

Here is a sample **/etc/yum.conf** configuration file:

```
[main]
cachedir=/var/cache/yum
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
installonly_limit=3
[comments abridged]
# PUT YOUR REPOS HERE OR IN separate files named file.repo
# in /etc/yum.repos.d
```

Here is a list of the most commonly-used options in the **[main]** section, and descriptions for each:

cachedir=**/var/cache/yum**

This option specifies the directory where **Yum** should store its cache and database files. By default, **Yum**'s cache directory is **/var/cache/yum**.

keepcache=<1 or 0>

Setting **keepcache=1** instructs **yum** to keep the cache of headers and packages after a successful installation. **keepcache=1** is the default.

reposdir=<absolute path to directory of .repo files>

This option allows you to specify a directory where **.repo** files are located. **.repo** files contain repository information (similar to the [*repository*] section(s) of **/etc/yum.conf**). **yum** collects all repository information from **.repo** files and the [*repository*] section of the **/etc/yum.conf** file to create a master list of repositories to use for transactions. Refer to *Section 1.3.2, “Setting [repository] Options”* for more information about options you can use for both the [*repository*] section and **.repo** files. If **reposdir** is not set, **yum** uses the default directory **/etc/yum.repos.d/**.

gpgcheck=<1 or 0>

This enables or disables GPG signature checking on packages in all repositories, including local package installation. The default is **gpgcheck=0**, which disables GPG-checking. If this option is set in the **[main]** section of the **/etc/yum.conf** file, it sets the GPG-checking rule for all repositories. However, you can also set this on individual repositories instead; i.e., you can enable GPG-checking on one repository while disabling it on another. Setting **gpgcheck=** for individual repositories overrides the default if it is present in **/etc/yum.conf**. Refer to *Section 3.3, “Checking a Package's Signature”* for further information on GPG signature-checking.

assumeyes=<1 or 0>

This determines whether or not **yum** should prompt for confirmation of critical actions. The default is **assumeyes=0**, which means **yum** will prompt you for confirmation. If **assumeyes=1** is set, **yum** behaves in the same way that the command line option `-y` does.

exclude="<package_name> [*more_names*]"

This option allows you to exclude packages by keyword during installation/updates. Listing multiple packages for exclusion can be accomplished by quoting a space-delimited list of packages. Shell globs using wildcards (for example, ***** and **?**) are allowed.

retries=<number_of_retries>

This sets the number of times **yum** should attempt to retrieve a file before returning an error. Setting this to `0` makes **yum** retry forever. The default value is `6`.

1.3.2. Setting [repository] Options

You can define individual **Yum** repositories by adding [*repository*] sections (where *repository* is a unique repository ID, such as [*my_personal_repo*]) to **/etc/yum.conf** or to **.repo** files in the **/etc/yum.repos.d/**directory. All **.repo** files in **/etc/yum.repos.d/**are read by **yum**; best practice is to define your repositories here instead of in **/etc/yum.conf**. You can create new, custom **.repo** files in this directory, add [*repository*] sections to those files, and the next time you run a **yum** command, it will take all newly-added repositories into account.

Here is a (bare-minimum) example of the form a **.repo** file should take:

```
[repository_ID]
name=A Repository Name
baseurl=http://path/to/repo or ftp://path/to/repo or file://path/to/local/
repo
```

Every [*repository*] section must contain the following minimum parts:

[repository_ID]

The repository ID is a unique, one-word (no spaces; underscores are allowed) string of characters (enclosed by brackets) that serves as a repository identifier.

name=<My Repository Name>
: This is a human-readable string describing the repository.

baseurl=http://path/to/repo, ftp://path/to/repo, file://path/to/local/repo
: This is a URL to the directory where the repodata directory of a repository is located. Usually this URL is an HTTP link, such as:

```
   baseurl=http://download.fedoraproject.org/pub/fedora/linux/releases/
$releasever/Everything/$basearch/os/
```

Yum always expands the $releasever, $arch and $basearch variables in URLs. See the following section for explanations of all **Yum** variables: *Section 1.3.3, “Using Yum Variables”*.

- If the repository is available over FTP, use: ftp://path/to/repo
- If the repository is local to the machine, use file://path/to/local/repo
- If a specific online repository requires basic HTTP authentication, you can specify your username and password in the **`baseurl= line`** by prepending it as **`username:password@link`**. For example, if a repository on http://www.example.com/repo/ requires a username of "user" and a password of "password", then the baseurl link can be specified as **`baseurl=http://user:password@www.example.com/repo/`**

Here are some other useful-but-optional [*`repository`*] options:

enabled=<1 or 0>
: Setting **`enabled=0`** instructs **`yum`** not to include that repository as a package source when performing updates and installs. This is an easy way of quickly turning repositories on and off, which is useful when you desire a single package from a repository that you do not want to enable for updates, etc. Turning repositories on and off can also be performed quickly by passing either the `--enablerepo=`*`<repo_name>`* or `--disablerepo=`*`<repo_name>`* option to **`yum`**, or easily through **PackageKit**'s **Add/Remove Software** window. For the latter, refer to *Section 2.2.1, “Refreshing Software Sources (Yum Repositories)”*.

Many more [*`repository`*] options exist. Refer to the **`[repository] OPTIONS`** section of **`man yum.conf`** for the exhaustive list.

1.3.3. Using Yum Variables

You can use and reference the following variables in **`yum`** commands and in all **Yum** configuration files (**`/etc/yum.conf`** and all **`.repo`** files in **`/etc/yum.repos.d/`**.

`$releasever`
: You can use this variable to reference the release version of Fedora. **Yum** obtains the value of `$releasever` from the **`distroverpkg=<value>`** line in the **`/etc/yum.conf`** configuration file. If there is no such line in **`/etc/yum.conf`**, then **`yum`** infers the correct value by deriving the version number from the **`redhat-release`** package.

`$arch`
: You can use this variable to refer to the system's CPU architecture as returned when calling Python's `os.uname()` function. Valid values for `$arch` include: **`i586`**, **`i686`** and **`x86_64`**.

`$basearch`

You can use `$basearch` to reference the base architecture of the system. For example, i686 and i586 machines both have a base architecture of **`i386`**, and AMD64 and Intel64 machines have a base architecture of **`x86_64`**.

`$YUM0-9`

These ten variables are each replaced with the value of any shell environment variables with the same name. If one of these variables is referenced (in **`/etc/yum.conf`** for example) and a shell environment variable with the same name does not exist, then the configuration file variable is not replaced.

1.3.4. Creating a Yum Repository

To set up a **Yum** repository, follow these steps:

Procedure 1.1. Setting Up a **Yum** repository

1. Install the **`createrepo`** package:

   ```
   ~]# yum install createrepo
   ```

2. Copy all of the packages into one directory, such as **`/mnt/local_repo/`**.

3. Run the **`createrepo --database`** command on that directory:

   ```
   ~]# createrepo --database /mnt/local_repo
   ```

This will create the necessary metadata for your **Yum** repository, as well as the **sqlite** database for speeding up **`yum`** operations.

1.4. Yum Plugins

Yum provides plugins that extend and enhance its operations. Certain plugins are installed by default. **Yum** always informs you which plugins, if any, are loaded and in effect whenever you call any **`yum`** command:

```
~]# yum info yum
Loaded plugins: presto, refresh-packagekit, security
[output truncated]
```

Note that the plugin names which follow **`Loaded plugins`** are the names you can provide to the `--disableplugins=<plugin_name>` option.

1.4.1. Enabling, Configuring and Disabling Yum Plugins

To enable **Yum** plugins, ensure that a line beginning with **`plugins=`** is present in the **`[main]`** section of **`/etc/yum.conf`**, and that its value is set to `1`:

```
plugins=1
```

You can disable all plugins by changing this line to **`plugins=0`**.

Every installed plugin has its own configuration file in the **`/etc/yum/pluginconf.d/`** directory. You can set plugin-specific options in these files. For example, here is the **security** plugin's **`security.conf`** configuration file:

```


[main]
enabled=1

```

Example 1.7. A minimal **Yum** plugin configuration file

Plugin configuration files always contain a **`[main]`** section (similar to **Yum**'s **`/etc/yum.conf`** file) in which there is (or you can place if it is missing) an **`enabled=`** option that controls whether the plugin is enabled when you run **`yum`** commands.

If you disable all plugins by setting **`enabled=0`** in **`/etc/yum.conf`**, then all plugins are disabled regardless of whether they are enabled in their individual configuration files.

If you merely want to disable all **Yum** plugins for a single **`yum`** command, use the `--noplugins` option.

If you simply want to disable one or more **Yum** plugins for a single **`yum`** command, then you can add the `--disableplugin=<plugin_name>` option to the command:

```
~]# yum update --disableplugin=presto
```

Example 1.8. Disabling the presto plugin while running yum update

The plugin names you provide to the `--disableplugin=` option are the same names listed after the **`Loaded plugins:`** line in the output of any **`yum`** command. You can disable multiple plugins by separating their names with commas. In addition, you can match multiple similarly-named plugin names or simply shorten long ones by using glob expressions: `--disableplugin=presto,refresh-pack*`.

1.4.2. Installing More Yum Plugins

Yum plugins usually adhere to the **`yum-plugin-<`*`plugin_name`*`>`** package-naming convention, but not always: the package which provides the **presto** plugin is named **`yum-presto`**, for example. You can install a **Yum** plugin in the same way you install other packages:

```
~]# yum install yum-plugin-security
```

1.4.3. Plugin Descriptions

Here are descriptions of a few useful **Yum** plugins:

presto (yum-presto)

The **presto** plugin adds support to **Yum** for downloading *delta RPM* packages, during updates, from repositories which have **presto** metadata enabled. Delta RPMs contain only the differences between the version of the the package installed on the client requesting the RPM package and the updated version in the repository. Downloading a delta RPM is much quicker than downloading the entire updated package, and can speed up updates considerably. Once the delta RPMs are downloaded, they must be rebuilt (the difference applied to the currently-installed package to create the full updated package) on the installing machine, which takes CPU time. Using delta RPMs is therefore a tradeoff between time-to-download, which depends on the network connection, and time-to-rebuild, which is CPU-bound. Using the **presto** plugin is recommended for fast machines and systems with slower network connections, while slower machines on very fast connections *may* benefit more from downloading normal RPM packages, i.e. by disabling **presto**. The **presto** plugin is enabled by default.

protect-packages (yum-plugin-protect-packages)

The **protect-packages** plugin prevents the **`yum`** package and all packages it depends on from being purposefully or accidentally removed. This simple scheme prevents many of the most important packages necessary for your system to run from being removed. In addition, you can list more packages, one per line, in the **`/etc/sysconfig/protected-packages`** file [1] (which you should create if it does not exist), and **protect-packages** will extend protection-from-removal to those packages as well. To temporarily override package protection, use the `--override-protection` option with an applicable **`yum`** command.

refresh-packagekit (PackageKit-yum-plugin)

This plugin updates metadata for **PackageKit** whenever **yum** is run. The **refresh-packagkit** plugin is installed by default.

security (yum-plugin-security)

Discovering information about and applying security updates easily and often is important to all system administrators. For this reason **Yum** provides the **security** plugin, which extends **`yum`** with a set of highly-useful security-related commands, subcommands and options.

You can check for all security-related updates as follows:

```
~]# yum check-update --security
Loaded plugins: presto, refresh-packagekit, security
Limiting package lists to security relevant ones
Needed 3 of 7 packages, for security
elinks.x86_64                   0.12-0.13.pre3.fc11       fedora
kernel.x86_64                   2.6.30.8-64.fc11          fedora
kernel-headers.x86_64           2.6.30.8-64.fc11          fedora
You can then update the system using only the security-related updates (and
 excluding all others, such as bug fix updates) with the command:
~]# yum update --security
```

Example 1.9. Updating only security-related packages

[1] You can also place files with the extension **`.list`** in the **`/etc/sysconfig/protected-packages.d/`** directory (which you should create if it does not exist), and list packages—one per line—in these files. **protect-packages** will protect these too.

Refer to **`man yum-security`** for usage details and further explanation of the enhancements the **security** plugin adds to **`yum`**.

1.5. Additional Resources

The **Yum** home page and wiki — *http://yum.baseurl.org/wiki/Guides*

The **`Yum Guides`** section of the wiki contains more **Yum** documentation.

Managing Software with **Yum** — *http://docs.fedoraproject.org/yum/en/index.html*

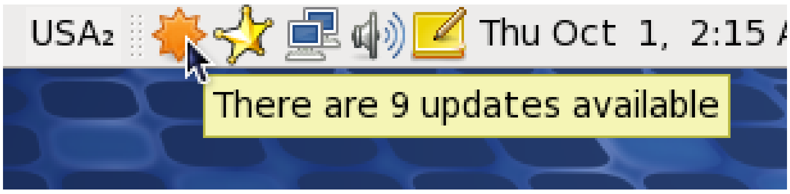
A useful resource that provides additional information about using the **Yum** package manager.

PackageKit

Fedora provides **PackageKit** for viewing, managing, updating, installing and uninstalling packages compatible with your system. **PackageKit** consists of several graphical interfaces that can be opened from the GNOME panel menu, or from the Notification Area when **PackageKit** alerts you that updates are available. For more information on **PackageKit's** architecture and available front ends, refer to *Section 2.3, “PackageKit Architecture”*.

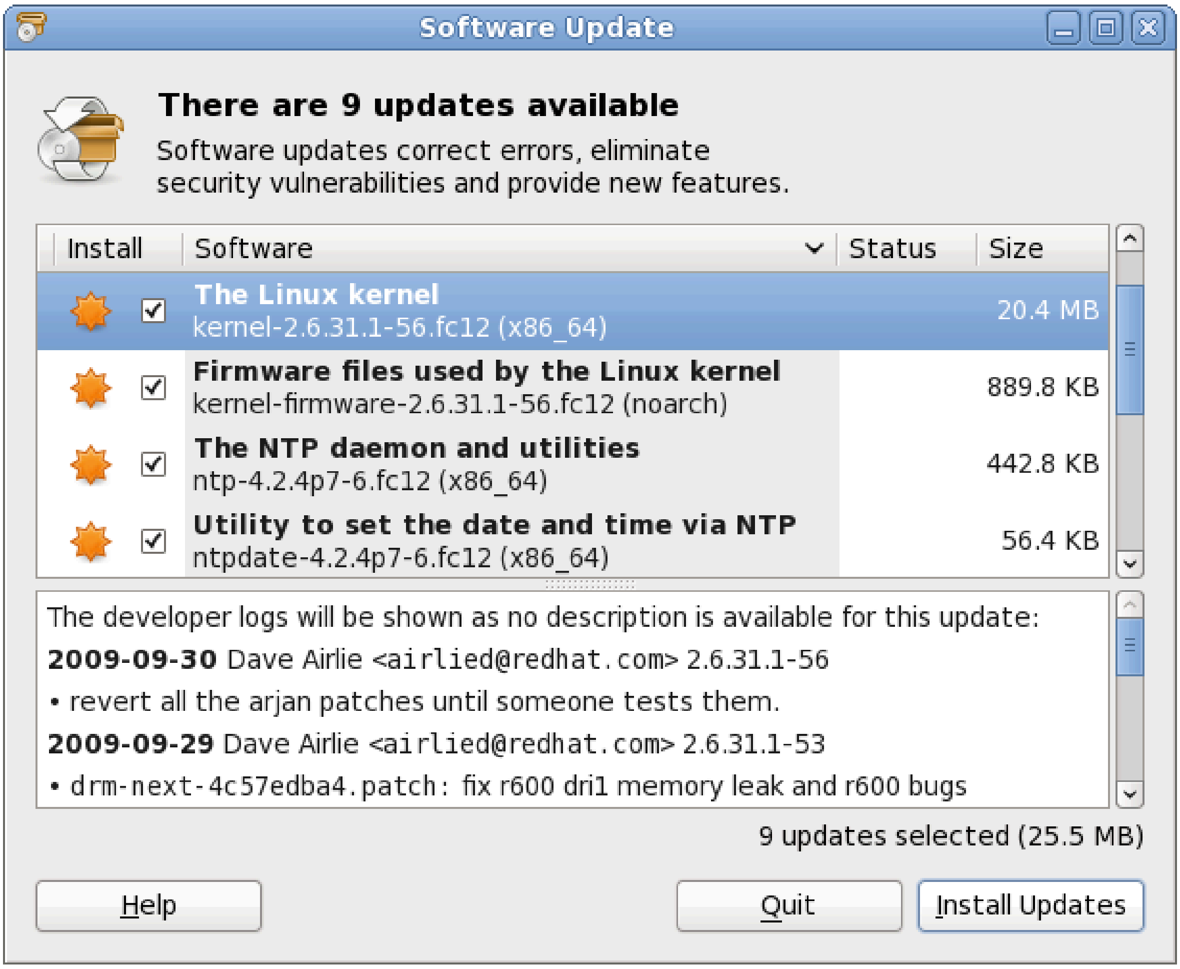
2.1. Updating Packages with Software Update

PackageKit displays a starburst icon in the Notification Area whenever updates are available to be installed on your system.



There are 9 updates available

Clicking on the notification icon opens the **Software Update** window. Alternatively, you can open **Software Updates** by clicking **System** → **Administration** → **Software Update** from the GNOME panel, or running the **`gpk-update-viewer`** command at the shell prompt. In the **Software Updates** window, all available updates are listed along with the names of the packages being updated (minus the **`.rpm`** suffix, but including the CPU architecture), a short summary of the package, and, usually, short descriptions of the changes the update provides. Any updates you do not wish to install can be de-selected here by unchecking the checkbox corresponding to the update.



There are 9 updates available

Software updates correct errors, eliminate security vulnerabilities and provide new features.

Install	Software	Status	Size
☑	**The Linux kernel** kernel-2.6.31.1-56.fc12 (x86_64)		20.4 MB
☑	**Firmware files used by the Linux kernel** kernel-firmware-2.6.31.1-56.fc12 (noarch)		889.8 KB
☑	**The NTP daemon and utilities** ntp-4.2.4p7-6.fc12 (x86_64)		442.8 KB
☑	**Utility to set the date and time via NTP** ntpdate-4.2.4p7-6.fc12 (x86_64)		56.4 KB

The developer logs will be shown as no description is available for this update:

2009-09-30 Dave Airlie <airlied@redhat.com> 2.6.31.1-56

- revert all the arjan patches until someone tests them.

2009-09-29 Dave Airlie <airlied@redhat.com> 2.6.31.1-53

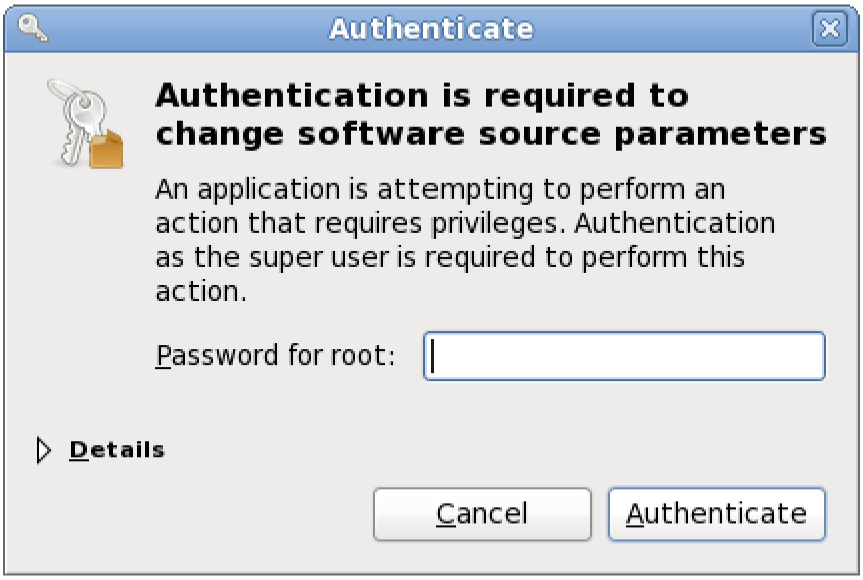
- drm-next-4c57edba4.patch: fix r600 dri1 memory leak and r600 bugs

9 updates selected (25.5 MB)

Figure 2.1. Installing updates with Software Update

The updates presented in the **Software Updates** window only represent the currently-installed packages on your system for which updates are available; dependencies of those packages, whether they are existing packages on your system or new ones, are not shown until you click **Install Updates**.

PackageKit utilizes the fine-grained user authentication capabilities provided by the **PolicyKit** toolkit whenever you request it to make changes to the system. Whenever you instruct **PackageKit** to update, install or remove packages, you will be prompted to enter the superuser password before changes are made to the system.



Authenticate

Authentication is required to change software source parameters

An application is attempting to perform an action that requires privileges. Authentication as the super user is required to perform this action.

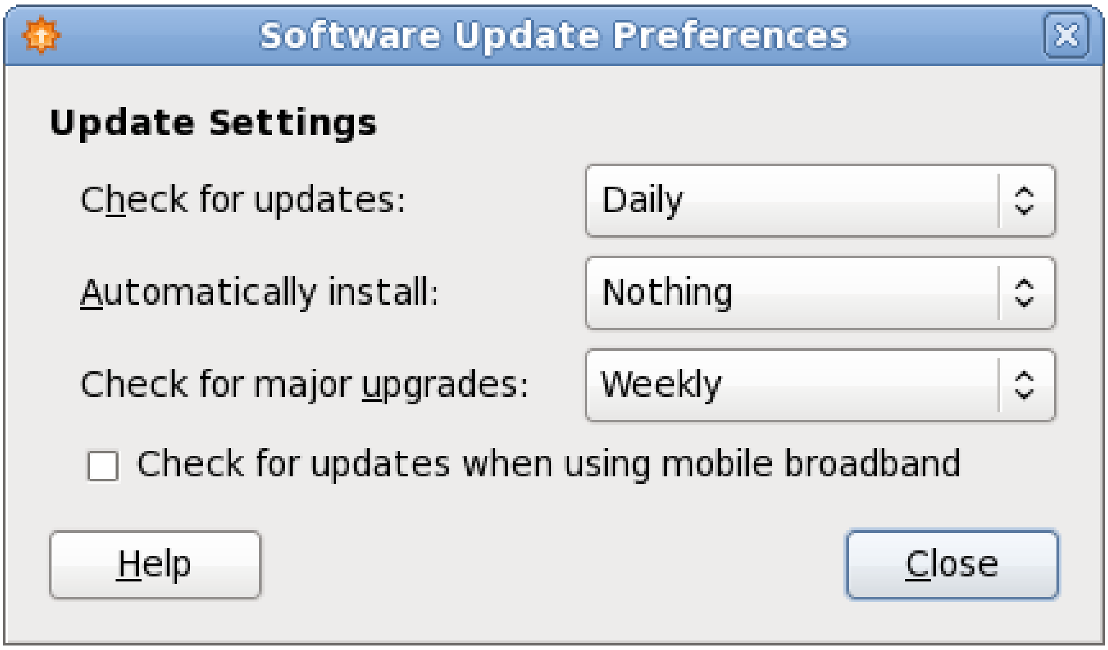
Password for root:

Figure 2.2. PackageKit uses PolicyKit to authenticate

If you instruct **PackageKit** to update the **`kernel`** package, then it will prompt you after installation, asking you whether you want to reboot the system and thereby boot into the newly-installed kernel.

Setting the Update-Checking Interval

Right-clicking on **PackageKit**'s Notification Area icon and clicking **Preferences** opens the **Software Update Preferences** window, where you can define the interval at which **PackageKit** checks for package updates, as well as whether or not to automatically install all updates or only security updates, and how often to check for major upgrades. Leaving the **Check for updates when using mobile broadband** box unchecked is handy for avoiding extraneous bandwidth usage when using a wireless connection on which you are charged for the amount of data you download.



Software Update Preferences

Update Settings

Check for updates: Daily

Automatically install: Nothing

Check for major upgrades: Weekly

☐ Check for updates when using mobile broadband

Figure 2.3. Setting PackageKit's update-checking interval

2.2. Using Add/Remove Software

PackageKit's **Software Update** GUI window is a separate application from its **Add/Remove Software** application, although the two have intuitively similar interfaces. To find and install a new package, on the GNOME panel click on **System** → **Administration** → **Add/Remove Software**, or run the **`gpk-application`** command at the shell prompt.

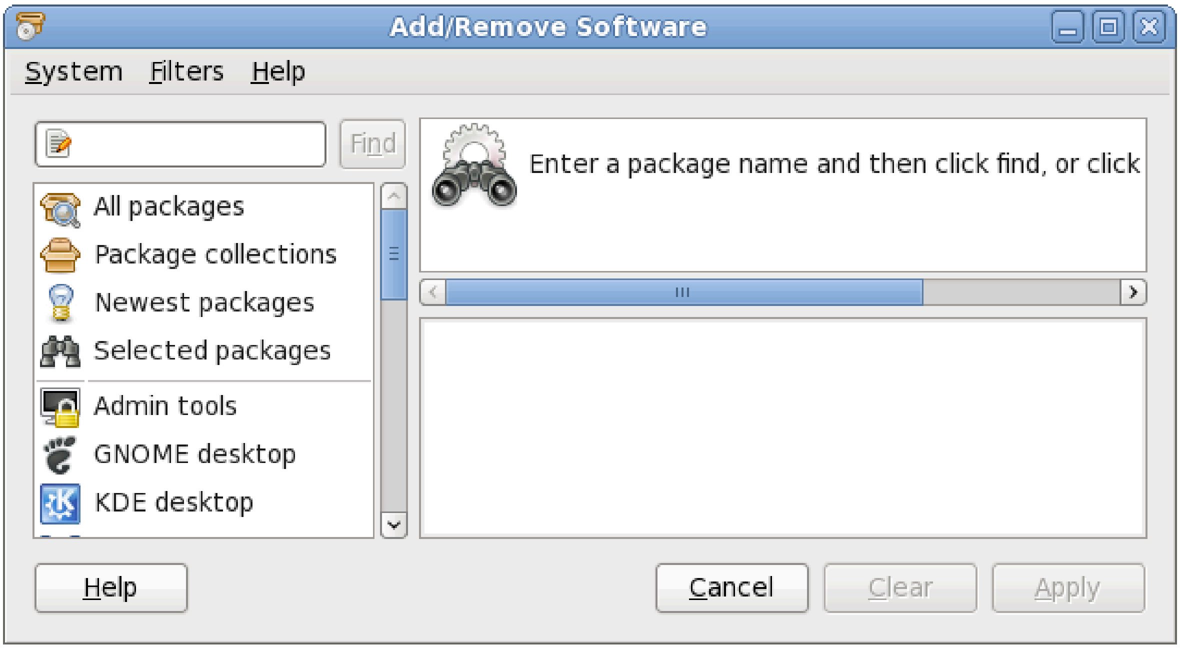


Figure 2.4. PackageKit's Add/Remove Software window

2.2.1. Refreshing Software Sources (Yum Repositories)

PackageKit refers to **Yum** repositories as *software sources*. It obtains all packages from enabled software sources. You can view the list of all *configured* and unfiltered (see below) **Yum** repositories by opening **Add/Remove Software** and clicking **System** → **Software sources**. The **Software Sources** dialog shows the repository name, as written on the `name=<My Repository Name>` field of all [*repository*] sections in the **`/etc/yum.conf`** configuration file, and in all ***`repository`*`.repo`** files in the **`/etc/yum.repos.d/`** directory.

Entries which are checked in the **Enabled** column indicate that the corresponding repository will be used to locate packages to satisfy all update and installation requests (including dependency resolution). The **Enabled**column corresponds to the `enabled=<1 or 0>` field in [*repository*] sections. Checking an unchecked box enables the Yum repository, and unchecking it disables it. Performing either function causes **PolicyKit** to prompt for superuser authentication to enable or disable the repository. **PackageKit** actually inserts the **`enabled=<1 or 0>`** line into the correct [*repository*] section if it does not exist, or changes the value if it does. This means that enabling or disabling a repository through the **Software Sources** window causes that change to persist after closing the window or rebooting the system. The ability to quickly enable and disable repositories based on our needs is a highly-convenient feature of **PackageKit**.

Note that it is not possible to add or remove **Yum** repositories through **PackageKit**. Refer to *Section 1.3, “Configuring Yum and Yum Repositories”* for information on how to set up and configure **Yum** repositories.

Showing Source RPM, Test and Debuginfo Repositories

Checking the box at the bottom of the **Software Sources** window causes **PackageKit** to display source RPM, testing and debuginfo repositories as well. This box is unchecked by defaut.

After enabling and/or disabling the correct **Yum** repositories, ensure that you have the latest list of available packages. Click on **System** → **Refresh package lists** and **PackageKit** will obtain the latest lists of packages from all enabled software sources, i.e. **Yum** repositories.

2.2.2. Finding Packages with Filters

Once the software sources have been updated, it is often beneficial to apply some filters so that **PackageKit** retrieves the results of our **Find** queries faster. This is especially helpful when performing many package searches. Four of the filters in the **Filters** drop-down menu are used to split results by matching or not matching a single criterium. By default when **PackageKit** starts, these filters are all unapplied (**No filter**), but once you do filter by one of them, that filter remains set until you either change it or close **PackageKit**.

Because you are usually searching for available packages that are *not* installed on the system, click **Filters** → **Installed** and select the **Only available** radio button.

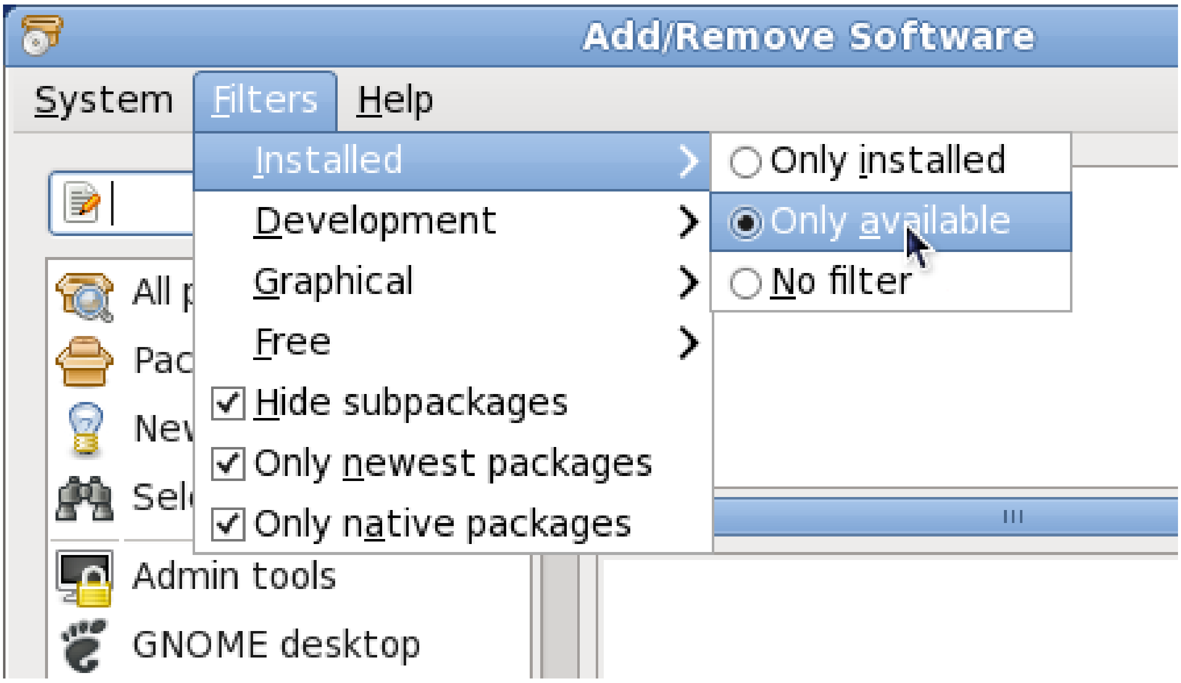


Figure 2.5. Filtering out already-installed packages

Also, unless we require development files such as C header files, we can filter for **Only end user files** and, in doing so, filter out all of the **<*package_name*>-devel** packages we are not interested in.

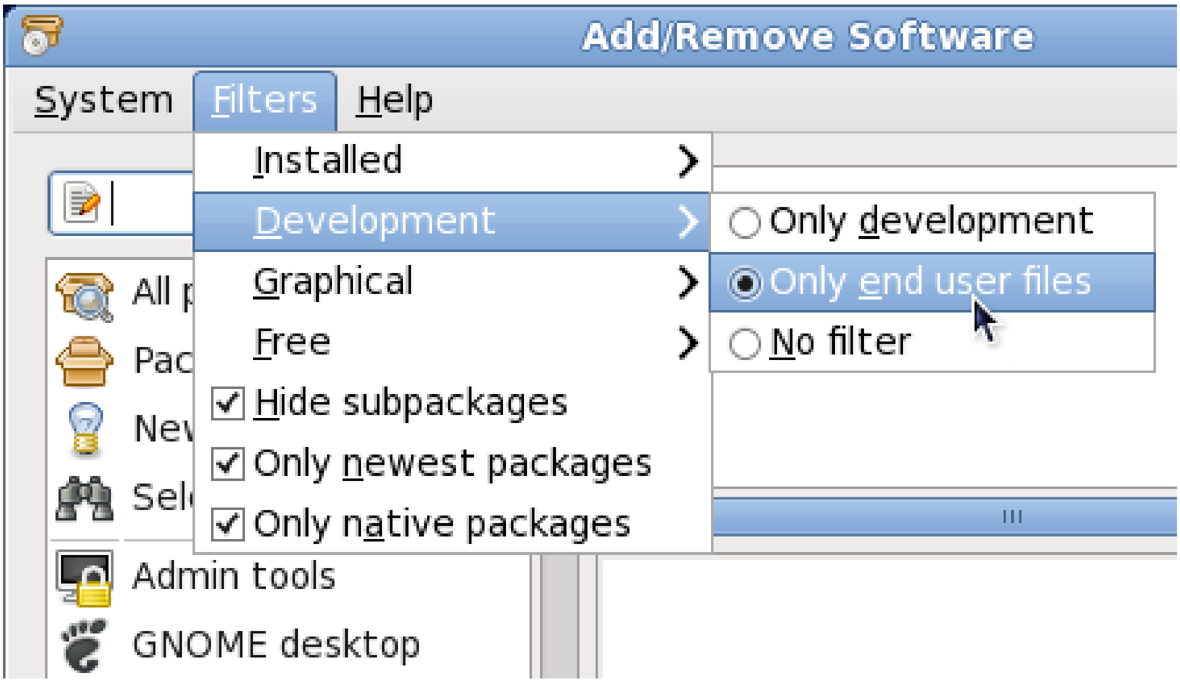


Figure 2.6. Filtering out development packages from the list of Find results

The two remaining filters with submenus are:

Graphical
: Narrows the search to either applications which provide a GUI interface or those that do not (**Only text**). This filter is useful when browsing for GUI applications that perform a specific function.

Free
: Search for packages which are considered to be free software Refer to the *Fedora Licensing List*[1] for details on licenses approved by the Fedora Project.

The remaining checkbox filters are always either checked or unchecked. They are:

Hide subpackages
: Checking the **Hide subpackages** checkbox filters out generally-uninteresting packages that are typically only dependencies of other packages that we want. For example, checking **Hide subpackages** and searching for ***<package>*** would cause the following related packages to be filtered out of the **Find** results (if it exists):

- ***<package>*-devel**
- ***<package>*-libs**
- ***<package>*-libs-devel**
- ***<package>*-debuginfo**

Only newest items
: Checking **Only newest items** filters out all older versions of the same package from the list of results, which is generally what we want.

[1] https://fedoraproject.org/wiki/Licensing#SoftwareLicenses

Important: Using the Only newest items filter

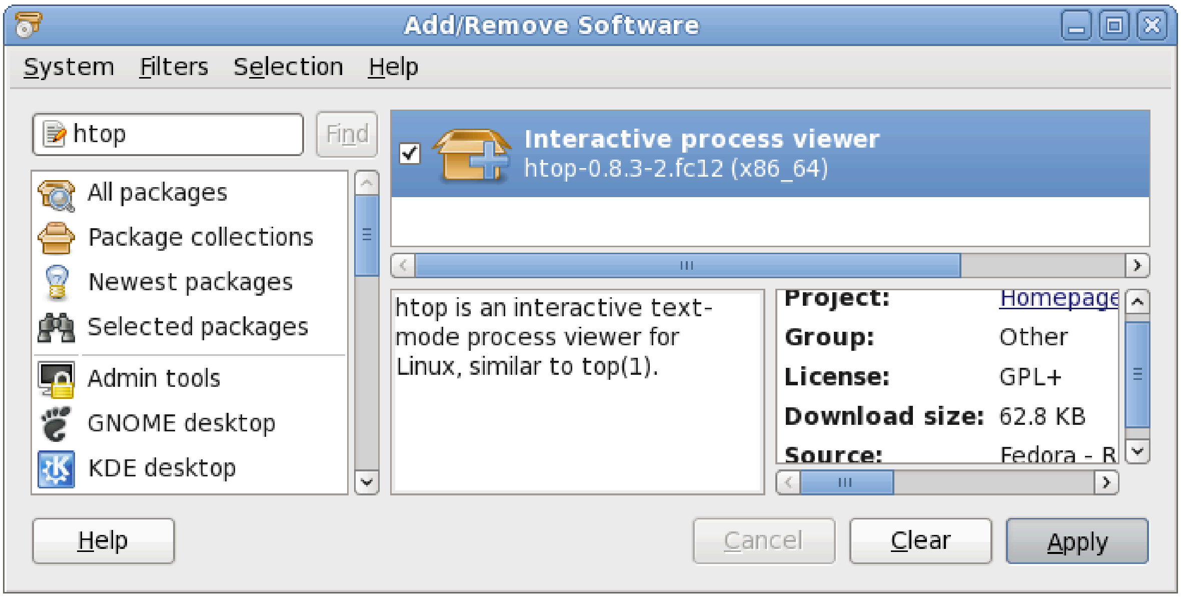
Checking **Only newest items** filters out all but the most recent version of any package from the results list. This filter is often combined with the **Only available** filter to search for the latest available versions of new (not installed) packages.

Only native packages

Checking the **Only native packages** box on a multilib system causes **PackageKit** to omit listing results for packages compiled for the architecture that runs in *compatibility mode*. For example, enabling this filter on a 64-bit system with an AMD64 CPU would cause all packages built for the 32-bit x86 CPU architecture not to be shown in the list of results, even though those packages are able to run on an AMD64 machine. Packages which are architecture-agnostic (i.e. *noarch* packages such as **`crontabs-1.10-31.fc12.noarch.rpm`**) are never filtered out by checking **Only native packages**. This filter has no affect on non-multilib systems, such as *x*86 machines.

2.2.3. Installing and Removing Packages (and Dependencies)

With the two filters selected, **Only available** and **Only end user files**, search for the **htop** interactive process viewer and highlight the package. You now have access to some very useful information about it, including: a clickable link to the project homepage; the **Yum** package group it is found in, if any; the license of the package; a pointer to the GNOME menu location from where the application can be opened, if applicable (**Applications** → **System Tools** → **Htop** in our case); and the size of the package, which is relevant when we download and install it.



htop is an interactive text-mode process viewer for Linux, similar to top(1).

Project: Homepage
Group: Other
License: GPL+
Download size: 62.8 KB
Source: Fedora - R

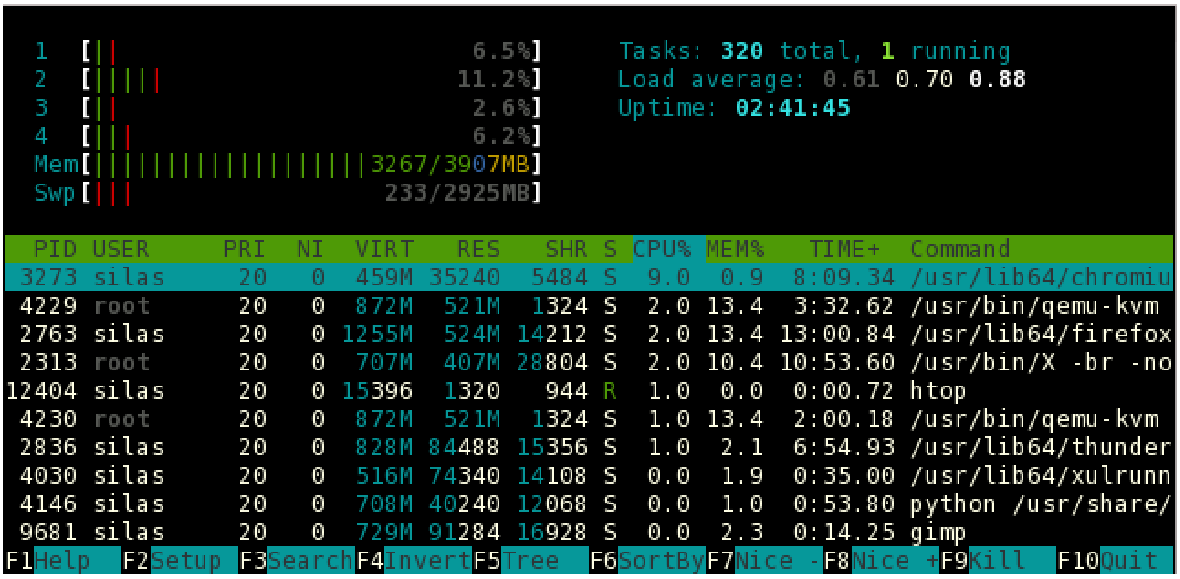
Figure 2.7. Viewing and installing a package with PackageKit's Add/Remove Software window

When the checkbox next to a package or group is checked, then that item is already installed on the system. Checking an unchecked box causes it to be *marked* for installation, which only occurs when the **Apply** button is clicked. In this way, you can search for and select multiple packages or package groups before performing the actual installation transactions. Additionally, you can remove installed packages by unchecking the checked box, and the removal will occur along with any pending installations when **Apply** is pressed. Dependency resolution, which may add additional packages to be installed or removed, is performed after pressing **Apply**. **PackageKit** will then display a window listing those additional packages to install or remove, and ask for confirmation to proceed.

Check **htop** and click the **Apply** button. You will then be prompted for the superuser password; enter it, and **PackageKit** will install **htop**. One nice feature of **PackageKit** is that, following installation, it sometimes presents you with a list of your newly-installed applications and offer you the choice of running them immediately. Alternatively, you will remember that finding a package and selecting it in the **Add/Remove Software** window shows you the **Location** of where in the GNOME menus its application shortcut is located, which is helpful when you want to run it.

Once it is installed, you can run `htop`, an colorful and enhanced version of the `top` process viewer, by opening a shell prompt and entering:

```
~]$ htop
```



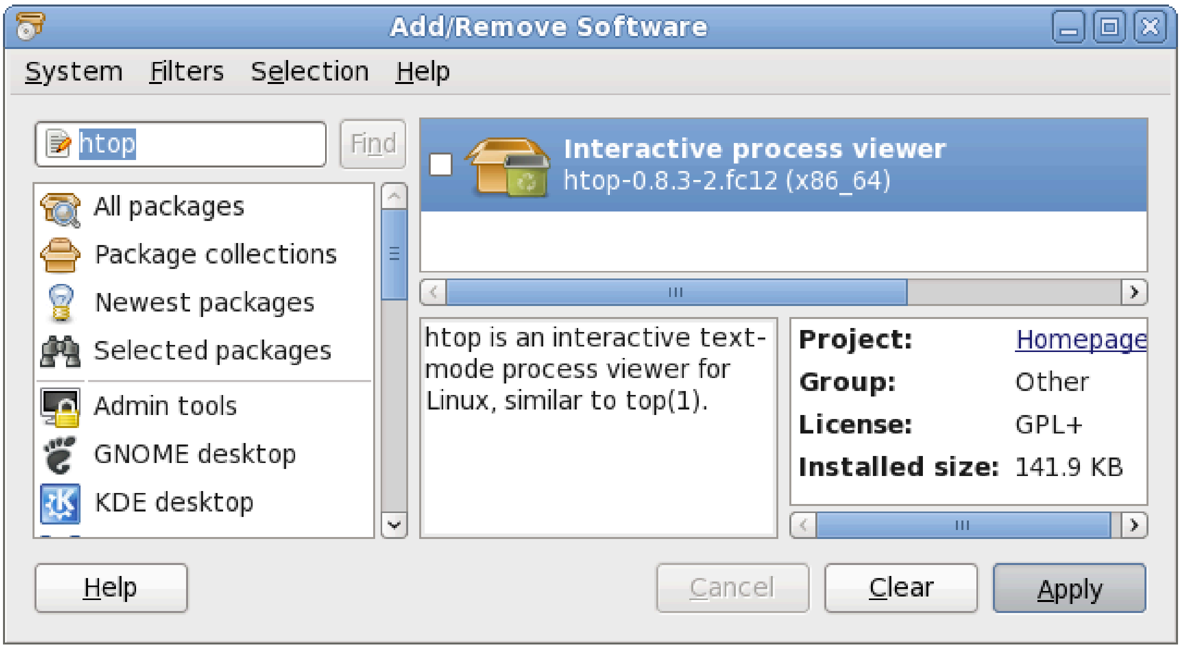
```
  1  [||                         6.5%]     Tasks: 320 total, 1 running
  2  [|||||                     11.2%]     Load average: 0.61 0.70 0.88
  3  [||                         2.6%]     Uptime: 02:41:45
  4  [|||                        6.2%]
  Mem[|||||||||||||||||||3267/3907MB]
  Swp[|||                  233/2925MB]

  PID USER     PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 3273 silas     20   0  459M 35240  5484 S  9.0  0.9  8:09.34 /usr/lib64/chromiu
 4229 root      20   0  872M  521M  1324 S  2.0 13.4  3:32.62 /usr/bin/qemu-kvm
 2763 silas     20   0 1255M  524M 14212 S  2.0 13.4 13:00.84 /usr/lib64/firefox
 2313 root      20   0  707M  407M 28804 S  2.0 10.4 10:53.60 /usr/bin/X -br -no
12404 silas     20   0 15396  1320   944 R  1.0  0.0  0:00.72 htop
 4230 root      20   0  872M  521M  1324 S  1.0 13.4  2:00.18 /usr/bin/qemu-kvm
 2836 silas     20   0  828M 84488 15356 S  1.0  2.1  6:54.93 /usr/lib64/thunder
 4030 silas     20   0  516M 74340 14108 S  0.0  1.9  0:35.00 /usr/lib64/xulrunn
 4146 silas     20   0  708M 40240 12068 S  0.0  1.0  0:53.80 python /usr/share/
 9681 silas     20   0  729M 91284 16928 S  0.0  2.3  0:14.25 gimp
F1Help  F2Setup F3SearchF4InvertF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

Viewing processes with **htop**!

htop is nifty, but we decide that `top` is good enough for us and we want to uninstall it. Remembering that we need to change the **Only installed** filter we recently used to install it to **Only installed** in **Filters** → **Installed**, we search for **htop** again and uncheck it. The program did not install any dependencies of its own; if it had, those would be automatically removed as well, as long as they were not also dependencies of any other packages still installed on our system.

Warning: Removing a Package when Other Packages Depend On It

Although **PackageKit** automatically resolves dependencies during package installation and removal, it is unable to remove a package without also removing packages which depend on it. This type of operation can only be performed by **RPM**, is not advised, and can potentially leave your system in a non-functioning state or cause applications to misbehave and/or crash. For further information, refer to *Section 3.2.3, “Uninstalling”*.



htop

☐ Interactive process viewer
htop-0.8.3-2.fc12 (x86_64)

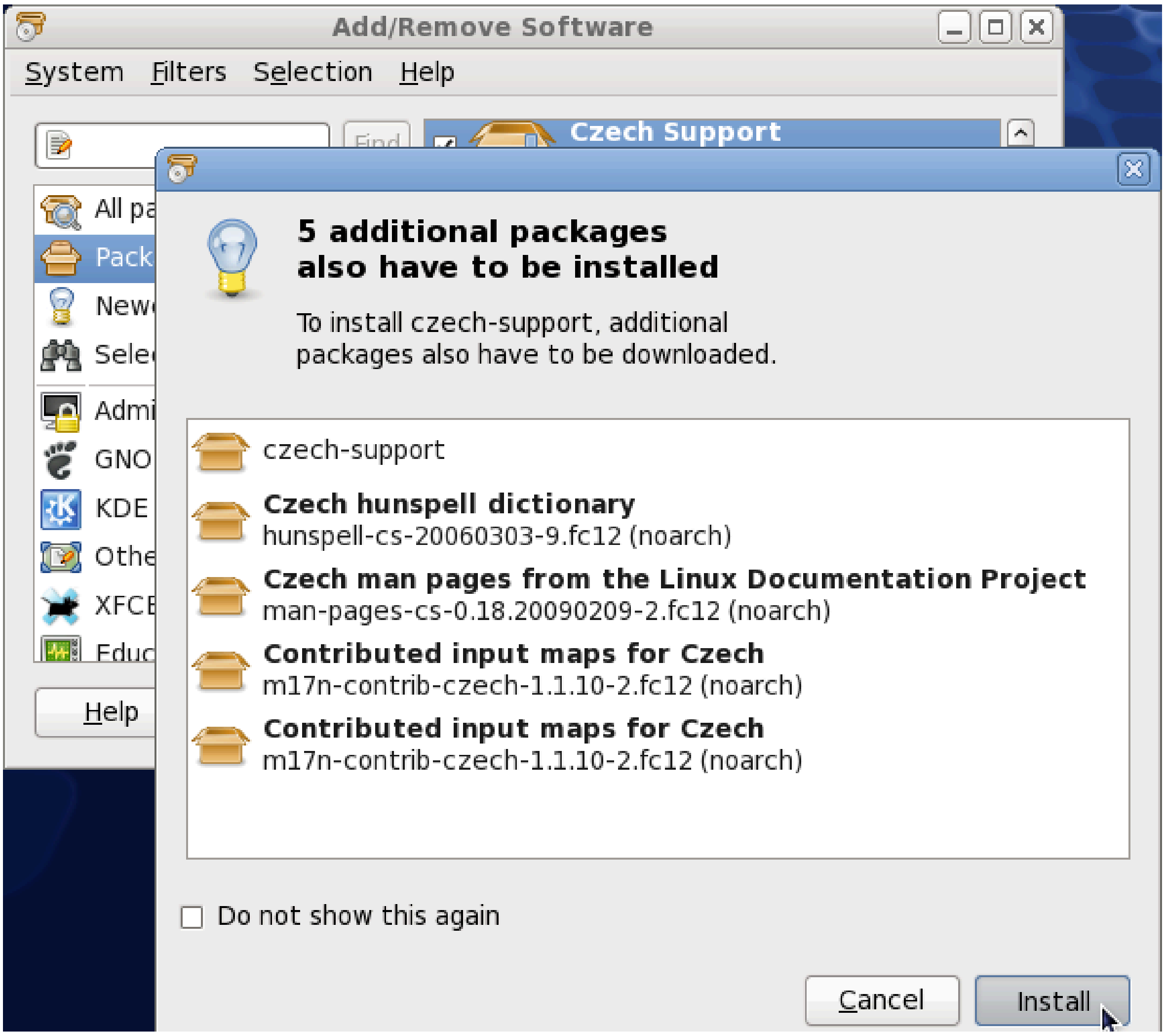
htop is an interactive text-mode process viewer for Linux, similar to top(1).

Project:	Homepage
Group:	Other
License:	GPL+
Installed size:	141.9 KB

Figure 2.8. Removing a package with PackageKit's Add/Remove Software window

2.2.4. Installing and Removing Package Groups

PackageKit also has the ability to install **Yum** package groups, which it calls **Package collections**. Clicking on **Package collections** in the top-left list of categories in the **Software Updates** window allows us to scroll through and find the package group we want to install. In this case, we want to install Czech language support (the **Czech Support** group). Checking the box and clicking **apply** informs us how many *additional* packages must be installed in order to fulfill the dependencies of the package group.



5 additional packages also have to be installed

To install czech-support, additional packages also have to be downloaded.

czech-support

Czech hunspell dictionary
hunspell-cs-20060303-9.fc12 (noarch)

Czech man pages from the Linux Documentation Project
man-pages-cs-0.18.20090209-2.fc12 (noarch)

Contributed input maps for Czech
m17n-contrib-czech-1.1.10-2.fc12 (noarch)

Contributed input maps for Czech
m17n-contrib-czech-1.1.10-2.fc12 (noarch)

☐ Do not show this again

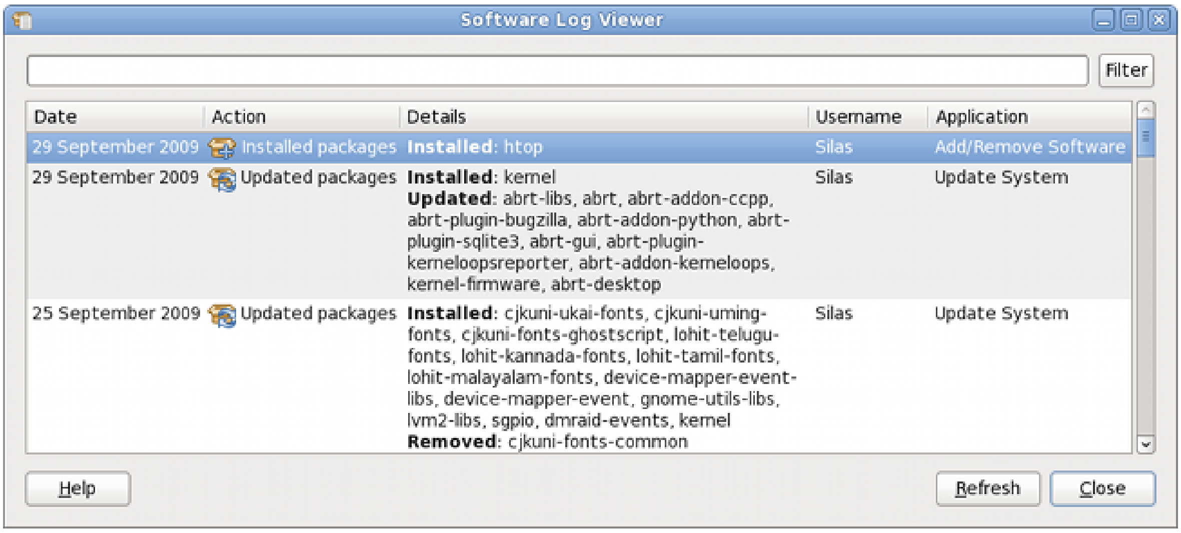
Figure 2.9. Installing the Czech Support package group

Similarly, installed package groups can be uninstalled by selecting **Package collections**, unchecking the appropriate checkbox, and **Apply**ing.

2.2.5. Viewing the Transaction Log

PackageKit maintains a log of the transactions that it performs. To view the log, from the **Add/Remove Software** window, click **System** → **Software log**, or run the **gpk-log** command at the shell prompt.

The **Software Log Viewer** shows the **Action**, such as *`Updated System`* or *`Installed Packages`*, the **Date** on which that action was performed, the **Username** of the user who performed the action, and the front end **Application** the user used (such as *`Update Icon`*, or *`kpackagekit`*). The **Details** column provides the types of the transactions, such as *`Updated`*, *`Installed`* or *`Removed`*, as well as the list of packages the transactions were performed on.



Date	Action	Details	Username	Application
29 September 2009	Installed packages	**Installed**: htop	Silas	Add/Remove Software
29 September 2009	Updated packages	**Installed**: kernel **Updated**: abrt-libs, abrt, abrt-addon-ccpp, abrt-plugin-bugzilla, abrt-addon-python, abrt-plugin-sqlite3, abrt-gui, abrt-plugin-kerneloopsreporter, abrt-addon-kerneloops, kernel-firmware, abrt-desktop	Silas	Update System
25 September 2009	Updated packages	**Installed**: cjkuni-ukai-fonts, cjkuni-uming-fonts, cjkuni-fonts-ghostscript, lohit-telugu-fonts, lohit-kannada-fonts, lohit-tamil-fonts, lohit-malayalam-fonts, device-mapper-event-libs, device-mapper-event, gnome-utils-libs, lvm2-libs, sgpio, dmraid-events, kernel **Removed**: cjkuni-fonts-common	Silas	Update System

Figure 2.10. Viewing the log of package management transactions with the Software Log Viewer

Typing the name of a package in the top text entry field filters the list of transactions to those which affected that package.

2.3. PackageKit Architecture

Fedora provides the **PackageKit** suite of applications for viewing, updating, installing and uninstalling packages and package groups compatible with your system. Architecturally, **PackageKit** consists of several graphical front ends that communicate with the **`packagekitd`** daemon back end, which communicates with a package manager-specific back end that utilizes **Yum** (on Fedora) to perform the actual transactions, such as installing and removing packages, etc.

Table 2.1, “PackageKit GUI Windows, Menu Locations, and Shell Prompt Commands” shows the name of the GUI window, how to start the window from the GNOME desktop or from the **Add/Remove Software** window, and the name of the command line application that opens that window.

Window Title	Function	How to Open	Shell Command
Add/Remove Software	Install, remove or view package info	From the GNOME panel: **System** → **Administration** → **Add/Remove Software**	gpk-application
Software Update	Perform package updates	From the GNOME panel: **System** → **Administration** → **Software Update**	gpk-update-viewer
Software Sources	Enable and disable **Yum** repositories	From **Add/Remove Software**: **System** → **Software sources**	gpk-repo
Software Log Viewer	View the transaction log	From **Add/Remove Software**: **System** → **Software log**	gpk-log

Window Title	Function	How to Open	Shell Command
Software Update Preferences	Set **PackageKit** preferences		gpk-prefs
(Notification Area Alert)	Alerts you when updates are available	From the GNOME panel: **System** → **Preferences** → **Session and Startup**, **Application Autostart** tab	gpk-update-icon

Table 2.1. PackageKit GUI Windows, Menu Locations, and Shell Prompt Commands

The **packagekitd** daemon runs outside the user session and communicates with the various graphical front ends. The **packagekitd** daemon[2] communicates via the **DBus** system message bus with another back end, which utilizes **Yum**'s Python API to perform queries and make changes to the sytem. On Linux systems other than Fedora, **packagekitd** can communicate with other back ends that are able to utilize the native package manager for that system. This modular architecture provides the abstraction necessary for the graphical interfaces to work with many different package managers to perform essentially the same types of package management tasks. Learning how to use the **PackageKit** front ends means that you can use the same familiar graphical interface across many different Linux distributions, even when they utilize a native package manager other than **Yum**.

In addition, **PackageKit**'s separation of concerns provides reliability in that a crash of one of the GUI windows—or even the user's X Window session—will not affect any package management tasks being supervised by the **packagekitd** daemon, which runs outside of the user session.

All of the front end graphical applications discussed in this chapter are provided by the **gnome-packagekit** package instead of by **PackageKit** and its dependencies. Users working in a KDE environment may prefer to install the **kpackagekit** package, which provides a KDE interface for **PackageKit**.

Finally, **PackageKit** also comes with a console-based frontend called **pkcon**.

2.4. Additional Resources

PackageKit home page — *http://www.packagekit.org/index.html*
Information about and mailing lists for **PackageKit**.

PackageKit FAQ — *http://www.packagekit.org/pk-faq.html*
An informative list of Frequently Asked Questions for the **PackageKit** software suite.

PackageKit Feature Matrix — *http://www.packagekit.org/pk-matrix.html*
Cross-reference **PackageKit**-provided features with the long list of package manager back ends.

[2] System daemons are typically long-running processes that provide services to the user or to other programs, and which are started, often at boot time, by special initialization scripts (often shortened to *init scripts*). Daemons respond to the **service** command and can be turned on or off permanently by using the **chkconfig on** or **chkconfig off**commands. They can typically be recognized by a “*d*” appended to their name, such as the **packagekitd** daemon. Refer to *Chapter 6, Controlling Access to Services* for information about system services.

RPM

The *RPM Package Manager* (RPM) is an open packaging system, which runs on Fedora as well as other Linux and UNIX systems. Red Hat, Inc. and the Fedora Project encourage other vendors to use RPM for their own products. RPM is distributed under the terms of the *GPL* (*GNU General Public License*).

The RPM Package Manager only works with packages built to work with the *RPM format*. RPM is itself provided as a pre-installed rpm package. For the end user, RPM makes system updates easy. Installing, uninstalling and upgrading RPM packages can be accomplished with short commands. RPM maintains a database of installed packages and their files, so you can invoke powerful queries and verifications on your system. If you prefer a graphical interface, you can use the **PackageKit** GUI to perform many RPM commands. Refer to *Chapter 2, PackageKit* for details.

Important

When installing a package, ensure it is compatible with your operating system and processor architecture. This can usually be determined by checking the package name. Many of the following examples show RPM packages compiled for the AMD64/Intel 64 computer architectures; thus, the RPM file name ends in **`x86_64.rpm`**.

During upgrades, RPM handles configuration files carefully, so that you never lose your customizations—something that you cannot accomplish with regular **`.tar.gz`** files.

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

Note

Because RPM makes changes to your system, you must be logged in as root to install, remove, or upgrade an RPM package.

3.1. RPM Design Goals

To understand how to use RPM, it can be helpful to understand the design goals of RPM:

Upgradability

With RPM, you can upgrade individual components of your system without completely reinstalling. When you get a new release of an operating system based on RPM, such as Fedora, you do not need to reinstall a fresh copy of the operating system your machine (as you might need to with operating systems based on other packaging systems). RPM allows intelligent, fully-automated, in-place upgrades of your system. In addition, configuration files in packages are preserved across upgrades, so you do not lose your customizations. There are no special upgrade files needed to upgrade a package because the same RPM file is used to both install and upgrade the package on your system.

Powerful Querying

RPM is designed to provide powerful querying options. You can perform searches on your entire database for packages or even just certain files. You can also easily find out what package a file belongs to and from where the package came. The files an RPM package contains are in a compressed archive, with a custom binary header containing useful information about the package and its contents, allowing you to query individual packages quickly and easily.

System Verification

Another powerful RPM feature is the ability to verify packages. If you are worried that you deleted an important file for some package, you can verify the package. You are then notified of anomalies, if any—at which point you can reinstall the package, if necessary. Any configuration files that you modified are preserved during reinstallation.

Pristine Sources

A crucial design goal was to allow the use of *pristine* software sources, as distributed by the original authors of the software. With RPM, you have the pristine sources along with any patches that were used, plus complete build instructions. This is an important advantage for several reasons. For instance, if a new version of a program is released, you do not necessarily have to start from scratch to get it to compile. You can look at the patch to see what you *might* need to do. All the compiled-in defaults, and all of the changes that were made to get the software to build properly, are easily visible using this technique.

The goal of keeping sources pristine may seem important only for developers, but it results in higher quality software for end users, too.

3.2. Using RPM

RPM has five basic modes of operation (not counting package building): installing, uninstalling, upgrading, querying, and verifying. This section contains an overview of each mode. For complete details and options, try **`rpm --help`** or **`man rpm`**. You can also refer to *Section 3.5, “Additional Resources”* for more information on RPM.

3.2.1. Finding RPM Packages

Before using any RPM packages, you must know where to find them. An Internet search returns many RPM repositories, but if you are looking for RPM packages built by the Fedora Project, they can be found at the following locations:

- The Fedora installation media contain many installable RPMs.

- The initial RPM repositories provided with the YUM package manager. Refer to *Chapter 1, Yum* for details on how to use the official Fedora package repositories.

- Unofficial, third-party repositories not affiliated with the Fedora Project also provide RPM packages.

> **Important**
>
> When considering third-party repositories for use with your Fedora system, pay close attention to the repository's web site with regard to package compatibility before adding the repository as a package source. Alternate package repositories may offer different,

incompatible versions of the same software, including packages already included in the Fedora repositories.

3.2.2. Installing

RPM packages typically have file names like **`tree-1.5.2.2-4.fc12.x86_64.rpm`**. The file name includes the package name (**`tree`**), version (**`1.5.2.2`**), release (**`4`**), operating system major version (**`fc12`**) and CPU architecture (**`x86_64`**). Assuming the **`tree-1.5.2.2-4.fc12.x86_64.rpm`** package is in the current directory, log in as root and type the following command at a shell prompt to install it:

```
rpm -ivh tree-1.5.2.2-4.fc12.x86_64.rpm
```

The `-i` option tells **`rpm`** to install the package, and the v and h options, while aren't strictly necessary, increase output information and display a progress meter, respectively.

Alternatively, you can use the `-U` option, which *upgrades* the package if an older version is already installed, or simply installs it if not:

```
rpm -Uvh tree-1.5.2.2-4.fc12.x86_64.rpm
```

If the installation is successful, the following output is displayed:

```
Preparing...                ###########################################
 [100%]
   1:tree                   ###########################################
 [100%]
```

As you can see, RPM prints out the name of the package and then prints a succession of hash marks as a progress meter while the package is installed.

The signature of a package is checked automatically when installing or upgrading a package. The signature confirms that the package was signed by an authorized party. For example, if the verification of the signature fails, an error message such as the following is displayed:

```
error: tree-1.5.2.2-4.fc12.x86_64.rpm: Header V3 RSA/SHA256 signature: BAD,
 key ID
d22e77f2
```

If it is a new, header-only, signature, an error message such as the following is displayed:

```
error: tree-1.5.2.2-4.fc12.x86_64.rpm: Header V3 RSA/SHA256 signature: BAD,
key ID d22e77f2
```

If you do not have the appropriate key installed to verify the signature, the message contains the word `NOKEY`:

```
warning: tree-1.5.2.2-4.fc12.x86_64.rpm: Header V3 RSA/SHA1 signature:
 NOKEY, key ID 57bbccba
```

Refer to *Section 3.3, “Checking a Package's Signature”* for more information on checking a package's signature.

> **Warning**
>
> If you are installing a kernel package, you should always use the **`rpm -ivh`** command (simple install) instead of **`rpm -Uvh`**. The reason for this is that *install* (`-i`) and *upgrade* (`-U`) take on specific meanings when installing kernel packages. Refer to *Chapter 29, Manually Upgrading the Kernel* for details.

3.2.2.1. Package Already Installed

If a package of the same name and version is already installed, the following output is displayed:

```
Preparing...                ###########################################
 [100%]
 package tree-1.5.2.2-4.fc12.x86_64 is already installed
```

However, if you want to install the package anyway, you can use the **`--replacepkgs`** option, which tells RPM to ignore the error:

```
rpm -ivh --replacepkgs tree-1.5.2.2-4.fc12.x86_64.rpm
```

This option is helpful if files installed from the RPM were deleted or if you want the original configuration files from the RPM to be installed.

3.2.2.2. Conflicting Files

If you attempt to install a package that contains a file which has already been installed by another package, the following is displayed:

```
Preparing...
 ##################################################
 file /usr/bin/foobar from install of foo-1.0-1.fc12 conflicts
with file from package bar-3.1.1.fc12
```

To make RPM ignore this error, use the **`--replacefiles`** option:

```
rpm -ivh --replacefiles foo-1.0-1.fc12.x86_64.rpm
```

3.2.2.3. Unresolved Dependency

RPM packages may sometimes depend on other packages, which means that they require other packages to be installed to run properly. If you try to install a package which has an unresolved dependency, output similar to the following is displayed:

```
error: Failed dependencies:
 bar.so.3()(64bit) is needed by foo-1.0-1.fc12.x86_64
    Suggested resolutions:
        bar-3.1.1.fc12.x86_64.rpm
```

If you are installing a package from the Fedora installation media, such as from a CD-ROM or DVD, it usually suggests the package or packages needed to resolve the dependency. Find the suggested package(s) on the Fedora installation media or on one of the active Fedora mirrors (*http://mirrors.fedoraproject.org/publiclist/*) and add it to the command:

```
rpm -ivh foo-1.0-1.fc12.x86_64.rpm bar-3.1.1.fc12.x86_64.rpm
```

If installation of both packages is successful, output similar to the following is displayed:

```
Preparing...                ###########################################
 [100%]
   1:foo                   ###########################################
 [ 50%]
   2:bar                   ###########################################
 [100%]
```

If it does not suggest a package to resolve the dependency, you can try the `--whatprovides` option to determine which package contains the required file.

```
rpm -q --whatprovides "bar.so.3"
```

If the package that contains **`bar.so.3`** is in the RPM database, the name of the package is displayed:

```
bar-3.1.1.fc12.i586.rpm
```

Warning: Forcing Package Installation

Although we can *force***`rpm`** to install a package that gives us a `Failed dependencies` error (using the `--nodeps` option), this is *not* recommended, and will usually result in the installed package failing to run. Installing or removing packages with **`rpm --nodeps`** can cause applications to misbehave and/or crash, and can cause serious package management problems or, possibly, system failure. For these reasons, it is best to heed such warnings; the package manager—whether **RPM**, **Yum** or **PackageKit**—shows us these warnings and suggests possible fixes because accounting for dependencies is critical. The **Yum** package manager can perform dependency resolution and fetch dependencies from online repositories, making it safer, easier and smarter than forcing **`rpm`** to carry out actions without regard to resolving dependencies.

3.2.3. Uninstalling

Uninstalling a package is just as simple as installing one. Type the following command at a shell prompt:

```
rpm -e foo
```

Note

Notice that we used the package *name*`foo`, not the name of the original package *file*, **`foo-1.0-1.fc12.x86_64`**. If you attempt to uninstall a package using the **`rpm -e`** command and the original full file name, you will receive a package name error.

You can encounter dependency errors when uninstalling a package if another installed package depends on the one you are trying to remove. For example:

```

rpm -e ghostscript
error: Failed dependencies:
 libgs.so.8()(64bit) is needed by (installed)
 libspectre-0.2.2-3.fc12.x86_64
 libgs.so.8()(64bit) is needed by (installed) foomatic-4.0.3-1.fc12.x86_64
 libijs-0.35.so()(64bit) is needed by (installed)
 gutenprint-5.2.4-5.fc12.x86_64
 ghostscript is needed by (installed) printer-filters-1.1-4.fc12.noarch
```

Similar to how we searched for a shared object library (i.e. a ***`<library_name>`*****`.so.`*****`<number>`*** file) in *Section 3.2.2.3, “Unresolved Dependency”*, we can search for a 64-bit shared object library using this exact syntax (and making sure to quote the file name):

```
~]# rpm -q --whatprovides "libgs.so.8()(64bit)"
ghostscript-8.70-1.fc12.x86_64
```

Warning: Forcing Package Installation

Although we can *force*`rpm` to remove a package that gives us a `Failed dependencies` error (using the `--nodeps` option), this is *not* recommended, and may cause harm to other installed applications. Installing or removing packages with **`rpm --nodeps`** can cause applications to misbehave and/or crash, and can cause serious package management problems or, possibly, system failure. For these reasons, it is best to heed such warnings; the package manager—whether **RPM**, **Yum** or **PackageKit**—shows us these warnings and suggests possible fixes because accounting for dependencies is critical. The **Yum** package manager can perform dependency resolution and fetch dependencies from online repositories, making it safer, easier and smarter than forcing **`rpm`** to carry out actions without regard to resolving dependencies.

3.2.4. Upgrading

Upgrading a package (using the `-U` option) is similar to installing one (the `-i` option). If we have the RPM named **`tree-1.5.3.0-1.fc12.x86_64.rpm`** in our current directory, and **`tree-1.5.2.2-4.fc12.x86_64.rpm`** is already installed on our system (**`rpm -qi`** will tell us which version of the **`tree`** package we have installed on our system, if any), then the following command will upgrade tree to the newer version:

```
rpm -Uvh tree-1.5.3.0-1.fc12.x86_64.rpm
```

As part of upgrading a package, RPM automatically uninstalls any old versions of the **`foo`** package. Note that `-U` will also install a package even when there are no previous versions of the package installed.

Important

It is not advisable to use the `-U` option for installing kernel packages because RPM completely replaces the previous kernel package. This does not affect a running system, but if the new kernel is unable to boot during your next restart, there would be no other kernel to boot instead.

Using the `-i` option adds the kernel to your GRUB boot menu (**`/etc/grub.conf`**). Similarly, removing an old, unneeded kernel removes the kernel from GRUB.

Because RPM performs intelligent upgrading of packages with configuration files, you may see one or the other of the following messages:

```
saving /etc/foo.conf as /etc/foo.conf.rpmsave
```

This message means that changes you made to the configuration file may not be *forward-compatible* with the new configuration file in the package, so RPM saved your original file and installed a new one. You should investigate the differences between the two configuration files and resolve them as soon as possible, to ensure that your system continues to function properly.

Alternatively, RPM may save the package's *new* configuration file as, for example, **`foo.conf.rpmnew`**, and leave the configuration file you modified untouched. You should still resolve any conflicts between your modified configuration file and the new one, usually by merging changes from the old one to the new one with a **`diff`** program.

If you attempt to upgrade to a package with an *older* version number (that is, if a higher version of the package is already installed), the output is similar to the following:

```
package foo-2.0-1.fc12.x86_64.rpm (which is newer than foo-1.0-1) is
 already installed
```

To force RPM to upgrade anyway, use the **`--oldpackage`** option:

```
rpm -Uvh --oldpackage foo-1.0-1.fc12.x86_64.rpm
```

3.2.5. Freshening

Freshening is similar to upgrading, except that only existent packages are upgraded. Type the following command at a shell prompt:

```
rpm -Fvh foo-2.0-1.fc12.x86_64.rpm
```

RPM's freshen option checks the versions of the packages specified on the command line against the versions of packages that have already been installed on your system. When a newer version of an already-installed package is processed by RPM's freshen option, it is upgraded to the newer version. However, RPM's freshen option does not install a package if no previously-installed package of the same name exists. This differs from RPM's upgrade option, as an upgrade *does* install packages whether or not an older version of the package was already installed.

Freshening works for single packages or package groups. If you have just downloaded a large number of different packages, and you only want to upgrade those packages that are already installed on your system, freshening does the job. Thus, you do not have to delete any unwanted packages from the group that you downloaded before using RPM.

In this case, issue the following with the **`*.rpm`** glob:

```
rpm -Fvh *.rpm
```

RPM then automatically upgrades only those packages that are already installed.

3.2.6. Querying

The RPM database stores information about all RPM packages installed in your system. It is stored in the directory **`/var/lib/rpm/`**, and is used to query what packages are installed, what versions each package is, and to calculate any changes to any files in the package since installation, among other use cases.

To query this database, use the **`-q`** option. The **`rpm -q`** ***`package name`*** command displays the package name, version, and release number of the installed package <*`package_name`*>. For example, using **`rpm -q tree`** to query installed package **`tree`** might generate the following output:

```
tree-1.5.2.2-4.fc12.x86_64
```

You can also use the following *Package Selection Options* (which is a subheading in the RPM man page: see **`man rpm`** for details) to further refine or qualify your query:

- **`-a`** — queries all currently installed packages.

- **`-f <`*`file_name`*`>`** — queries the RPM database for which package owns **`<`*`file_name`*`>`**. Specify the absolute path of the file (for example, **`rpm -qf /bin/ls`** instead of **`rpm -qf ls`**).

- **`-p <`*`package_file`*`>`** — queries the uninstalled package **`<`*`package_file`*`>`**.

There are a number of ways to specify what information to display about queried packages. The following options are used to select the type of information for which you are searching. These are called the *Package Query Options*.

- **`-i`** displays package information including name, description, release, size, build date, install date, vendor, and other miscellaneous information.
- **`-l`** displays the list of files that the package contains.
- **`-s`** displays the state of all the files in the package.
- **`-d`** displays a list of files marked as documentation (man pages, info pages, READMEs, etc.) in the package.
- **`-c`** displays a list of files marked as configuration files. These are the files you edit after installation to adapt and customize the package to your system (for example, **`sendmail.cf`**, **`passwd`**, **`inittab`**, etc.).

For options that display lists of files, add **`-v`** to the command to display the lists in a familiar **`ls -l`** format.

3.2.7. Verifying

Verifying a package compares information about files installed from a package with the same information from the original package. Among other things, verifying compares the file size, MD5 sum, permissions, type, owner, and group of each file.

The command **`rpm -V`** verifies a package. You can use any of the *Verify Options* listed for querying to specify the packages you wish to verify. A simple use of verifying is **`rpm -V tree`**, which verifies that all the files in the **`tree`** package are as they were when they were originally installed. For example:

- To verify a package containing a particular file:

  ```
  rpm -Vf /usr/bin/tree
  ```

  In this example, **`/usr/bin/tree`** is the absolute path to the file used to query a package.

- To verify ALL installed packages throughout the system (which will take some time):

  ```
  rpm -Va
  ```

- To verify an installed package against an RPM package file:

  ```
  rpm -Vp tree-1.5.2.2-4.fc12.x86_64.rpm
  ```

  This command can be useful if you suspect that your RPM database is corrupt.

If everything verified properly, there is no output. If there are any discrepancies, they are displayed. The format of the output is a string of eight characters (a "c" denotes a configuration file) and then the file name. Each of the eight characters denotes the result of a comparison of one attribute of the file to the value of that attribute recorded in the RPM database. A single period (`.`) means the test passed. The following characters denote specific discrepancies:

- 5 — MD5 checksum

- `S` — file size
- `L` — symbolic link
- `T` — file modification time
- `D` — device
- `U` — user
- `G` — group
- `M` — mode (includes permissions and file type)
- `?` — unreadable file (file permission errors, for example)

If you see any output, use your best judgment to determine if you should remove the package, reinstall it, or fix the problem in another way.

3.3. Checking a Package's Signature

If you wish to verify that a package has not been corrupted or tampered with, you can examine just the md5sum by entering this command at the shell prompt: (where <*rpm_file*> is the file name of the RPM package):

```
rpm -K --nosignature <rpm_file>
```

The output `<rpm_file>: rsa sha1 (md5) pgp md5 OK` (specifically the *OK* part of it) indicates that the file was not corrupted during download. To see a more verbose message, replace `-K` with `-Kvv` in the command.

On the other hand, how trustworthy is the developer who created the package? If the package is *signed* with the developer's GnuPG *key*, you know that the developer really is who they say they are.

An RPM package can be signed using *Gnu Privacy Guard* (or GnuPG), to help you make certain your downloaded package is trustworthy.

GnuPG is a tool for secure communication; it is a complete and free replacement for the encryption technology of PGP, an electronic privacy program. With GnuPG, you can authenticate the validity of documents and encrypt/decrypt data to and from other recipients. GnuPG is capable of decrypting and verifying PGP 5.*x* files as well.

During installation, GnuPG is installed by defaut, which enables you to immediately start using it to verify any packages that you download from the Fedora Project. Before doing so, you first need to import the correct Fedora key.

3.3.1. Importing Keys

Fedora GnuPG keys are located in the **`/etc/pki/rpm-gpg/`** directory. To verify a Fedora Project package, first import the correct key based on your processor architecture:

```
rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-x86_64
```

To display a list of all keys installed for RPM verification, execute the command:

```
rpm -qa gpg-pubkey*
```

For the Fedora Project key, the output states:

```
gpg-pubkey-57bbccba-4a6f97af
```

To display details about a specific key, use **`rpm -qi`** followed by the output from the previous command:

```
rpm -qi gpg-pubkey-57bbccba-4a6f97af
```

3.3.2. Verifying Signature of Packages

To check the GnuPG signature of an RPM file after importing the builder's GnuPG key, use the following command (replace <*rpm_file*> with the filename of the RPM package):

```
rpm -K <rpm_file>
```

If all goes well, the following message is displayed: `rsa sha1 (md5) pgp md5 OK`. This means that the signature of the package has been verified, that it is not corrupt, and is therefore safe to install and use.

For more information, including a list of currently-used Fedora Project keys and their fingerprints, refer to *http://fedoraproject.org/en/keys*.

3.4. Practical and Common Examples of RPM Usage

RPM is a useful tool for both managing your system and diagnosing and fixing problems. The best way to make sense of all its options is to look at some examples.

- Perhaps you have deleted some files by accident, but you are not sure what you deleted. To verify your entire system and see what might be missing, you could try the following command:

  ```
  rpm -Va
  ```

  If some files are missing or appear to have been corrupted, you should probably either re-install the package or uninstall and then re-install the package.

- At some point, you might see a file that you do not recognize. To find out which package owns it, enter:

```
rpm -qf /usr/bin/ghostscript
```

The output would look like the following:

```
ghostscript-8.70-1.fc12.x86_64
```

- We can combine the above two examples in the following scenario. Say you are having problems with **/usr/bin/paste**. You would like to verify the package that owns that program, but you do not know which package owns **paste**. Enter the following command,

  ```
  rpm -Vf /usr/bin/paste
  ```

  and the appropriate package is verified.

- Do you want to find out more information about a particular program? You can try the following command to locate the documentation which came with the package that owns that program:

  ```
  rpm -qdf /usr/bin/free
  ```

  The output would be similar to the following:

  ```
  /usr/share/doc/procps-3.2.8/BUGS
  /usr/share/doc/procps-3.2.8/FAQ
  /usr/share/doc/procps-3.2.8/NEWS
  /usr/share/doc/procps-3.2.8/TODO
  /usr/share/man/man1/free.1.gz
  /usr/share/man/man1/pgrep.1.gz
  /usr/share/man/man1/pkill.1.gz
  /usr/share/man/man1/pmap.1.gz
  /usr/share/man/man1/ps.1.gz
  /usr/share/man/man1/pwdx.1.gz
  /usr/share/man/man1/skill.1.gz
  /usr/share/man/man1/slabtop.1.gz
  /usr/share/man/man1/snice.1.gz
  /usr/share/man/man1/tload.1.gz
  /usr/share/man/man1/top.1.gz
  /usr/share/man/man1/uptime.1.gz
  /usr/share/man/man1/w.1.gz
  /usr/share/man/man1/watch.1.gz
  /usr/share/man/man5/sysctl.conf.5.gz
  /usr/share/man/man8/sysctl.8.gz
  /usr/share/man/man8/vmstat.8.gz
  ```

- You may find a new RPM, but you do not know what it does. To find information about it, use the following command:

```
rpm -qip crontabs-1.10-31.fc12.noarch.rpm
```

The output would be similar to the following:

```
Name        : crontabs                     Relocations: (not relocatable)
Version     : 1.10                              Vendor: Fedora Project
Release     : 31.fc12                       Build Date: Sat 25 Jul 2009
 06:37:57 AM CEST
Install Date: (not installed)               Build Host:
 x86-6.fedora.phx.redhat.com
Group       : System Environment/Base       Source RPM:
 crontabs-1.10-31.fc12.src.rpm
Size        : 2486                             License: Public Domain and
 GPLv2
Signature   : RSA/SHA1, Tue 11 Aug 2009 01:11:19 PM CEST, Key ID
 9d1cc34857bbccba
Packager    : Fedora Project
Summary     : Root crontab files used to schedule the execution of
 programs
Description :
The crontabs package contains root crontab files and directories.
You will need to install cron daemon to run the jobs from the crontabs.
The cron daemon such as cronie or fcron checks the crontab files to
see when particular commands are scheduled to be executed.  If commands
are scheduled, it executes them.

Crontabs handles a basic system function, so it should be installed on
your system.
```

- Perhaps you now want to see what files the **`crontabs`** RPM package installs. You would enter the following:

```
rpm -qlp crontabs-1.10-31.fc12.noarch.rpm
```

The output is similar to the following:

```
/etc/cron.daily
/etc/cron.hourly
/etc/cron.monthly
/etc/cron.weekly
/etc/crontab
/usr/bin/run-parts
/usr/share/man/man4/crontabs.4.gz
```

These are just a few examples. As you use RPM, you may find more uses for it.

3.5. Additional Resources

RPM is an extremely complex utility with many options and methods for querying, installing, upgrading, and removing packages. Refer to the following resources to learn more about RPM.

3.5.1. Installed Documentation

- **`rpm --help`** — This command displays a quick reference of RPM parameters.
- **`man rpm`** — The RPM man page gives more detail about RPM parameters than the **`rpm --help`** command.

3.5.2. Useful Websites

- The RPM website — *http://www.rpm.org/*
- The RPM mailing list can be subscribed to, and its archives read from, here — *https://lists.rpm.org/mailman/listinfo/rpm-list*[1]

3.5.3. Related Books

Maximum RPM — *http://www.redhat.com/docs/books/max-rpm/*

The *Maximum RPM* book, which you can read online or download in HTML or PDF, covers everything from general RPM usage to building your own RPMs to programming with rpmlib.

Red Hat RPM Guide — *http://docs.fedoraproject.org/drafts/rpm-guide-en/*

The *Red Hat RPM Guide* by Eric Foster-Johnson is an excellent resource on all details of the RPM package format and the RPM package management utility.

Part II. Network-Related Configuration

After explaining how to configure the network, this part discusses topics related to networking such as how to allow remote logins, share files and directories over the network, and set up a Web server.

Network Interfaces

Under Fedora, all network communications occur between configured software *interfaces* and *physical networking devices* connected to the system.

The configuration files for network interfaces are located in the **`/etc/sysconfig/network-scripts/`** directory. The scripts used to activate and deactivate these network interfaces are also located here. Although the number and type of interface files can differ from system to system, there are three categories of files that exist in this directory:

1. *Interface configuration files*

2. *Interface control scripts*

3. *Network function files*

The files in each of these categories work together to enable various network devices.

This chapter explores the relationship between these files and how they are used.

4.1. Network Configuration Files

Before delving into the interface configuration files, let us first itemize the primary configuration files used in network configuration. Understanding the role these files play in setting up the network stack can be helpful when customizing a Fedora system.

The primary network configuration files are as follows:

`/etc/hosts`
: The main purpose of this file is to resolve hostnames that cannot be resolved any other way. It can also be used to resolve hostnames on small networks with no DNS server. Regardless of the type of network the computer is on, this file should contain a line specifying the IP address of the loopback device (**`127.0.0.1`**) as **`localhost.localdomain`**. For more information, refer to the **`hosts`** man page.

`/etc/resolv.conf`
: This file specifies the IP addresses of DNS servers and the search domain. Unless configured to do otherwise, the network initialization scripts populate this file. For more information about this file, refer to the **`resolv.conf`** man page.

`/etc/sysconfig/network`
: This file specifies routing and host information for all network interfaces. For more information about this file and the directives it accepts, refer to *Section 17.1.21, “**`/etc/sysconfig/network`**”*.

`/etc/sysconfig/network-scripts/ifcfg-<`*`interface-name`*`>`
: For each network interface, there is a corresponding interface configuration script. Each of these files provide information specific to a particular network interface. Refer to *Section 4.2, “Interface Configuration Files”* for more information on this type of file and the directives it accepts.

Warning

The **/etc/sysconfig/networking/** directory is used by the **Network Administration Tool** (**system-config-network**) and its contents should **not** be edited manually. Using only one method for network configuration is strongly encouraged, due to the risk of configuration deletion.

For more information about configuring network interfaces using the **Network Administration Tool**, refer to *Chapter 5, Network Configuration*

4.2. Interface Configuration Files

Interface configuration files control the software interfaces for individual network devices. As the system boots, it uses these files to determine what interfaces to bring up and how to configure them. These files are usually named **ifcfg-<*name*>**, where <*name*> refers to the name of the device that the configuration file controls.

4.2.1. Ethernet Interfaces

One of the most common interface files is **ifcfg-eth0**, which controls the first Ethernet *network interface card* or *NIC* in the system. In a system with multiple NICs, there are multiple **ifcfg-eth<*X*>** files (where <*X*> is a unique number corresponding to a specific interface). Because each device has its own configuration file, an administrator can control how each interface functions individually.

The following is a sample **ifcfg-eth0** file for a system using a fixed IP address:

```
DEVICE=eth0 BOOTPROTO=none ONBOOT=yes NETWORK=10.0.1.0
 NETMASK=255.255.255.0 IPADDR=10.0.1.27 USERCTL=no
```

The values required in an interface configuration file can change based on other values. For example, the **ifcfg-eth0** file for an interface using DHCP looks different because IP information is provided by the DHCP server:

```
DEVICE=eth0 BOOTPROTO=dhcp ONBOOT=yes
```

The **Network Administration Tool** (**system-config-network**) is an easy way to make changes to the various network interface configuration files (refer to *Chapter 5, Network Configuration* for detailed instructions on using this tool).

However, it is also possible to manually edit the configuration files for a given network interface.

Below is a listing of the configurable parameters in an Ethernet interface configuration file:

BONDING_OPTS=<*parameters*>

sets the configuration parameters for the bonding device, and is used in **/etc/sysconfig/network-scripts/ifcfg-bond<*N*>** (see *Section 4.2.3, "Channel Bonding Interfaces"*). These parameters are identical to those used for bonding devices in **/sys/class/net/<*bonding device*>/bonding**, and the module parameters for the bonding driver as described in ***bonding*** *Module Directives*.

This configuration method is used so that multiple bonding devices can have different configurations. If you use **`BONDING_OPTS`** in **`ifcfg-<name>`**, do *not* use **`/etc/modprobe.conf`** to specify options for the bonding device.

`BOOTPROTO=<protocol>`

where **`<protocol>`** is one of the following:

- **`none`** — No boot-time protocol should be used.
- **`bootp`** — The BOOTP protocol should be used.
- **`dhcp`** — The DHCP protocol should be used.

`BROADCAST=<address>`

where **`<address>`** is the broadcast address. This directive is deprecated, as the value is calculated automatically with **`ifcalc`**.

`DEVICE=<name>`

where **`<name>`** is the name of the physical device (except for dynamically-allocated PPP devices where it is the *logical name*).

`DHCP_HOSTNAME`

Use this option only if the DHCP server requires the client to specify a hostname before receiving an IP address.

`DNS{1,2}=<address>`

where **`<address>`** is a name server address to be placed in **`/etc/resolv.conf`** if the **`PEERDNS`** directive is set to **`yes`**.

`ETHTOOL_OPTS=<options>`

where **`<options>`** are any device-specific options supported by **`ethtool`**. For example, if you wanted to force 100Mb, full duplex:

```
ETHTOOL_OPTS="autoneg off speed 100 duplex full"
```

Instead of a custom initscript, use **`ETHTOOL_OPTS`** to set the interface speed and duplex settings. Custom initscripts run outside of the network init script lead to unpredictable results during a post-boot network service restart.

> **Note**
>
> Changing speed or duplex settings almost always requires disabling autonegotiation with the **`autoneg off`** option. This needs to be stated first, as the option entries are order-dependent.

`GATEWAY=<address>`

where `<address>` is the IP address of the network router or gateway device (if any).

`HWADDR=<MAC-address>`

where `<MAC-address>` is the hardware address of the Ethernet device in the form `AA:BB:CC:DD:EE:FF`. This directive must be used in machines containing more than one NIC to ensure that the interfaces are assigned the correct device names regardless of the configured load order for each NIC's module. This directive should **not** be used in conjunction with **`MACADDR`**.

`IPADDR=<address>`
where **`<address>`** is the IP address.

`MACADDR=<MAC-address>`
where *`<MAC-address>`* is the hardware address of the Ethernet device in the form *`AA:BB:CC:DD:EE:FF`*. This directive is used to assign a MAC address to an interface, overriding the one assigned to the physical NIC. This directive should **not** be used in conjunction with **`HWADDR`**.

`MASTER=<bond-interface>`
where **`<bond-interface>`** is the channel bonding interface to which the Ethernet interface is linked.

This directive is used in conjunction with the **`SLAVE`** directive.

Refer to *Section 4.2.3, “Channel Bonding Interfaces”* for more information about channel bonding interfaces.

`NETMASK=<mask>`
where **`<mask>`** is the netmask value.

`NETWORK=<address>`
where **`<address>`** is the network address. This directive is deprecated, as the value is calculated automatically with **`ifcalc`**.

`ONBOOT=<answer>`
where **`<answer>`** is one of the following:

- **`yes`** — This device should be activated at boot-time.
- **`no`** — This device should not be activated at boot-time.

`PEERDNS=<answer>`
where **`<answer>`** is one of the following:

- **`yes`** — Modify **`/etc/resolv.conf`** if the DNS directive is set. If using DHCP, then **`yes`** is the default.
- **`no`** — Do not modify **`/etc/resolv.conf`**.

`SLAVE=<bond-interface>`
where **`<bond-interface>`** is one of the following:

- **`yes`** — This device is controlled by the channel bonding interface specified in the **`MASTER`** directive.
- **`no`** — This device is *not* controlled by the channel bonding interface specified in the **`MASTER`** directive.

This directive is used in conjunction with the **`MASTER`** directive.

Refer to *Section 4.2.3, “Channel Bonding Interfaces”* for more about channel bonding interfaces.

SRCADDR=*<address>*
where ***<address>*** is the specified source IP address for outgoing packets.

USERCTL=*<answer>*
where ***<answer>*** is one of the following:

- **yes** — Non-root users are allowed to control this device.
- **no** — Non-root users are not allowed to control this device.

4.2.2. IPsec Interfaces

The following example shows the **ifcfg** file for a network-to-network IPsec connection for LAN A. The unique name to identify the connection in this example is **ipsec1**, so the resulting file is named **/etc/sysconfig/network-scripts/ifcfg-ipsec1**.

```
TYPE=IPsec
ONBOOT=yes
IKE_METHOD=PSK
SRCNET=192.168.1.0/24
DSTNET=192.168.2.0/24
DST=X.X.X.X
```

In the example above, *X.X.X.X* is the publicly routable IP address of the destination IPsec router.

Below is a listing of the configurable parameters for an IPsec interface:

DST=*<address>*
where <*address*> is the IP address of the IPsec destination host or router. This is used for both host-to-host and network-to-network IPsec configurations.

DSTNET=*<network>*
where <*network*> is the network address of the IPsec destination network. This is only used for network-to-network IPsec configurations.

SRC=*<address>*
where <*address*> is the IP address of the IPsec source host or router. This setting is optional and is only used for host-to-host IPsec configurations.

SRCNET=*<network>*
where <*network*> is the network address of the IPsec source network. This is only used for network-to-network IPsec configurations.

TYPE=*<interface-type>*
where <*interface-type*> is **IPSEC**. Both applications are part of the **ipsec-tools** package.

If manual key encryption with IPsec is being used, refer to **/usr/share/doc/initscripts-*<version-number>*/sysconfig.txt** (replace <*version-number*> with the version of the **initscripts** package installed) for configuration parameters.

The **racoon** IKEv1 key management daemon negotiates and configures a set of parameters for IPSec. It can use preshared keys, RSA signatures, or GSS-API. If **racoon** is used to automatically manage key encryption, the following options are required:

IKE_METHOD=<*encryption-method*>

where <*encryption-method*> is either **PSK**, **X509**, or **GSSAPI**. If **PSK** is specified, the **IKE_PSK** parameter must also be set. If **X509** is specified, the **IKE_CERTFILE** parameter must also be set.

IKE_PSK=<*shared-key*>

where <*shared-key*> is the shared, secret value for the PSK (preshared keys) method.

IKE_CERTFILE=<*cert-file*>

where <*cert-file*> is a valid **X.509** certificate file for the host.

IKE_PEER_CERTFILE=<*cert-file*>

where <*cert-file*> is a valid **X.509** certificate file for the *remote* host.

IKE_DNSSEC=<*answer*>

where <*answer*> is **yes**. The **racoon** daemon retrieves the remote host's **X.509** certificate via DNS. If a **IKE_PEER_CERTFILE** is specified, do *not* include this parameter.

For more information about the encryption algorithms available for IPsec, refer to the **setkey** man page. For more information about **racoon**, refer to the **racoon** and **racoon.conf** man pages.

4.2.3. Channel Bonding Interfaces

Fedora allows administrators to bind multiple network interfaces together into a single channel using the **bonding** kernel module and a special network interface called a *channel bonding interface*. Channel bonding enables two or more network interfaces to act as one, simultaneously increasing the bandwidth and providing redundancy.

To create a channel bonding interface, create a file in the **/etc/sysconfig/network-scripts/** directory called **ifcfg-bond<*N*>**, replacing <*N*> with the number for the interface, such as **0**.

The contents of the file can be identical to whatever type of interface is getting bonded, such as an Ethernet interface. The only difference is that the **DEVICE=** directive must be **bond<*N*>**, replacing <*N*> with the number for the interface.

The following is a sample channel bonding configuration file:

```
DEVICE=bond0
BONDING_OPTS="mode=1 miimon=500"
BOOTPROTO=none
ONBOOT=yes
NETWORK=10.0.1.0
NETMASK=255.255.255.0
IPADDR=10.0.1.27
USERCTL=no
```

After the channel bonding interface is created, the network interfaces to be bound together must be configured by adding the **MASTER=** and **SLAVE=** directives to their configuration files. The configuration files for each of the channel-bonded interfaces can be nearly identical.

For example, if two Ethernet interfaces are being channel bonded, both **eth0** and **eth1** may look like the following example:

```
DEVICE=eth<N>
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
USERCTL=no
```

In this example, replace <*N*> with the numerical value for the interface.

> **Important**
>
> Important aspects of the channel bonding interface are controlled through the kernel module. For more information about controlling the **`bonding`** modules, refer to *Section 30.5.2, “The Channel Bonding Module”*.

4.2.4. Alias and Clone Files

Two lesser-used types of interface configuration files are *alias* and *clone* files.

Alias interface configuration files, which are used to bind multiple addresses to a single interface, use the **`ifcfg-<if-name>:<alias-value>`** naming scheme.

For example, an **`ifcfg-eth0:0`** file could be configured to specify `DEVICE=eth0:0` and a static IP address of 10.0.0.2, serving as an alias of an Ethernet interface already configured to receive its IP information via DHCP in **`ifcfg-eth0`**. Under this configuration, **`eth0`** is bound to a dynamic IP address, but the same physical network card can receive requests via the fixed, 10.0.0.2 IP address.

> **Caution**
>
> Alias interfaces do not support DHCP.

A clone interface configuration file should use the following naming convention: **`ifcfg-<if-name>-<clone-name>`**. While an alias file allows multiple addresses for an existing interface, a clone file is used to specify additional options for an interface. For example, a standard DHCP Ethernet interface called **`eth0`**, may look similar to this:

```
DEVICE=eth0 ONBOOT=yes BOOTPROTO=dhcp
```

Since the default value for the **`USERCTL`** directive is **`no`** if it is not specified, users cannot bring this interface up and down. To give users the ability to control the interface, create a clone by copying **`ifcfg-eth0`** to **`ifcfg-eth0-user`** and add the following line to **`ifcfg-eth0-user`**:

```
USERCTL=yes
```

This way a user can bring up the **`eth0`** interface using the **`/sbin/ifup eth0-user`** command because the configuration options from **`ifcfg-eth0`** and **`ifcfg-eth0-user`** are combined. While this is a very basic example, this method can be used with a variety of options and interfaces.

The easiest way to create alias and clone interface configuration files is to use the graphical **Network Administration Tool**. For more information on using this tool, refer to *Chapter 5, Network Configuration*.

4.2.5. Dialup Interfaces

If you are connecting to the Internet via a dialup connection, a configuration file is necessary for the interface.

PPP interface files are named using the following format:

`ifcfg-ppp<X>`
where *<X>* is a unique number corresponding to a specific interface.

The PPP interface configuration file is created automatically when **`wvdial`**, the **Network Administration Tool** or **Kppp** is used to create a dialup account. It is also possible to create and edit this file manually.

The following is a typical **`ifcfg-ppp0`** file:

```
DEVICE=ppp0
NAME=test
WVDIALSECT=test
MODEMPORT=/dev/modem
LINESPEED=115200
PAPNAME=test
USERCTL=true
ONBOOT=no
PERSIST=no
DEFROUTE=yes
PEERDNS=yes
DEMAND=no
IDLETIMEOUT=600
```

Serial Line Internet Protocol (SLIP) is another dialup interface, although it is used less frequently. SLIP files have interface configuration file names such as **`ifcfg-sl0`**.

Other options that may be used in these files include:

`DEFROUTE=<answer>`
where **`<answer>`** is one of the following:

- **`yes`** — Set this interface as the default route.
- **`no`** — Do not set this interface as the default route.

`DEMAND=<answer>`
where **`<answer>`** is one of the following:

- **`yes`** — This interface allows **pppd** to initiate a connection when someone attempts to use it.
- **`no`** — A connection must be manually established for this interface.

`IDLETIMEOUT=<value>`
where **`<value>`** is the number of seconds of idle activity before the interface disconnects itself.

`INITSTRING=<`*`string`*`>`

where **`<`*`string`*`>`** is the initialization string passed to the modem device. This option is primarily used in conjunction with SLIP interfaces.

`LINESPEED=<`*`value`*`>`

where **`<`*`value`*`>`** is the baud rate of the device. Possible standard values include **`57600`**, **`38400`**, **`19200`**, and **`9600`**.

`MODEMPORT=<`*`device`*`>`

where **`<`*`device`*`>`** is the name of the serial device that is used to establish the connection for the interface.

`MTU=<`*`value`*`>`

where **`<`*`value`*`>`** is the *Maximum Transfer Unit (MTU)* setting for the interface. The MTU refers to the largest number of bytes of data a frame can carry, not counting its header information. In some dialup situations, setting this to a value of **`576`** results in fewer packets dropped and a slight improvement to the throughput for a connection.

`NAME=<`*`name`*`>`

where **`<`*`name`*`>`** is the reference to the title given to a collection of dialup connection configurations.

`PAPNAME=<`*`name`*`>`

where **`<`*`name`*`>`** is the username given during the *Password Authentication Protocol (PAP)* exchange that occurs to allow connections to a remote system.

`PERSIST=<`*`answer`*`>`

where **`<`*`answer`*`>`** is one of the following:

- **`yes`** — This interface should be kept active at all times, even if deactivated after a modem hang up.

- **`no`** — This interface should not be kept active at all times.

`REMIP=<`*`address`*`>`

where **`<`*`address`*`>`** is the IP address of the remote system. This is usually left unspecified.

`WVDIALSECT=<`*`name`*`>`

where **`<`*`name`*`>`** associates this interface with a dialer configuration in **`/etc/wvdial.conf`**. This file contains the phone number to be dialed and other important information for the interface.

4.2.6. Other Interfaces

Other common interface configuration files include the following:

`ifcfg-lo`

A local *loopback interface* is often used in testing, as well as being used in a variety of applications that require an IP address pointing back to the same system. Any data sent to the loopback device is immediately returned to the host's network layer.

> **Warning**
>
> The loopback interface script, **`/etc/sysconfig/network-scripts/ifcfg-lo`**, should never be edited manually. Doing so can prevent the system from operating correctly.

`ifcfg-irlan0`
: An *infrared interface* allows information between devices, such as a laptop and a printer, to flow over an infrared link. This works in a similar way to an Ethernet device except that it commonly occurs over a peer-to-peer connection.

`ifcfg-plip0`
: A *Parallel Line Interface Protocol (PLIP)* connection works much the same way as an Ethernet device, except that it utilizes a parallel port.

`ifcfg-tr0`
: *Token Ring* topologies are not as common on *Local Area Networks* (*LANs*) as they once were, having been eclipsed by Ethernet.

4.3. Interface Control Scripts

The interface control scripts activate and deactivated system interfaces. There are two primary interface control scripts that call on control scripts located in the **`/etc/sysconfig/network-scripts/`** directory: **`/sbin/ifdown`** and **`/sbin/ifup`**.

The **`ifup`** and **`ifdown`** interface scripts are symbolic links to scripts in the **`/sbin/`** directory. When either of these scripts are called, they require the value of the interface to be specified, such as:

```
ifup eth0
```

> **Caution**
>
> The **`ifup`** and **`ifdown`** interface scripts are the only scripts that the user should use to bring up and take down network interfaces.
>
> The following scripts are described for reference purposes only.

Two files used to perform a variety of network initialization tasks during the process of bringing up a network interface are **`/etc/rc.d/init.d/functions`** and **`/etc/sysconfig/network-scripts/network-functions`**. Refer to *Section 4.5, “Network Function Files”* for more information.

After verifying that an interface has been specified and that the user executing the request is allowed to control the interface, the correct script brings the interface up or down. The following are common interface control scripts found within the **`/etc/sysconfig/network-scripts/`** directory:

`ifup-aliases`
: Configures IP aliases from interface configuration files when more than one IP address is associated with an interface.

`ifup-ippp` and **`ifdown-ippp`**
: Brings ISDN interfaces up and down.

`ifup-ipsec` and **`ifdown-ipsec`**
: Brings IPsec interfaces up and down.

`ifup-ipv6` and **`ifdown-ipv6`**
: Brings IPv6 interfaces up and down.

`ifup-ipx`
: Brings up an IPX interface.

`ifup-plip`
: Brings up a PLIP interface.

`ifup-plusb`
: Brings up a USB interface for network connections.

`ifup-post` and **`ifdown-post`**
: Contains commands to be executed after an interface is brought up or down.

`ifup-ppp` and **`ifdown-ppp`**
: Brings a PPP interface up or down.

`ifup-routes`
: Adds static routes for a device as its interface is brought up.

`ifdown-sit` and **`ifup-sit`**
: Contains function calls related to bringing up and down an IPv6 tunnel within an IPv4 connection.

`ifup-sl` and **`ifdown-sl`**
: Brings a SLIP interface up or down.

`ifup-wireless`
: Brings up a wireless interface.

> **Warning**
>
> Removing or modifying any scripts in the **`/etc/sysconfig/network-scripts/`** directory can cause interface connections to act irregularly or fail. Only advanced users should modify scripts related to a network interface.

The easiest way to manipulate all network scripts simultaneously is to use the **`/sbin/service`** command on the network service (**`/etc/rc.d/init.d/network`**), as illustrated the following command:

```
/sbin/service network <action>
```

Here, *`<action>`* can be either **`start`**, **`stop`**, or **`restart`**.

To view a list of configured devices and currently active network interfaces, use the following command:

```
/sbin/service network status
```

4.4. Configuring Static Routes

Routing will be configured on routing devices, therefore it should not be necessary to configure static routes on Fedora servers or clients. However, if static routes are required they can be configured for each interface. This can be useful if you have multiple interfaces in different subnets. Use the **`route`** command to display the IP routing table.

Static route configuration is stored in a **`/etc/sysconfig/network-scripts/route-`*`interface`*** file. For example, static routes for the eth0 interface would be stored in the **`/etc/sysconfig/network-scripts/route-eth0`** file. The **`route-`*`interface`*** file has two formats: IP command arguments and network/netmask directives.

IP Command Arguments Format

Define a default gateway on the first line. This is only required if the default gateway is not set via DHCP:

```
default X.X.X.X dev interface
```

`X.X.X.X` is the IP address of the default gateway. The *`interface`* is the interface that is connected to, or can reach, the default gateway.

Define a static route. Each line is parsed as an individual route:

```
X.X.X.X/X via X.X.X.X dev interface
```

`X.X.X.X/X` is the network number and netmask for the static route. *`X.X.X.X`* and *`interface`* are the IP address and interface for the default gateway respectively. The *`X.X.X.X`* address does not have to be the default gateway IP address. In most cases, *`X.X.X.X`* will be an IP address in a different subnet, and *`interface`* will be the interface that is connected to, or can reach, that subnet. Add as many static routes as required.

The following is a sample **`route-eth0`** file using the IP command arguments format. The default gateway is 192.168.0.1, interface eth0. The two static routes are for the 10.10.10.0/24 and 172.16.1.0/24 networks:

```
default 192.168.0.1 dev eth0
10.10.10.0/24 via 192.168.0.1 dev eth0
172.16.1.0/24 via 192.168.0.1 dev eth0
```

Static routes should only be configured for other subnets. The above example is not necessary, since packets going to the 10.10.10.0/24 and 172.16.1.0/24 networks will use the default gateway anyway. Below is an example of setting static routes to a different subnet, on a machine in a 192.168.0.0/24 subnet. The example machine has an eth0 interface in the 192.168.0.0/24 subnet, and an eth1 interface (10.10.10.1) in the 10.10.10.0/24 subnet:

```
10.10.10.0/24 via 10.10.10.1 dev eth1
```

Duplicate Default Gateways

If the default gateway is already assigned from DHCP, the IP command arguments format can cause one of two errors during start-up, or when bringing up an interface from the down state using the **`ifup`** command: "RTNETLINK answers: File exists" or 'Error: either "to" is a duplicate, or "*`X.X.X.X`*" is a garbage.', where *`X.X.X.X`* is the gateway, or a different IP address. These errors can also occur if you have another route to another network using the default gateway. Both of these errors are safe to ignore.

Network/Netmask Directives Format

You can also use the network/netmask directives format for **`route-`*`interface`*** files. The following is a template for the network/netmask format, with instructions following afterwards:

```
ADDRESS0=X.X.X.X
NETMASK0=X.X.X.X
GATEWAY0=X.X.X.X
```

- `ADDRESS0=`*`X.X.X.X`* is the network number for the static route.
- `NETMASK0=`*`X.X.X.X`* is the netmask for the network number defined with `ADDRESS0=`*`X.X.X.X`*.
- `GATEWAY0=`*`X.X.X.X`* is the default gateway, or an IP address that can be used to reach `ADDRESS0=`*`X.X.X.X`*

The following is a sample **`route-eth0`** file using the network/netmask directives format. The default gateway is 192.168.0.1, interface eth0. The two static routes are for the 10.10.10.0/24 and 172.16.1.0/24 networks. However, as mentioned before, this example is not necessary as the 10.10.10.0/24 and 172.16.1.0/24 networks would use the default gateway anyway:

```
ADDRESS0=10.10.10.0
NETMASK0=255.255.255.0
GATEWAY0=192.168.0.1
ADDRESS1=172.16.1.0
NETMASK1=255.255.255.0
GATEWAY1=192.168.0.1
```

Subsequent static routes must be numbered sequentially, and must not skip any values. For example, `ADDRESS0`, `ADDRESS1`, `ADDRESS2`, and so on.

Below is an example of setting static routes to a different subnet, on a machine in the 192.168.0.0/24 subnet. The example machine has an eth0 interface in the 192.168.0.0/24 subnet, and an eth1 interface (10.10.10.1) in the 10.10.10.0/24 subnet:

```
ADDRESS0=10.10.10.0
NETMASK0=255.255.255.0
GATEWAY0=10.10.10.1
```

DHCP should assign these settings automatically, therefore it should not be necessary to configure static routes on Fedora servers or clients.

4.5. Network Function Files

Fedora makes use of several files that contain important common functions used to bring interfaces up and down. Rather than forcing each interface control file to contain these functions, they are grouped together in a few files that are called upon when necessary.

The **`/etc/sysconfig/network-scripts/network-functions`** file contains the most commonly used IPv4 functions, which are useful to many interface control scripts. These functions include contacting running programs that have requested information about changes in the status of an interface, setting hostnames, finding a gateway device, verifying whether or not a particular device is down, and adding a default route.

As the functions required for IPv6 interfaces are different from IPv4 interfaces, a **`/etc/sysconfig/network-scripts/network-functions-ipv6`** file exists specifically to hold this information. The functions in this file configure and delete static IPv6 routes, create and remove tunnels, add and remove IPv6 addresses to an interface, and test for the existence of an IPv6 address on an interface.

4.6. Additional Resources

The following are resources which explain more about network interfaces.

4.6.1. Installed Documentation

`/usr/share/doc/initscripts-<`*`version`*`>/sysconfig.txt`

A guide to available options for network configuration files, including IPv6 options not covered in this chapter.

`/usr/share/doc/iproute-<`*`version`*`>/ip-cref.ps`

This file contains a wealth of information about the **`ip`** command, which can be used to manipulate routing tables, among other things. Use the **ggv** or **kghostview** application to view this file.

Network Configuration

To communicate with each other, computers must have a network connection. This is accomplished by having the operating system recognize an interface card (such as Ethernet, ISDN modem, or token ring) and configuring the interface to connect to the network.

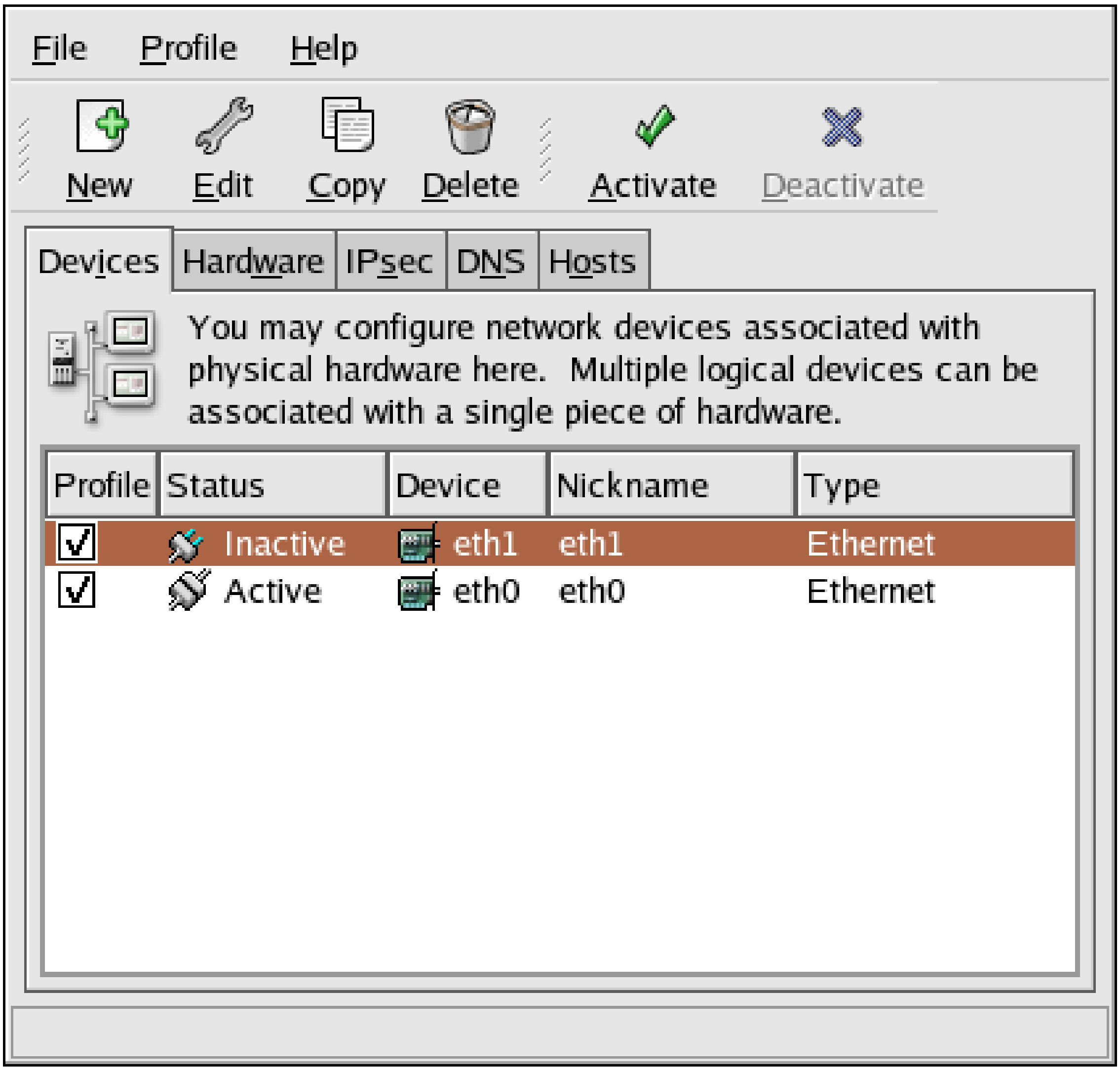
The **Network Administration Tool** can be used to configure the following types of network interfaces:

- Ethernet
- ISDN
- modem
- xDSL
- token ring
- CIPE
- wireless devices

It can also be used to configure IPsec connections, manage DNS settings, and manage the **`/etc/hosts`** file used to store additional hostnames and IP address combinations.

To use the **Network Administration Tool**, you must have root privileges. To start the application, go to the Applications (the main menu on the panel) > **System Settings** > **Network**, or type the command **`system-config-network`** at a shell prompt (for example, in an **XTerm** or a **GNOME terminal**). If you type the command, the graphical version is displayed if **X** is running; otherwise, the text-based version is displayed.

To use the command line version, execute the command **`system-config-network-cmd --help`** as root to view all of the options.



You may configure network devices associated with physical hardware here. Multiple logical devices can be associated with a single piece of hardware.

Profile	Status	Device	Nickname	Type
☑	Inactive	eth1	eth1	Ethernet
☑	Active	eth0	eth0	Ethernet

Figure 5.1. **Network Administration Tool**

Tip

Use the Red Hat Hardware Compatibility List (*http://hardware.redhat.com/hcl/*) to determine if Fedora supports your hardware device.

5.1. Overview

To configure a network connection with the **Network Administration Tool**, perform the following steps:

1. Add a network device associated with the physical hardware device.
2. Add the physical hardware device to the hardware list, if it does not already exist.
3. Configure the hostname and DNS settings.
4. Configure any hosts that cannot be looked up through DNS.

This chapter discusses each of these steps for each type of network connection.

5.2. Establishing an Ethernet Connection

To establish an Ethernet connection, you need a network interface card (NIC), a network cable (usually a CAT5 cable), and a network to connect to. Different networks are configured to use different network speeds; make sure your NIC is compatible with the network to which you want to connect.

To add an Ethernet connection, follow these steps:

1. Click the **Devices** tab.

2. Click the **New** button on the toolbar.

3. Select **Ethernet connection** from the **Device Type** list, and click **Forward**.

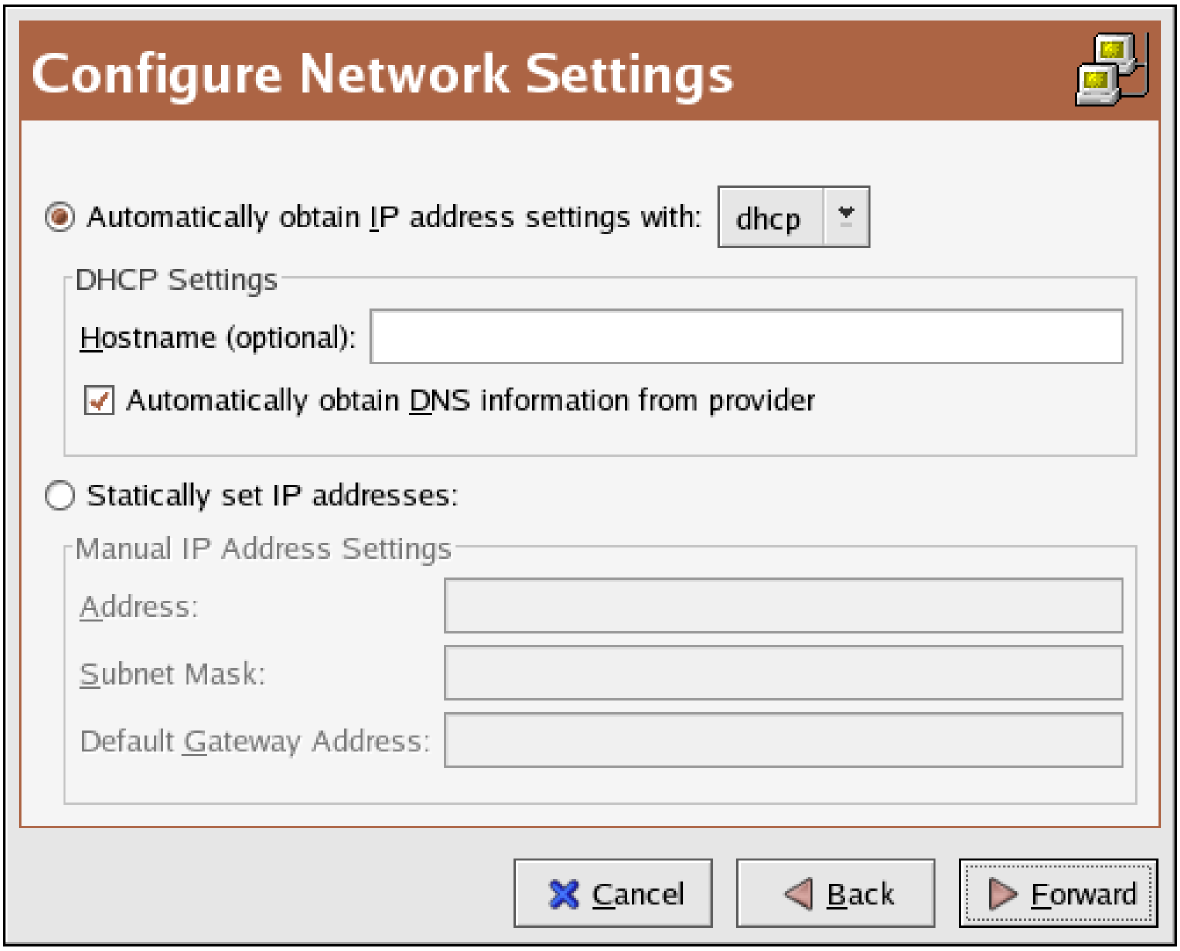
4. If you have already added the network interface card to the hardware list, select it from the **Ethernet card** list. Otherwise, select **Other Ethernet Card** to add the hardware device.

   > **Note**
   >
   > The installation program detects supported Ethernet devices and prompts you to configure them. If you configured any Ethernet devices during the installation, they are displayed in the hardware list on the **Hardware** tab.

5. If you selected **Other Ethernet Card**, the **Select Ethernet Adapter** window appears. Select the manufacturer and model of the Ethernet card. Select the device name. If this is the system's first Ethernet card, select **eth0** as the device name; if this is the second Ethernet card, select **eth1** (and so on). The **Network Administration Tool** also allows you to configure the resources for the NIC. Click **Forward** to continue.

6. In the **Configure Network Settings** window shown in *Figure 5.2, “Ethernet Settings”*, choose between DHCP and a static IP address. If the device receives a different IP address each time the network is started, do not specify a hostname. Click **Forward** to continue.

7. Click **Apply** on the **Create Ethernet Device** page.



Configure Network Settings

● Automatically obtain IP address settings with: dhcp

DHCP Settings

Hostname (optional):

☑ Automatically obtain DNS information from provider

○ Statically set IP addresses:

Manual IP Address Settings

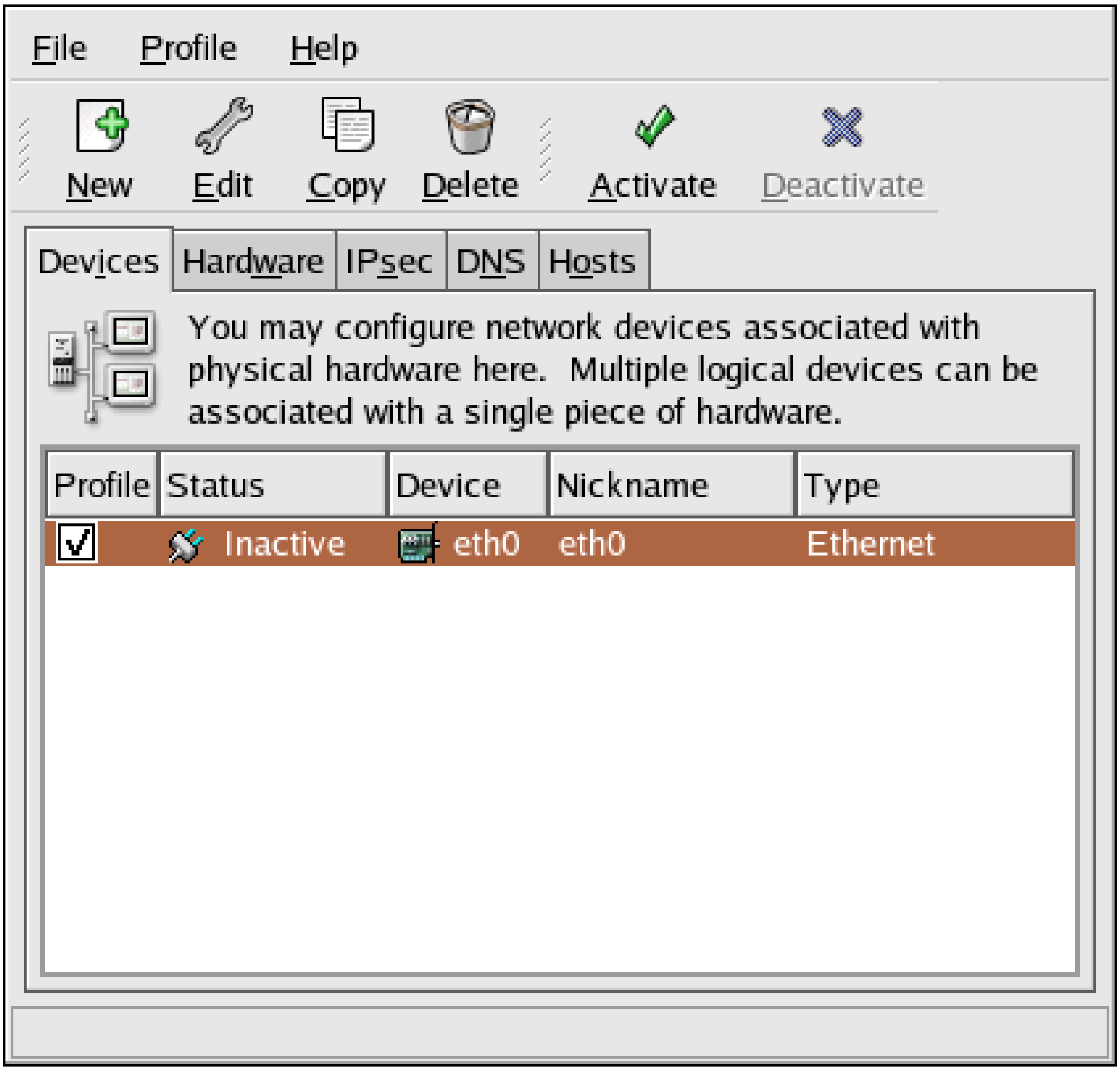
Address:

Subnet Mask:

Default Gateway Address:

Figure 5.2. Ethernet Settings

After configuring the Ethernet device, it appears in the device list as shown in *Figure 5.3, “Ethernet Device”*.



You may configure network devices associated with physical hardware here. Multiple logical devices can be associated with a single piece of hardware.

Profile	Status	Device	Nickname	Type
☑	Inactive	eth0	eth0	Ethernet

Figure 5.3. Ethernet Device

Be sure to select **File** > **Save** to save the changes.

After adding the Ethernet device, you can edit its configuration by selecting the device from the device list and clicking **Edit**. For example, when the device is added, it is configured to start at boot time by default. To change this setting, select to edit the device, modify the **Activate device when computer starts** value, and save the changes.

When the device is added, it is not activated immediately, as seen by its **Inactive** status. To activate the device, select it from the device list, and click the **Activate** button. If the system is configured to activate the device when the computer starts (the default), this step does not have to be performed again.

If you associate more than one device with an Ethernet card, the subsequent devices are *device aliases*. A device alias allows you to setup multiple virtual devices for one physical device, thus giving the one physical device more than one IP address. For example, you can configure an eth1 device and an eth1:1 device. For details, refer to *Section 5.11, “Device Aliases”*.

5.3. Establishing an ISDN Connection

An ISDN connection is an Internet connection established with a ISDN modem card through a special phone line installed by the phone company. ISDN connections are popular in Europe.

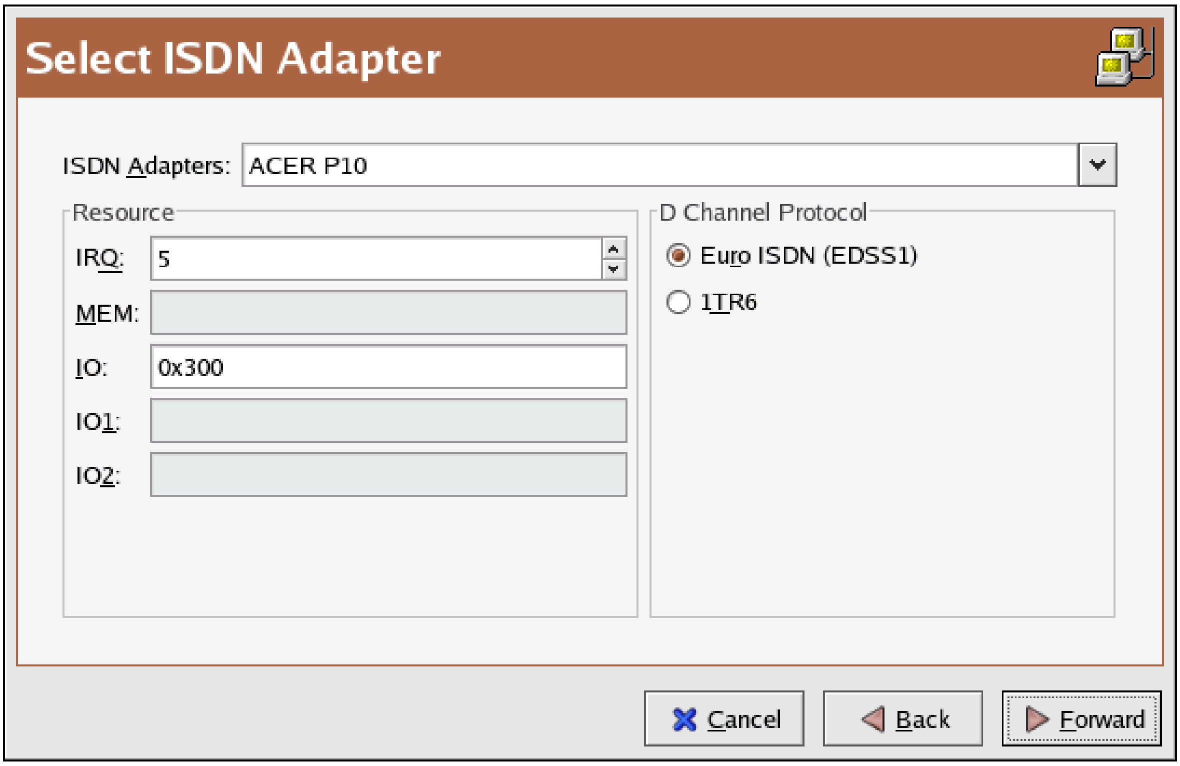
To add an ISDN connection, follow these steps:

1. Click the **Devices** tab.

2. Click the **New** button on the toolbar.

3. Select **ISDN connection** from the **Device Type** list, and click **Forward**.

4. Select the ISDN adapter from the pulldown menu. Then configure the resources and D channel protocol for the adapter. Click **Forward** to continue.



Select ISDN Adapter

ISDN Adapters: ACER P10

Resource

IRQ: 5

MEM:

IO: 0x300

IO1:

IO2:

D Channel Protocol

● Euro ISDN (EDSS1)

○ 1TR6

Figure 5.4. ISDN Settings

5. If your Internet Service Provider (ISP) is in the pre-configured list, select it. Otherwise, enter the required information about your ISP account. If you do not know the values, contact your ISP. Click **Forward**.

6. In the **IP Settings** window, select the **Encapsulation Mode** and whether to obtain an IP address automatically or to set a static IP instead. Click **Forward** when finished.

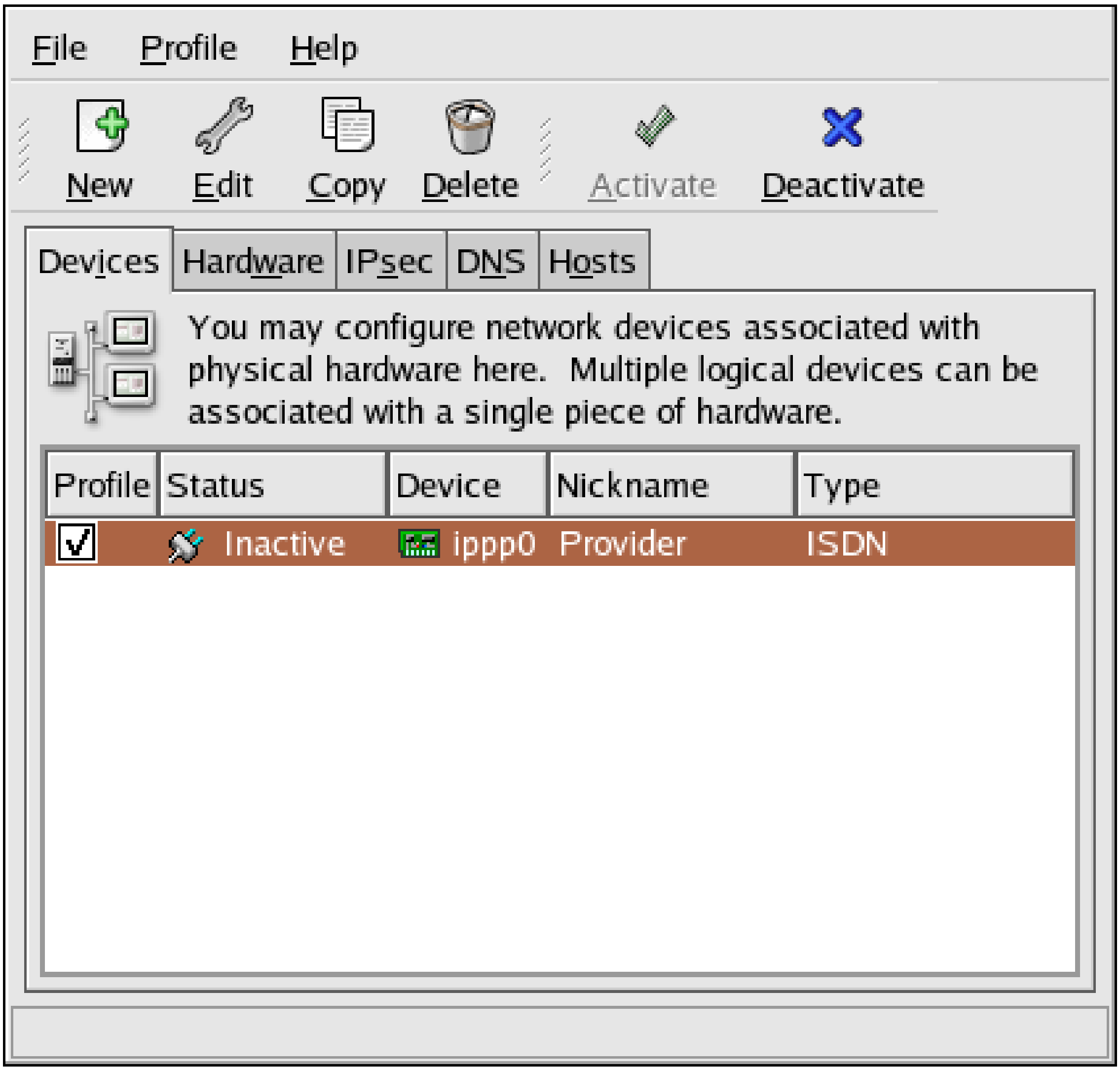
7. On the **Create Dialup Connection** page, click **Apply**.

After configuring the ISDN device, it appears in the device list as a device with type **ISDN** as shown in *Figure 5.5, “ISDN Device”*.

Be sure to select **File** > **Save** to save the changes.

After adding the ISDN device, you can edit its configuration by selecting the device from the device list and clicking **Edit**. For example, when the device is added, it is configured not to start at boot time by default. Edit its configuration to modify this setting. Compression, PPP options, login name, password, and more can be changed.

When the device is added, it is not activated immediately, as seen by its **Inactive** status. To activate the device, select it from the device list, and click the **Activate** button. If the system is configured to activate the device when the computer starts (the default), this step does not have to be performed again.



You may configure network devices associated with physical hardware here. Multiple logical devices can be associated with a single piece of hardware.

Profile	Status	Device	Nickname	Type
☑	Inactive	ippp0	Provider	ISDN

Figure 5.5. ISDN Device

5.4. Establishing a Modem Connection

A modem can be used to configure an Internet connection over an active phone line. An Internet Service Provider (ISP) account (also called a dial-up account) is required.

To add a modem connection, follow these steps:

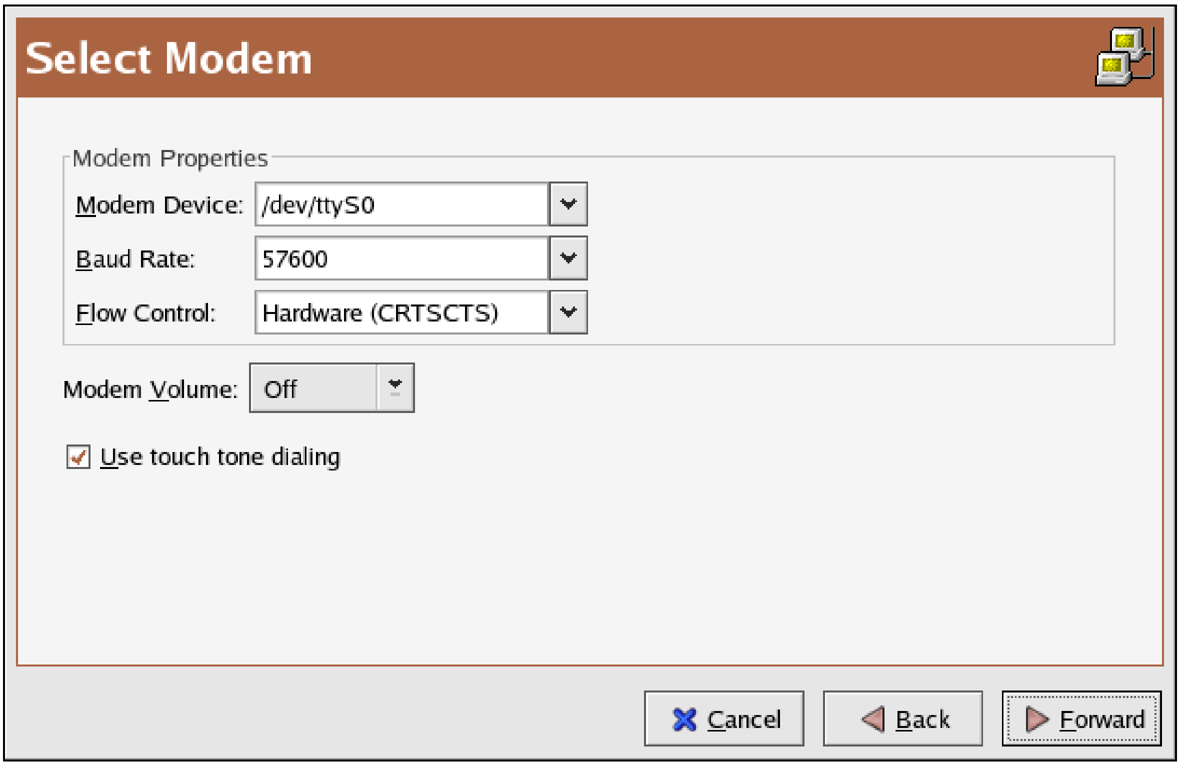
1. Click the **Devices** tab.

2. Click the **New** button on the toolbar.

3. Select **Modem connection** from the **Device Type** list, and click **Forward**.

4. If there is a modem already configured in the hardware list (on the **Hardware** tab), the **Network Administration Tool** assumes you want to use it to establish a modem connection. If there are no modems already configured, it tries to detect any modems in the system. This probe might take a while. If a modem is not found, a message is displayed to warn you that the settings shown are not values found from the probe.

5. After probing, the window in *Figure 5.6, "Modem Settings"* appears.



Select Modem

Modem Properties

Modem Device: /dev/ttyS0

Baud Rate: 57600

Flow Control: Hardware (CRTSCTS)

Modem Volume: Off

☑ Use touch tone dialing

Figure 5.6. Modem Settings

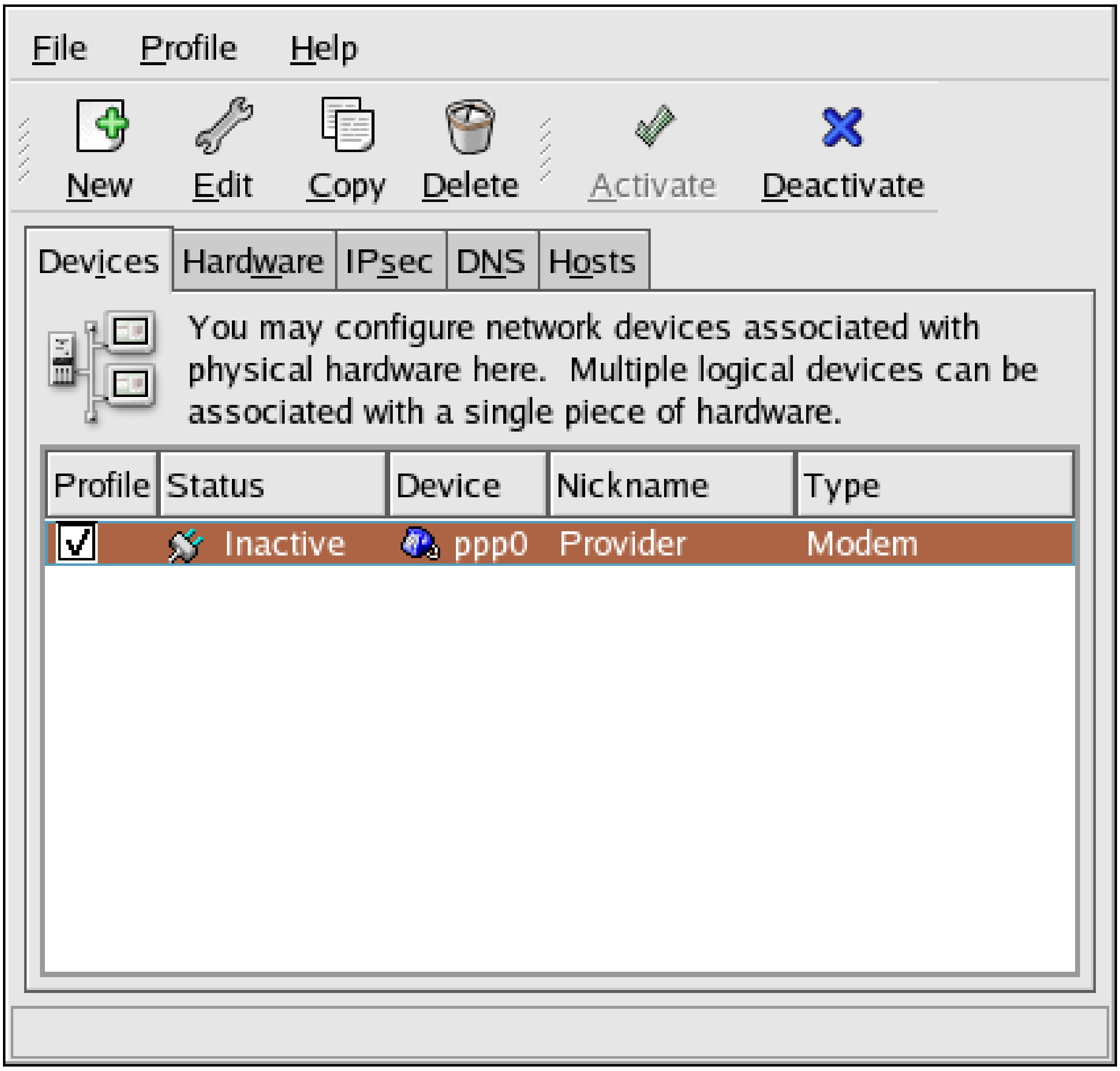
6. Configure the modem device, baud rate, flow control, and modem volume. If you do not know these values, accept the defaults if the modem was probed successfully. If you do not have touch tone dialing, uncheck the corresponding checkbox. Click **Forward**.

7. If your ISP is in the pre-configured list, select it. Otherwise, enter the required information about your ISP account. If you do not know these values, contact your ISP. Click **Forward**.

8. On the **IP Settings** page, select whether to obtain an IP address automatically or whether to set one statically. Click **Forward** when finished.

9. On the **Create Dialup Connection** page, click **Apply**.

After configuring the modem device, it appears in the device list with the type `Modem` as shown in *Figure 5.7, "Modem Device"*.



You may configure network devices associated with physical hardware here. Multiple logical devices can be associated with a single piece of hardware.

Profile	Status	Device	Nickname	Type
☑	Inactive	ppp0	Provider	Modem

Figure 5.7. Modem Device

Be sure to select **File** > **Save** to save the changes.

After adding the modem device, you can edit its configuration by selecting the device from the device list and clicking **Edit**. For example, when the device is added, it is configured not to start at boot time by default. Edit its configuration to modify this setting. Compression, PPP options, login name, password, and more can also be changed.

When the device is added, it is not activated immediately, as seen by its **Inactive** status. To activate the device, select it from the device list, and click the **Activate** button. If the system is configured to activate the device when the computer starts (the default), this step does not have to be performed again.

5.5. Establishing an xDSL Connection

DSL stands for *Digital Subscriber Lines*. There are different types of DSL such as ADSL, IDSL, and SDSL. The **Network Administration Tool** uses the term *xDSL* to mean all types of DSL connections.

Some DSL providers require that the system is configured to obtain an IP address through DHCP with an Ethernet card. Some DSL providers require you to configure a PPPoE (Point-to-Point Protocol over Ethernet) connection with an Ethernet card. Ask your DSL provider which method to use.

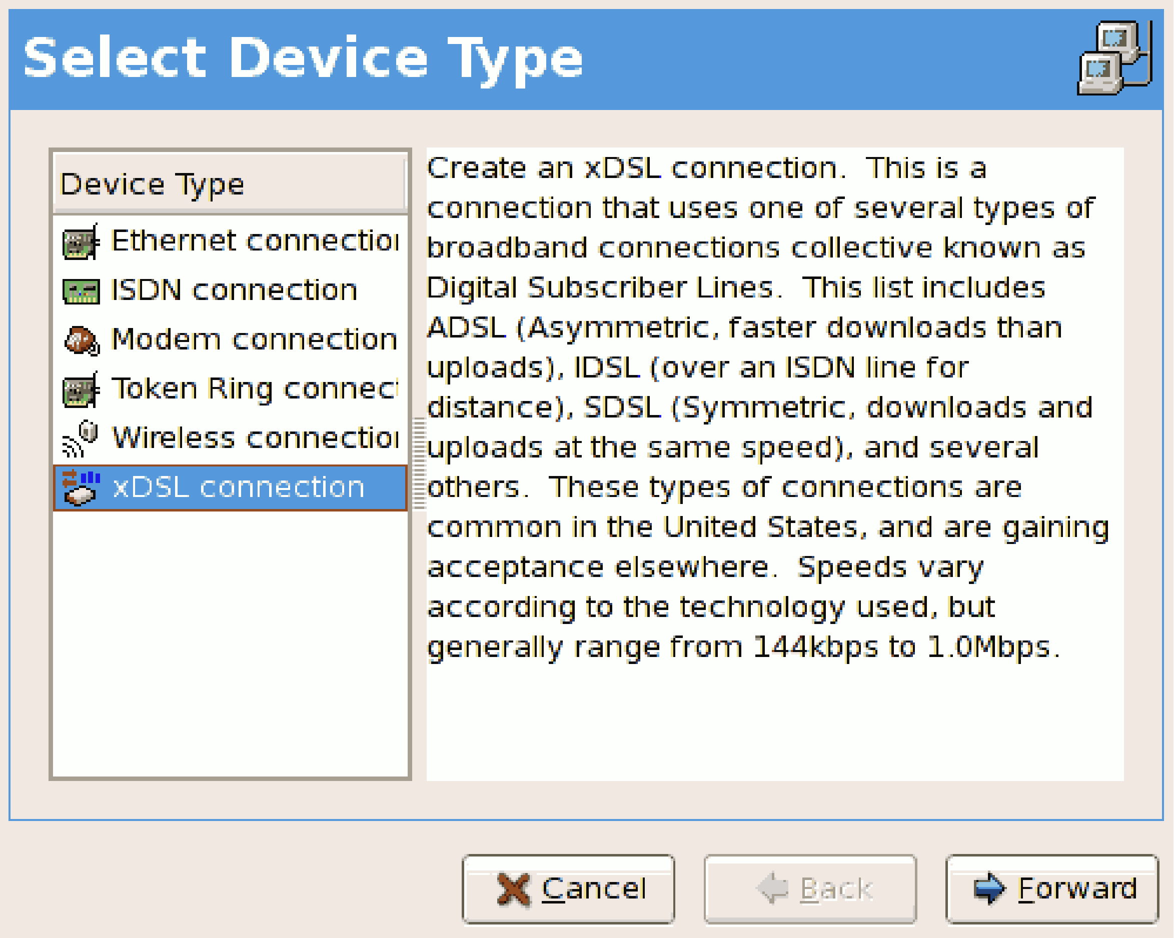
If you are required to use DHCP, refer to *Section 5.2, “Establishing an Ethernet Connection”* to configure your Ethernet card.

If you are required to use PPPoE, follow these steps:

1. Click the **Devices** tab.

2. Click the **New** button.

3. Select **xDSL connection** from the **Device Type** list, and click **Forward** as shown in *Figure 5.8, “Select Device Type”*.



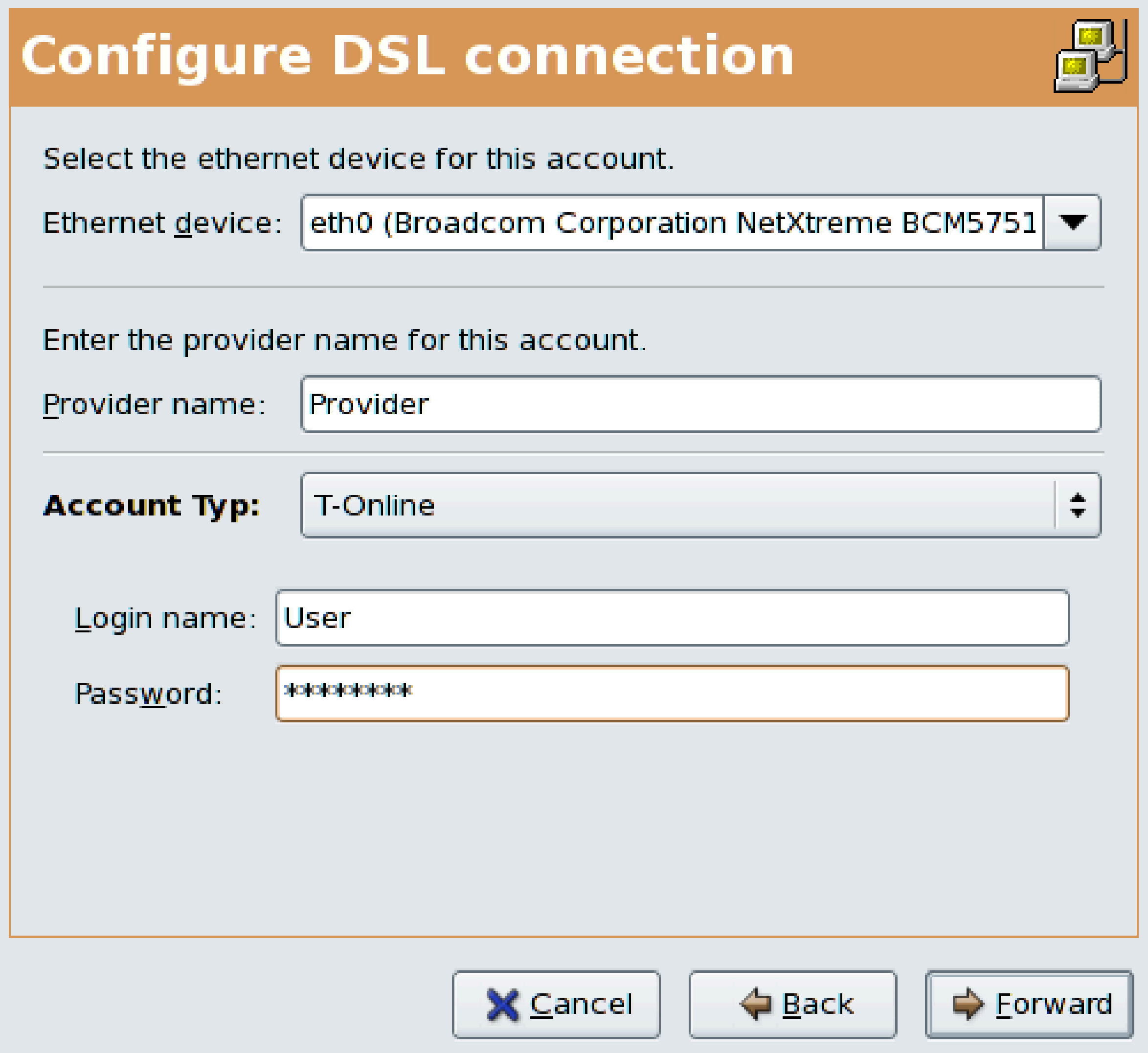
Create an xDSL connection. This is a connection that uses one of several types of broadband connections collective known as Digital Subscriber Lines. This list includes ADSL (Asymmetric, faster downloads than uploads), IDSL (over an ISDN line for distance), SDSL (Symmetric, downloads and uploads at the same speed), and several others. These types of connections are common in the United States, and are gaining acceptance elsewhere. Speeds vary according to the technology used, but generally range from 144kbps to 1.0Mbps.

Figure 5.8. Select Device Type

4. If your Ethernet card is in the hardware list, select the **Ethernet Device** from the pulldown menu from the page shown in *Figure 5.9, “xDSL Settings”*. Otherwise, the **Select Ethernet Adapter** window appears.

Note

The installation program detects supported Ethernet devices and prompts you to configure them. If you configured any Ethernet devices during the installation, they are displayed in the hardware list on the **Hardware** tab.



Configure DSL connection

Select the ethernet device for this account.

Ethernet device: eth0 (Broadcom Corporation NetXtreme BCM5751

Enter the provider name for this account.

Provider name: Provider

Account Typ: T-Online

Login name: User

Password: *********

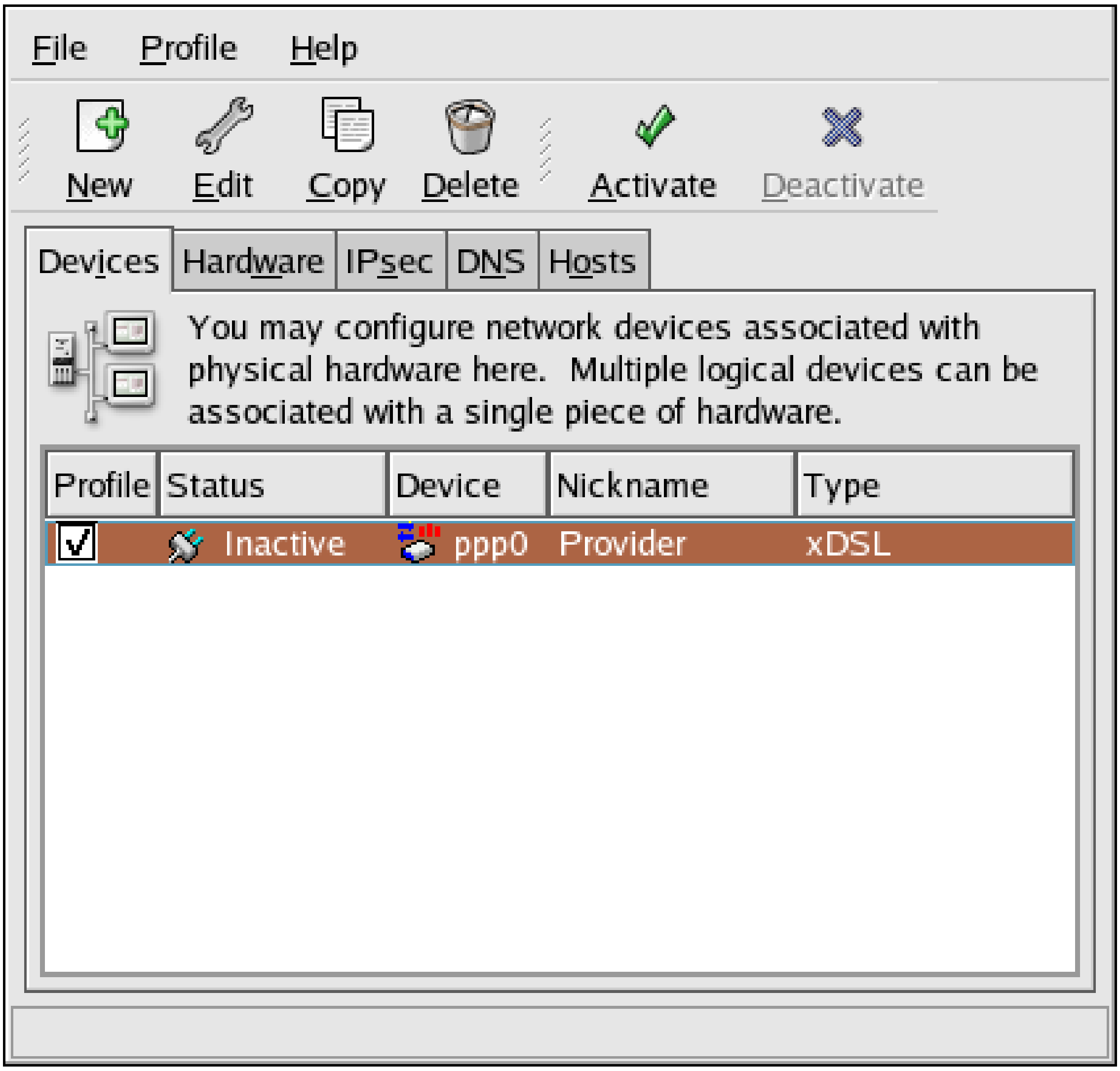
Figure 5.9. xDSL Settings

5. Enter the **Provider Name**, **Login Name**, and **Password**. If you are not setting up a T-Online account, select **Normal** from the **Account Type** pulldown menu.

   If you are setting up a T-Online account, select **T-Online** from the **Account Type** pulldown menu and enter any values in the **Login name** and **Password** field. You can further configure your T-Online account settings once the DSL connection has been fully configured (refer to *Setting Up a T-Online Account*).

6. Click the **Forward** to go to the **Create DSL Connection** menu. Check your settings and click **Apply** to finish.

7. After configuring the DSL connection, it appears in the device list as shown in *Figure 5.10, “xDSL Device”*.

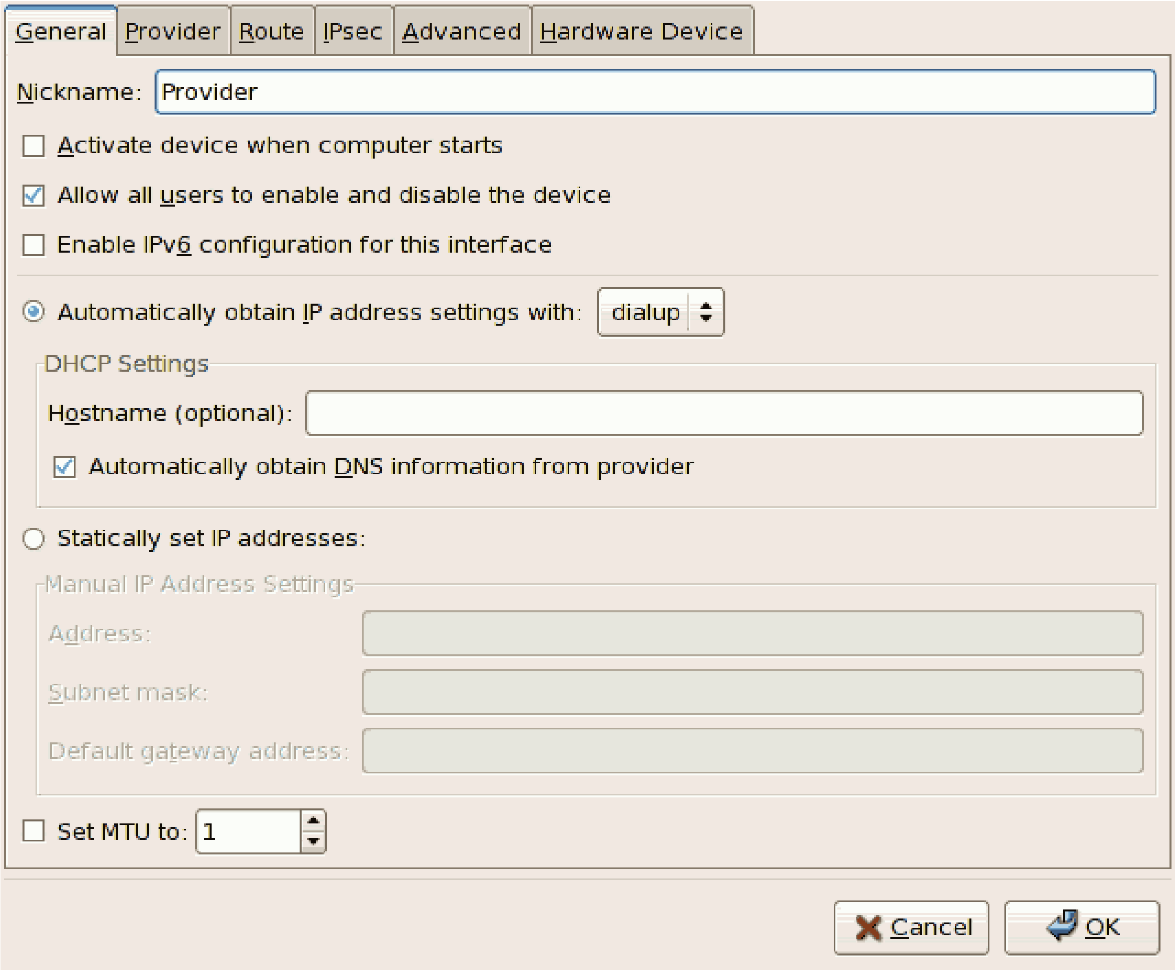


You may configure network devices associated with physical hardware here. Multiple logical devices can be associated with a single piece of hardware.

Profile	Status	Device	Nickname	Type
☑	Inactive	ppp0	Provider	xDSL

Figure 5.10. xDSL Device

8. After adding the xDSL connection, you can edit its configuration by selecting the device from the device list and clicking **Edit**.



Nickname: Provider

☐ Activate device when computer starts

☑ Allow all users to enable and disable the device

☐ Enable IPv6 configuration for this interface

● Automatically obtain IP address settings with: dialup

DHCP Settings

Hostname (optional):

☑ Automatically obtain DNS information from provider

○ Statically set IP addresses:

Manual IP Address Settings

Address:

Subnet mask:

Default gateway address:

☐ Set MTU to: 1

Figure 5.11. xDSL Configuration

For example, when the device is added, it is configured not to start at boot time by default. Edit its configuration to modify this setting. Click **OK** when finished.

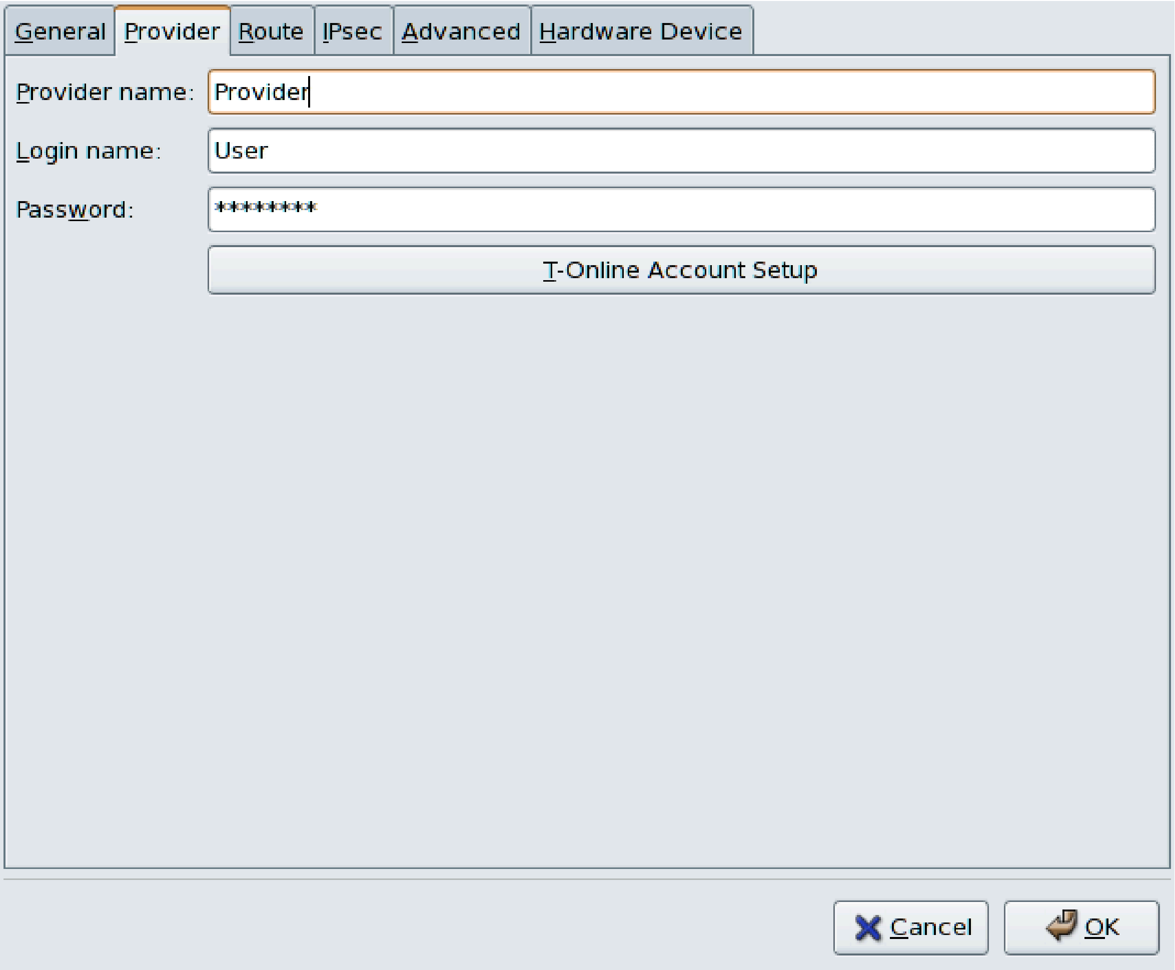
9. Once you are satisfied with your xDSL connection settings, select **File** > **Save** to save the changes.

Setting Up a T-Online Account

If you are setting up a T-Online Account, follow these additional steps:

1. Select the device from the device list and click **Edit**.

2. Select the **Provider** tab from the **xDSL Configuration** menu as shown in *Figure 5.12, “xDSL Configuration - Provider Tab”*.



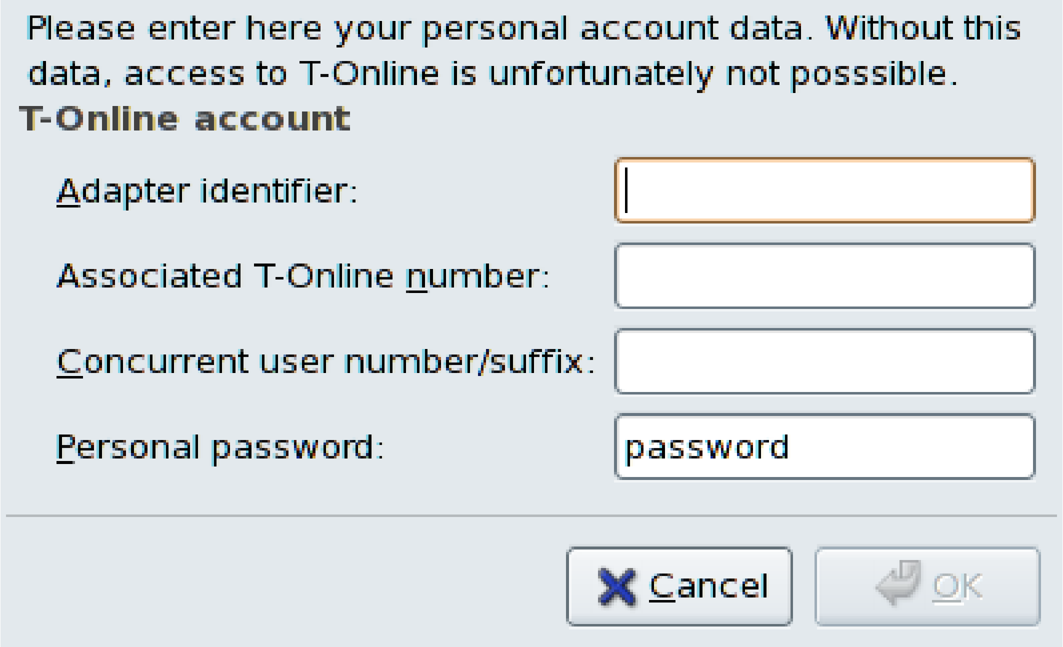
Provider name: Provider

Login name: User

Password: ********

Figure 5.12. xDSL Configuration - Provider Tab

3. Click the **T-Online Account Setup** button. This will open the **Account Setup** window for your T-Online account as shown in *Figure 5.13, “Account Setup”*.



Please enter here your personal account data. Without this data, access to T-Online is unfortunately not posssible.

T-Online account

Adapter identifier:

Associated T-Online number:

Concurrent user number/suffix:

Personal password: password

Figure 5.13. Account Setup

4. Enter your **Adapter identifier**, **Associated T-Online number**, **Concurrent user number/suffix**, and **Personal password.**. Click **OK** when finished to close the **Account Setup** window.

5. On the **xDSL Configuration** window, click **OK**. Be sure to select **File** > **Save** from the **Network Administration Tool** to save the changes.

When the device is added, it is not activated immediately, as seen by its **Inactive** status. To activate the device, select it from the device list, and click the **Activate** button. If the system is configured to activate the device when the computer starts (the default), this step does not have to be performed again.

5.6. Establishing a Token Ring Connection

A *token ring network* is a network in which all the computers are connected in a circular pattern. A *token*, or a special network packet, travels around the token ring and allows computers to send information to each other.

Tip

For more information on using token rings under Linux, refer to the *Linux Token Ring Project* website available at *http://www.linuxtr.net/*.

To add a token ring connection, follow these steps:

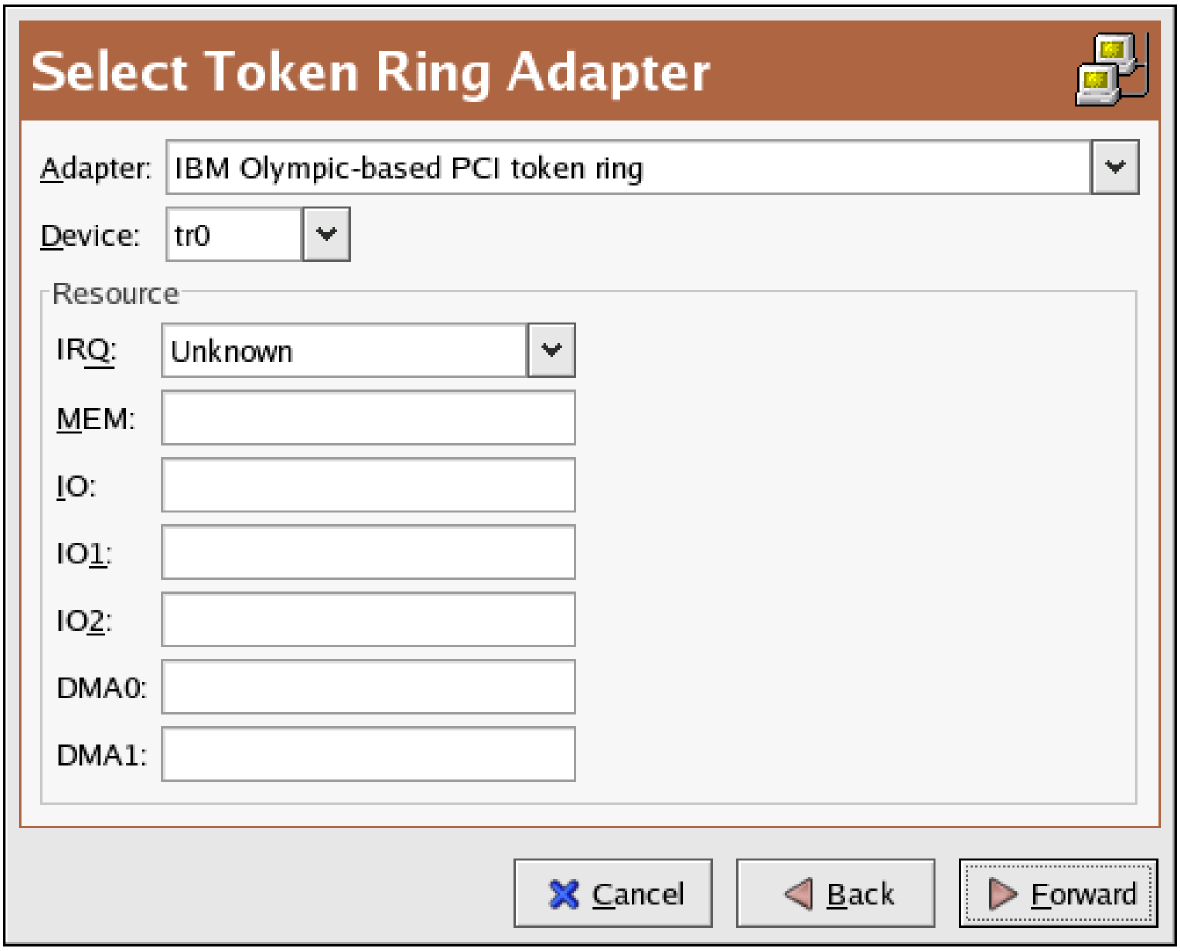
1. Click the **Devices** tab.

2. Click the **New** button on the toolbar.

3. Select **Token Ring connection** from the **Device Type** list and click **Forward**.

4. If you have already added the token ring card to the hardware list, select it from the **Tokenring card** list. Otherwise, select **Other Tokenring Card** to add the hardware device.

5. If you selected **Other Tokenring Card**, the **Select Token Ring Adapter** window as shown in *Figure 5.14, “Token Ring Settings”* appears. Select the manufacturer and model of the adapter. Select the device name. If this is the system's first token ring card, select **tr0**; if this is the second token ring card, select **tr1** (and so on). The **Network Administration Tool** also allows the user to configure the resources for the adapter. Click **Forward** to continue.



Select Token Ring Adapter

Adapter: IBM Olympic-based PCI token ring

Device: tr0

Resource

IRQ: Unknown

MEM:

IO:

IO1:

IO2:

DMA0:

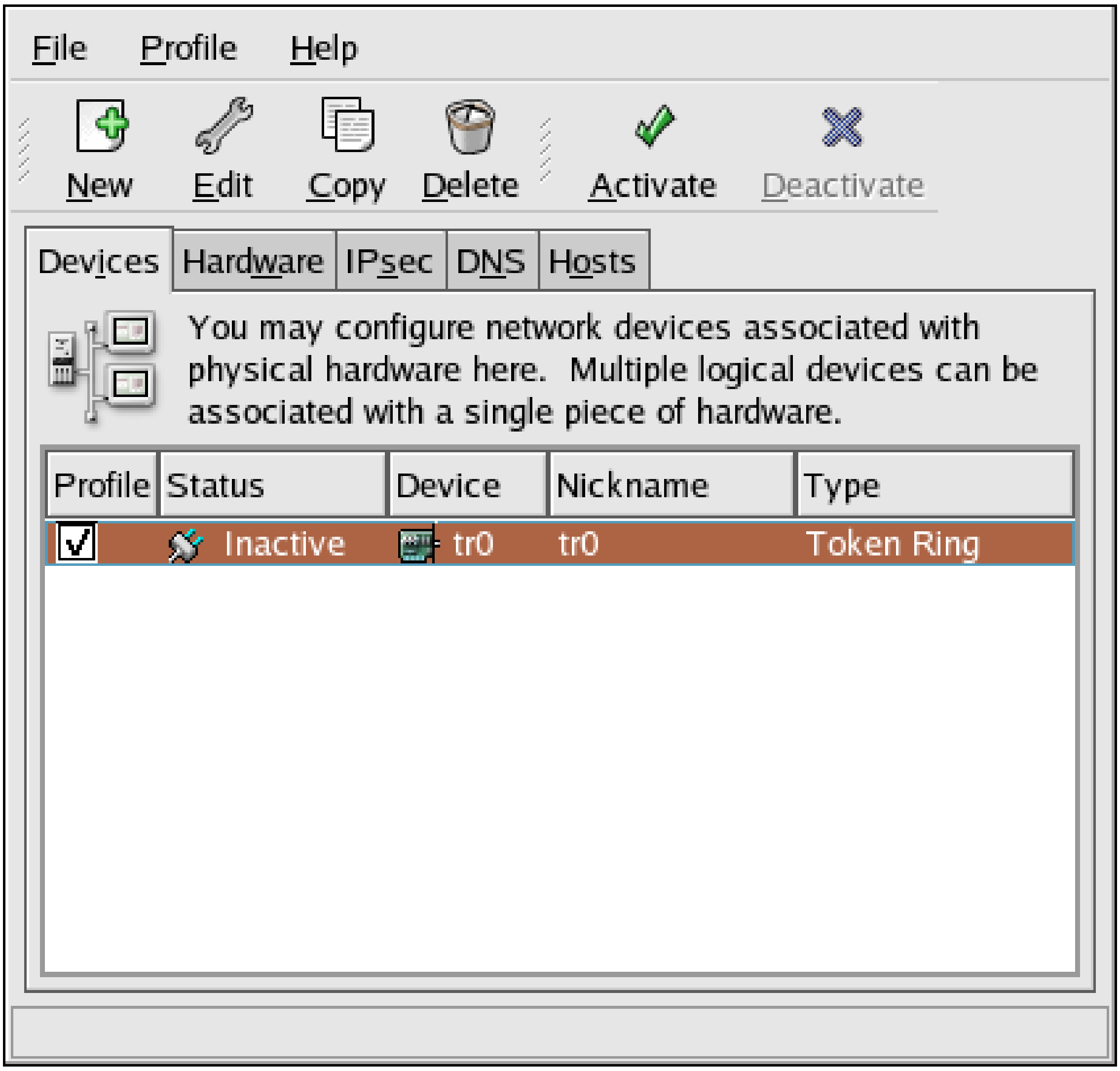
DMA1:

Figure 5.14. Token Ring Settings

6. On the **Configure Network Settings** page, choose between DHCP and static IP address. You may specify a hostname for the device. If the device receives a dynamic IP address each time the network is started, do not specify a hostname. Click **Forward** to continue.

7. Click **Apply** on the **Create Tokenring Device** page.

After configuring the token ring device, it appears in the device list as shown in *Figure 5.15, “Token Ring Device”*.



You may configure network devices associated with physical hardware here. Multiple logical devices can be associated with a single piece of hardware.

Profile	Status	Device	Nickname	Type
☑	Inactive	tr0	tr0	Token Ring

Figure 5.15. Token Ring Device

Be sure to select **File** > **Save** to save the changes.

After adding the device, you can edit its configuration by selecting the device from the device list and clicking **Edit**. For example, you can configure whether the device is started at boot time.

When the device is added, it is not activated immediately, as seen by its **Inactive** status. To activate the device, select it from the device list, and click the **Activate** button. If the system is configured to activate the device when the computer starts (the default), this step does not have to be performed again.

5.7. Establishing a Wireless Connection

Wireless Ethernet devices are becoming increasingly popular. The configuration is similar to the Ethernet configuration except that it allows you to configure settings such as the SSID and key for the wireless device.

To add a wireless Ethernet connection, follow these steps:

1. Click the **Devices** tab.

2. Click the **New** button on the toolbar.

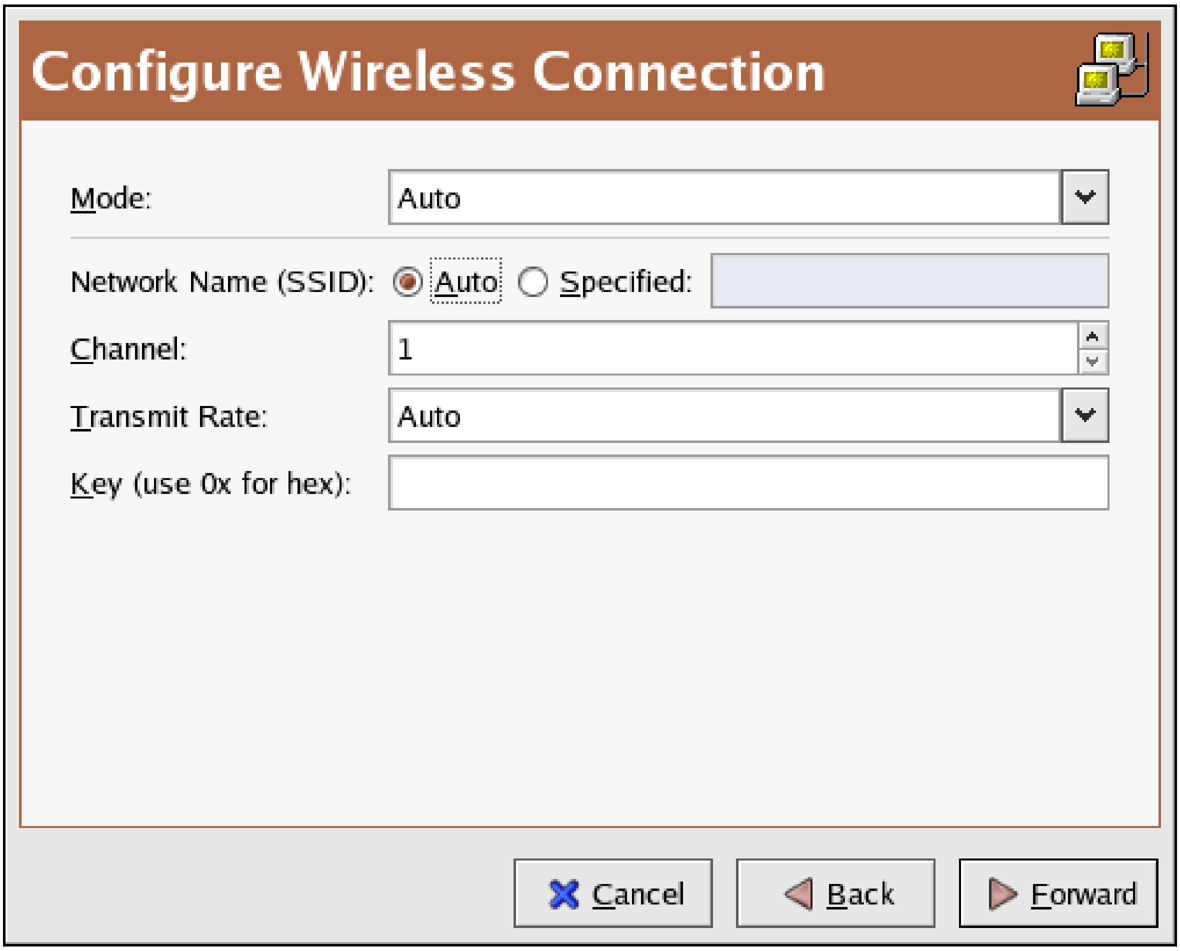
3. Select **Wireless connection** from the **Device Type** list and click **Forward**.

4. If you have already added the wireless network interface card to the hardware list, select it from the **Wireless card** list. Otherwise, select **Other Wireless Card** to add the hardware device.

> **Note**
>
> The installation program usually detects supported wireless Ethernet devices and prompts you to configure them. If you configured them during the installation, they are displayed in the hardware list on the **Hardware** tab.

5. If you selected **Other Wireless Card**, the **Select Ethernet Adapter** window appears. Select the manufacturer and model of the Ethernet card and the device. If this is the first Ethernet card for the system, select **eth0**; if this is the second Ethernet card for the system, select **eth1** (and so on). The **Network Administration Tool** also allows the user to configure the resources for the wireless network interface card. Click **Forward** to continue.

6. On the **Configure Wireless Connection** page as shown in *Figure 5.16, "Wireless Settings"*, configure the settings for the wireless device.



Configure Wireless Connection

Mode: Auto

Network Name (SSID): ● Auto ○ Specified:

Channel: 1

Transmit Rate: Auto

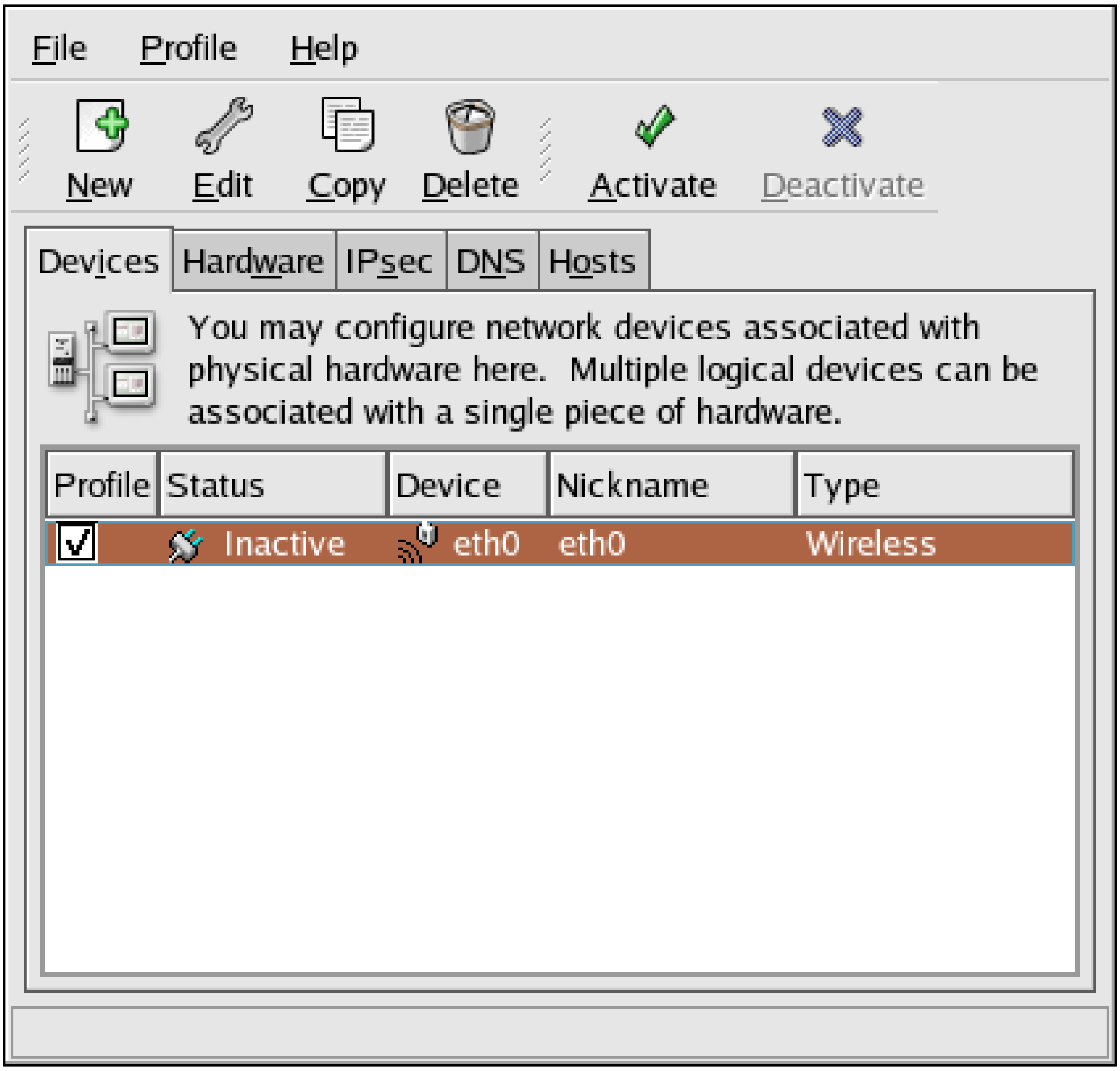
Key (use 0x for hex):

Figure 5.16. Wireless Settings

7. On the **Configure Network Settings** page, choose between DHCP and static IP address. You may specify a hostname for the device. If the device receives a dynamic IP address each time the network is started, do not specify a hostname. Click **Forward** to continue.

8. Click **Apply** on the **Create Wireless Device** page.

After configuring the wireless device, it appears in the device list as shown in *Figure 5.17, “Wireless Device”*.



You may configure network devices associated with physical hardware here. Multiple logical devices can be associated with a single piece of hardware.

Profile	Status	Device	Nickname	Type
☑	Inactive	eth0	eth0	Wireless

Figure 5.17. Wireless Device

Be sure to select **File** > **Save** to save the changes.

After adding the wireless device, you can edit its configuration by selecting the device from the device list and clicking **Edit**. For example, you can configure the device to activate at boot time.

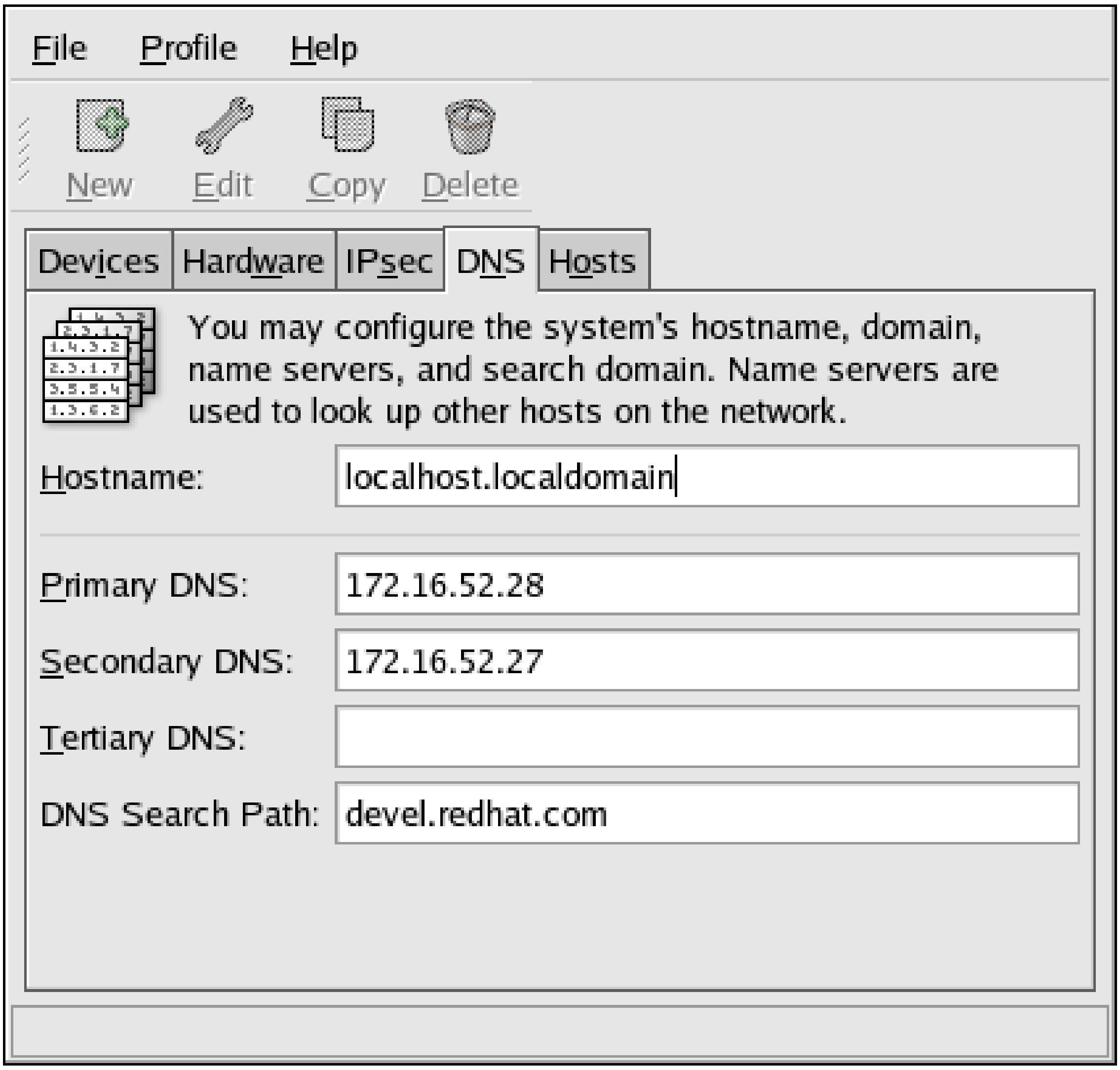
When the device is added, it is not activated immediately, as seen by its **Inactive** status. To activate the device, select it from the device list, and click the **Activate** button. If the system is configured to activate the device when the computer starts (the default), this step does not have to be performed again.

5.8. Managing DNS Settings

The **DNS** tab allows you to configure the system's hostname, domain, name servers, and search domain. Name servers are used to look up other hosts on the network.

If the DNS server names are retrieved from DHCP or PPPoE (or retrieved from the ISP), do not add primary, secondary, or tertiary DNS servers.

If the hostname is retrieved dynamically from DHCP or PPPoE (or retrieved from the ISP), do not change it.



You may configure the system's hostname, domain, name servers, and search domain. Name servers are used to look up other hosts on the network.

Hostname: localhost.localdomain

Primary DNS: 172.16.52.28

Secondary DNS: 172.16.52.27

Tertiary DNS:

DNS Search Path: devel.redhat.com

Figure 5.18. DNS Configuration

Note

The name servers section does not configure the system to be a name server. Instead, it configures which name servers to use when resolving IP addresses to hostnames and vice-versa.

Warning

If the hostname is changed and **`system-config-network`** is started on the local host, you may not be able to start another **X11** application. As such, you may have to re-login to a new desktop session.

5.9. Managing Hosts

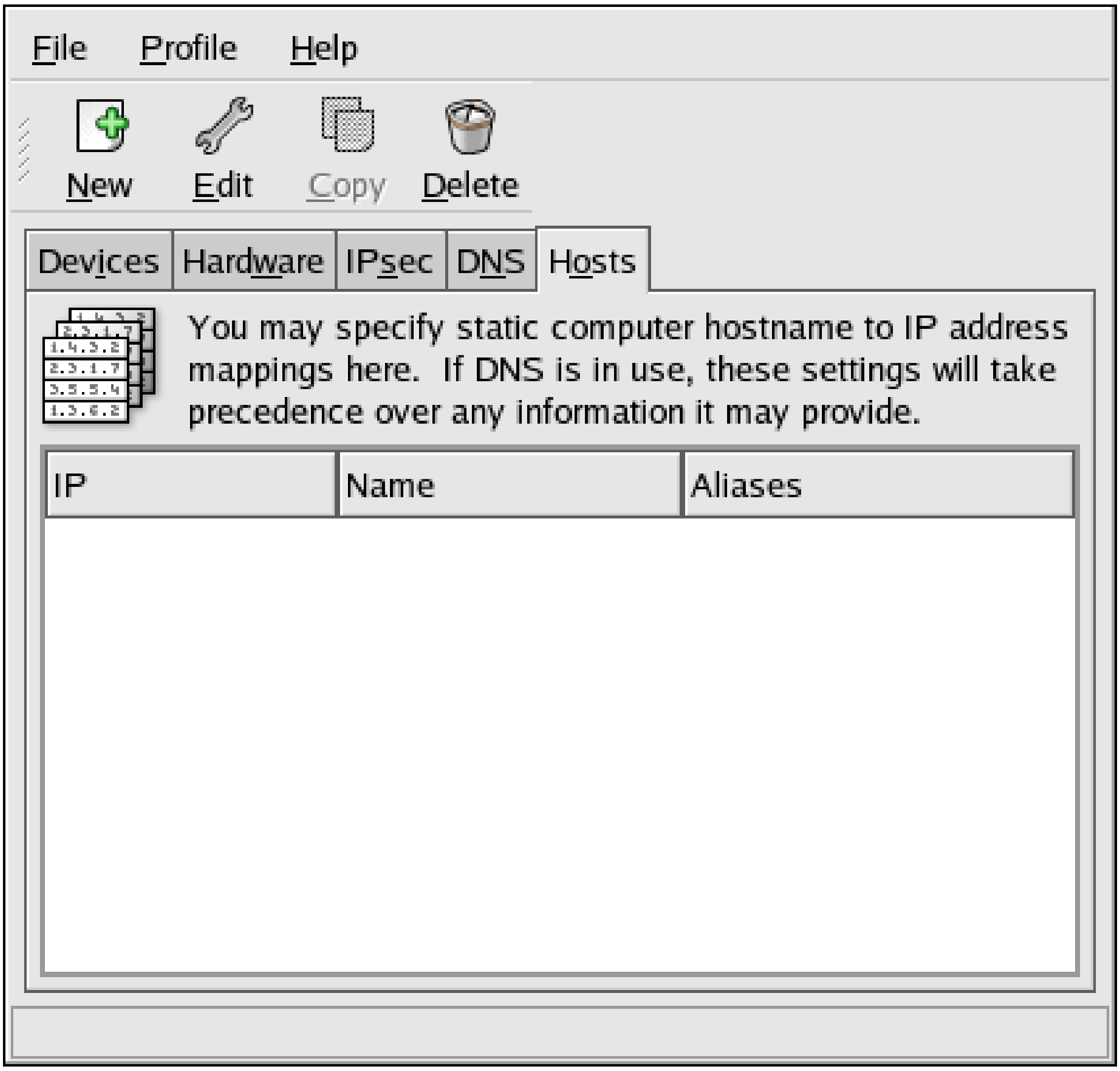
The **Hosts** tab allows you to add, edit, or remove hosts from the **`/etc/hosts`** file. This file contains IP addresses and their corresponding hostnames.

When your system tries to resolve a hostname to an IP address or tries to determine the hostname for an IP address, it refers to the **`/etc/hosts`** file before using the name servers (if you are using the default Fedora configuration). If the IP address is listed in the **`/etc/hosts`** file, the name servers are not used. If your network contains computers whose IP addresses are not listed in DNS, it is recommended that you add them to the **`/etc/hosts`** file.

To add an entry to the **`/etc/hosts`** file, go to the **Hosts** tab, click the **New** button on the toolbar, provide the requested information, and click **OK**. Select **File** > **Save** or press **`Ctrl`**+**S** to save the changes to the **`/etc/hosts`** file. The network or network services do not need to be restarted since the current version of the file is referred to each time an address is resolved.

Warning

Do not remove the `localhost` entry. Even if the system does not have a network connection or have a network connection running constantly, some programs need to connect to the system via the localhost loopback interface.



You may specify static computer hostname to IP address mappings here. If DNS is in use, these settings will take precedence over any information it may provide.

Figure 5.19. Hosts Configuration

Tip

To change lookup order, edit the **`/etc/host.conf`** file. The line `order hosts, bind` specifies that **`/etc/hosts`** takes precedence over the name servers. Changing the line to **`order bind, hosts`** configures the system to resolve hostnames and IP addresses using the name servers first. If the IP address cannot be resolved through the name servers, the system then looks for the IP address in the **`/etc/hosts`** file.

5.10. Working with Profiles

Multiple logical network devices can be created for each physical hardware device. For example, if you have one Ethernet card in your system (eth0), you can create logical network devices with different nicknames and different configuration options, all to be specifically associated with eth0.

Logical network devices are different from device aliases. Logical network devices associated with the same physical device must exist in different profiles and cannot be activated simultaneously. Device aliases are also associated with the same physical hardware device, but device aliases associated with the same physical hardware can be activated at the same time. Refer to *Section 5.11, “Device Aliases”* for details about creating device aliases.

Profiles can be used to create multiple configuration sets for different networks. A configuration set can include logical devices as well as hosts and DNS settings. After configuring the profiles, you can use the **Network Administration Tool** to switch back and forth between them.

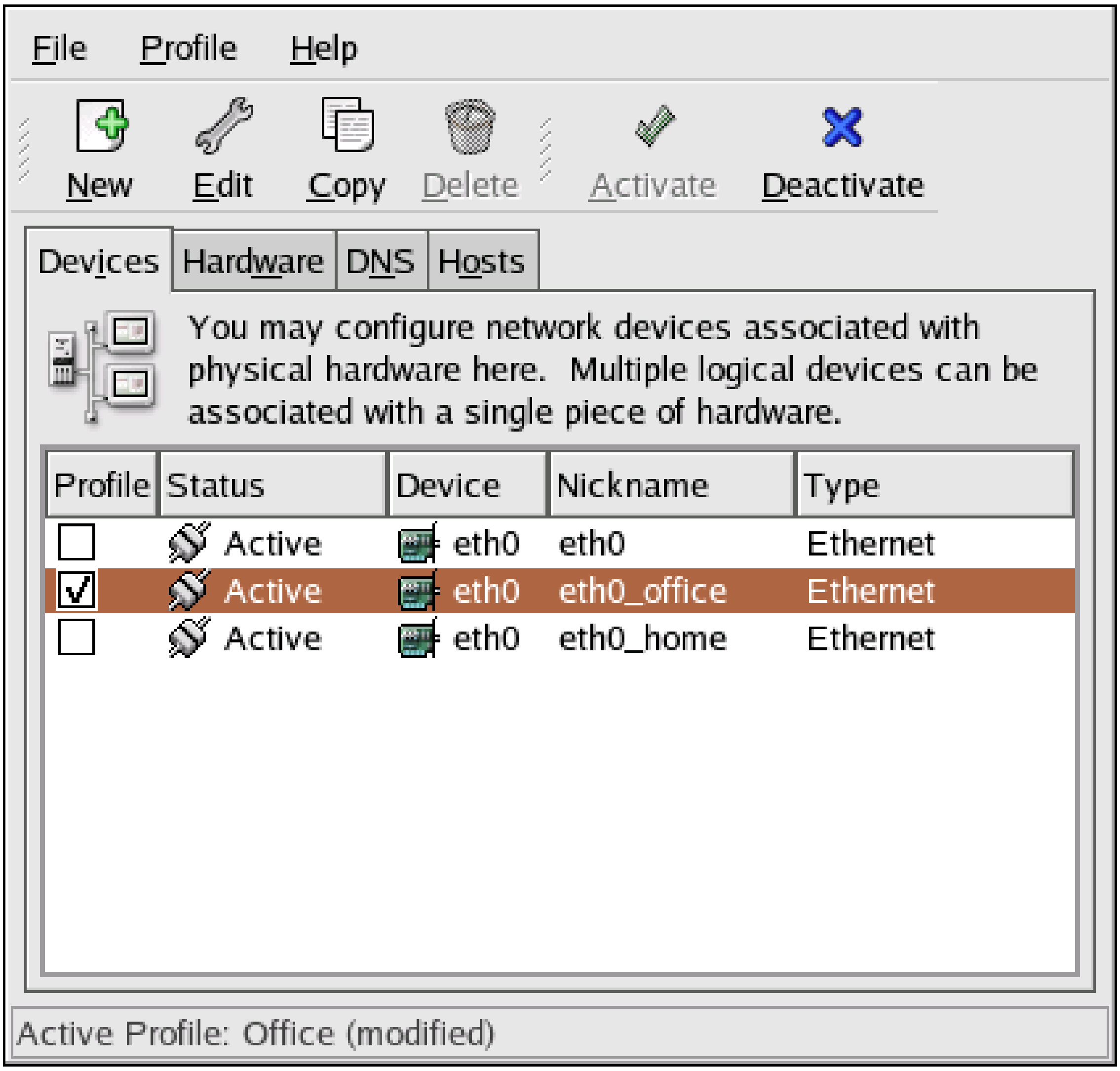
By default, there is one profile called **Common**. To create a new profile, select **Profile** > **New** from the pull-down menu, and enter a unique name for the profile.

You are now modifying the new profile as indicated by the status bar at the bottom of the main window.

Click on an existing device already in the list and click the **Copy** button to copy the existing device to a logical network device. If you use the **New** button, a network alias is created, which is incorrect. To change the properties of the logical device, select it from the list and click **Edit**. For example, the nickname can be changed to a more descriptive name, such as **`eth0_office`**, so that it can be recognized more easily.

In the list of devices, there is a column of checkboxes labeled **Profile**. For each profile, you can check or uncheck devices. Only the checked devices are included for the currently selected profile. For example, if you create a logical device named **`eth0_office`** in a profile called **`Office`** and want to activate the logical device if the profile is selected, uncheck the `eth0` device and check the `eth0_office` device.

For example, *Figure 5.20, “Office Profile”* shows a profile called **Office** with the logical device **eth0_office**. It is configured to activate the first Ethernet card using DHCP.



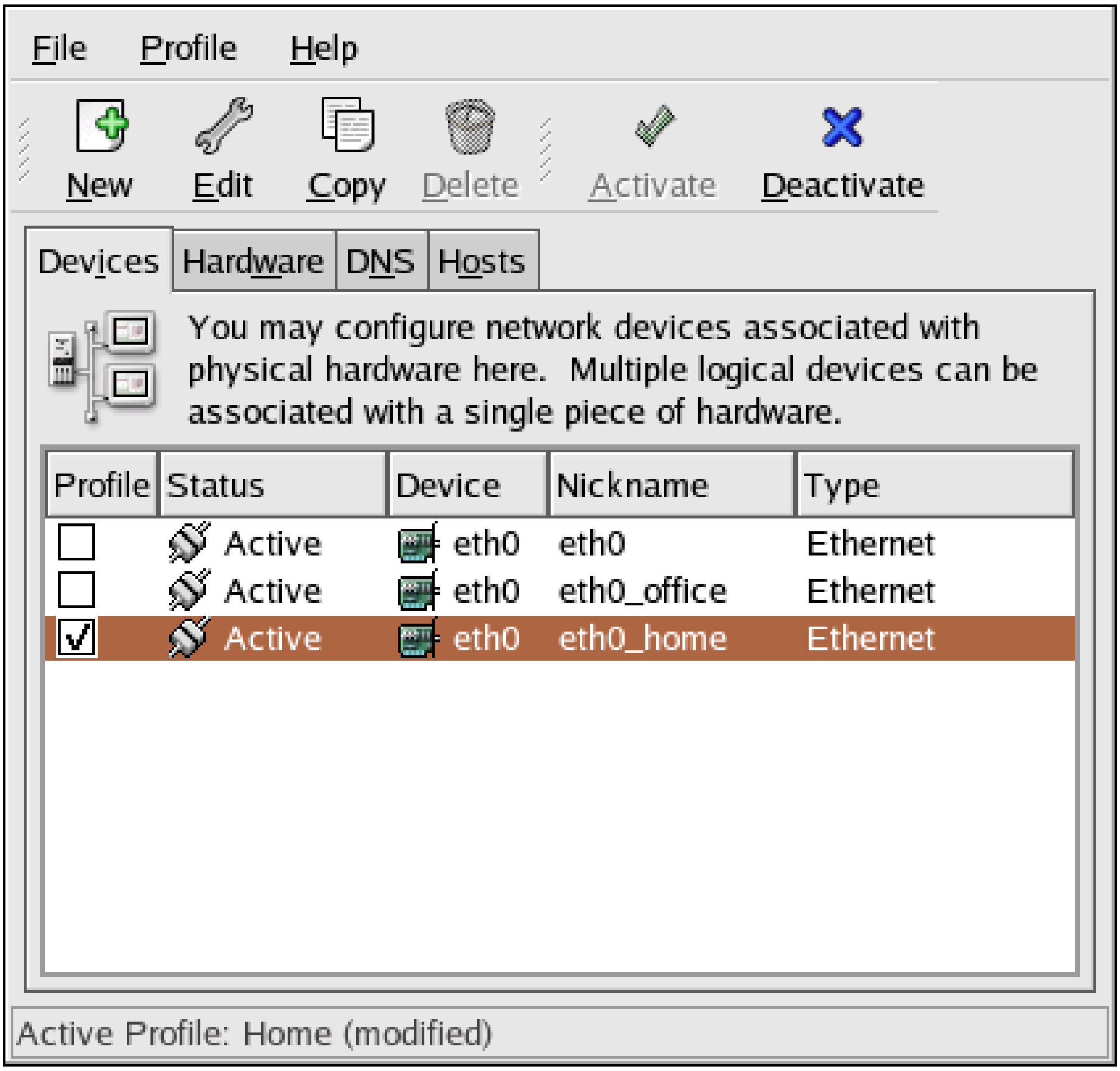
You may configure network devices associated with physical hardware here. Multiple logical devices can be associated with a single piece of hardware.

Profile	Status	Device	Nickname	Type
□	Active	eth0	eth0	Ethernet
☑	Active	eth0	eth0_office	Ethernet
□	Active	eth0	eth0_home	Ethernet

Active Profile: Office (modified)

Figure 5.20. Office Profile

Notice that the **Home** profile as shown in *Figure 5.21, “Home Profile”* activates the **eth0_home** logical device, which is associated with `eth0`.



You may configure network devices associated with physical hardware here. Multiple logical devices can be associated with a single piece of hardware.

Profile	Status	Device	Nickname	Type
□	Active	eth0	eth0	Ethernet
□	Active	eth0	eth0_office	Ethernet
☑	Active	eth0	eth0_home	Ethernet

Active Profile: Home (modified)

Figure 5.21. Home Profile

You can also configure `eth0` to activate in the **Office** profile only and to activate a PPP (modem) device in the **Home** profile only. Another example is to have the **Common** profile activate `eth0` and an **Away** profile activate a PPP device for use while traveling.

To activate a profile at boot time, modify the boot loader configuration file to include the `netprofile=<`*`profilename`*`>` option. For example, if the system uses GRUB as the boot loader and **`/boot/grub/grub.conf`** contains:

```
title Red Hat Enterprise Linux (2.6.9-5.EL)
        root (hd0,0)
  kernel /vmlinuz-2.6.9-5.EL ro root=/dev/VolGroup00/LogVol00 rhgb quiet
  initrd /initrd-2.6.9-5.EL.img
```

Modify it to the following (where <*`profilename`*> is the name of the profile to be activated at boot time):

```
title Red Hat Enterprise Linux (2.6.9-5.EL)
        root (hd0,0)
```

```
kernel /vmlinuz-2.6.9-5.EL ro root=/dev/VolGroup00/LogVol00 \
 netprofile=<profilename> \    rhgb quiet
initrd /initrd-2.6.9-5.EL.img
```

To switch profiles after the system has booted, go to Applications (the main menu on the panel) > **System Tools** > **Network Device Control** (or type the command **`system-control-network`**) to select a profile and activate it. The activate profile section only appears in the **Network Device Control** interface if more than the default **Common** interface exists.

Alternatively, execute the following command to enable a profile (replace <*`profilename`*> with the name of the profile):

```
system-config-network-cmd --profile <profilename> --activate
```

5.11. Device Aliases

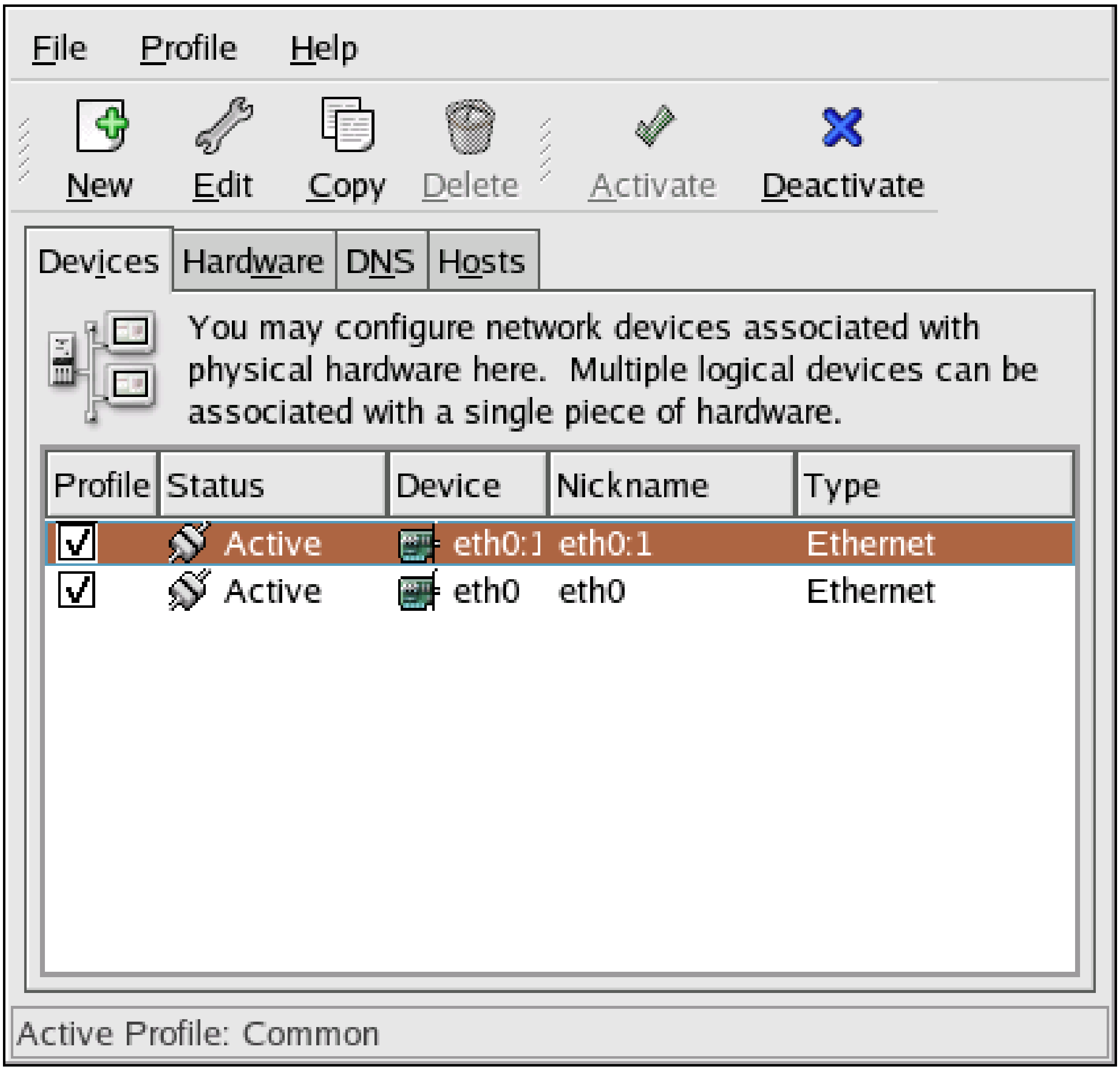
Device aliases are virtual devices associated with the same physical hardware, but they can be activated at the same time to have different IP addresses. They are commonly represented as the device name followed by a colon and a number (for example, eth0:1). They are useful if you want to have multiple IP addresses for a system that only has one network card.

After configuring the Ethernet device —such as `eth0` —to use a static IP address (DHCP does not work with aliases), go to the **Devices** tab and click **New**. Select the Ethernet card to configure with an alias, set the static IP address for the alias, and click **Apply** to create it. Since a device already exists for the Ethernet card, the one just created is the alias, such as `eth0:1`.

Warning

If you are configuring an Ethernet device to have an alias, neither the device nor the alias can be configured to use DHCP. You must configure the IP addresses manually.

Figure 5.22, “Network Device Alias Example” shows an example of one alias for the `eth0` device. Notice the `eth0:1` device — the first alias for `eth0`. The second alias for `eth0` would have the device name `eth0:2`, and so on. To modify the settings for the device alias, such as whether to activate it at boot time and the alias number, select it from the list and click the **Edit** button.



You may configure network devices associated with physical hardware here. Multiple logical devices can be associated with a single piece of hardware.

Profile	Status	Device	Nickname	Type
☑	Active	eth0:1	eth0:1	Ethernet
☑	Active	eth0	eth0	Ethernet

Active Profile: Common

Figure 5.22. Network Device Alias Example

Select the alias and click the **Activate** button to activate the alias. If you have configured multiple profiles, select which profiles in which to include it.

To verify that the alias has been activated, use the command **`/sbin/ifconfig`**. The output should show the device and the device alias with different IP addresses:

```
eth0      Link encap:Ethernet
 HWaddr 00:A0:CC:60:B7:G4
 inet addr:192.168.100.5  Bcast:192.168.100.255  Mask:255.255.255.0
 UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
 RX packets:161930 errors:1 dropped:0 overruns:0 frame:0
 TX packets:244570 errors:0 dropped:0 overruns:0 carrier:0
 collisions:475 txqueuelen:100
 RX bytes:55075551 (52.5 Mb)  TX bytes:178108895 (169.8 Mb)
 Interrupt:10 Base address:0x9000  eth0:1    Link encap:Ethernet  HWaddr
 00:A0:CC:60:B7:G4
 inet addr:192.168.100.42  Bcast:192.168.100.255  Mask:255.255.255.0
 UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

```
Interrupt:10 Base address:0x9000  lo
Link encap:Local Loopback
inet addr:127.0.0.1  Mask:255.0.0.0
UP LOOPBACK RUNNING  MTU:16436  Metric:1
RX packets:5998 errors:0 dropped:0 overruns:0 frame:0
TX packets:5998 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:1627579 (1.5 Mb)  TX bytes:1627579 (1.5 Mb)
```

5.12. Saving and Restoring the Network Configuration

The command line version of **Network Administration Tool** can be used to save the system's network configuration to a file. This file can then be used to restore the network settings to a Fedora system.

This feature can be used as part of an automated backup script, to save the configuration before upgrading or reinstalling, or to copy the configuration to a different Fedora system.

To save, or *export*, the network configuration of a system to the file **`/tmp/network-config`**, execute the following command as root:

```
system-config-network-cmd -e > /tmp/network-config
```

To restore, or *import*, the network configuration from the file created from the previous command, execute the following command as root:

```
system-config-network-cmd -i -c -f /tmp/network-config
```

The `-i` option means to import the data, the `-c` option means to clear the existing configuration prior to importing, and the `-f` option specifies that the file to import is as follows.

Controlling Access to Services

Maintaining security on your system is extremely important, and one approach for this task is to manage access to system services carefully. Your system may need to provide open access to particular services (for example, **`httpd`** if you are running a Web server). However, if you do not need to provide a service, you should turn it off to minimize your exposure to possible bug exploits.

There are several different methods for managing access to system services. Choose which method of management to use based on the service, your system's configuration, and your level of Linux expertise.

The easiest way to deny access to a service is to turn it off. Both the services managed by **`xinetd`** and the services in the **`/etc/rc.d/init.d`** hierarchy (also known as SysV services) can be configured to start or stop using three different applications:

Services Configuration Tool

This is a graphical application that displays a description of each service, displays whether each service is started at boot time (for runlevels 3, 4, and 5), and allows services to be started, stopped, and restarted.

ntsysv

This is a text-based application that allows you to configure which services are started at boot time for each runlevel. Non-**`xinetd`** services can not be started, stopped, or restarted using this program.

`chkconfig`

This is a command line utility that allows you to turn services on and off for the different runlevels. Non-**`xinetd`** services can not be started, stopped, or restarted using this utility.

You may find that these tools are easier to use than the alternatives — editing the numerous symbolic links located in the directories below **`/etc/rc.d`** by hand or editing the **`xinetd`** configuration files in **`/etc/xinetd.d`**.

Another way to manage access to system services is by using **`iptables`** to configure an IP firewall. If you are a new Linux user, note that **`iptables`** may not be the best solution for you. Setting up **`iptables`** can be complicated, and is best tackled by experienced Linux system administrators.

On the other hand, the benefit of using **`iptables`** is flexibility. For example, if you need a customized solution which provides certain hosts access to certain services, **`iptables`** can provide it for you. Refer to and for more information about **`iptables`**.

Alternatively, if you are looking for a utility to set general access rules for your home machine, and/or if you are new to Linux, try the **Security Level Configuration Tool** (**`system-config-securitylevel`**), which allows you to select the security level for your system, similar to the **Firewall Configuration** screen in the installation program.

Refer to for more information.

> **Important**
>
> When you allow access for new services, always remember that both the firewall and SELinux need to be configured as well. One of the most common mistakes committed

when configuring a new service is neglecting to implement the necessary firewall configuration and SELinux policies to allow access for it. Refer to for more information.

6.1. Runlevels

Before you can configure access to services, you must understand Linux runlevels. A runlevel is a state, or *mode*, that is defined by the services listed in the directory **`/etc/rc.d/rc<x>.d`**, where *<x>* is the number of the runlevel.

The following runlevels exist:

- 0 — Halt
- 1 — Single-user mode
- 2 — Not used (user-definable)
- 3 — Full multi-user mode
- 4 — Not used (user-definable)
- 5 — Full multi-user mode (with an X-based login screen)
- 6 — Reboot

If you use a text login screen, you are operating in runlevel 3. If you use a graphical login screen, you are operating in runlevel 5.

The default runlevel can be changed by modifying the **`/etc/inittab`** file, which contains a line near the top of the file similar to the following:

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

Change the number in this line to the desired runlevel. The change does not take effect until you reboot the system.

6.2. TCP Wrappers

Many UNIX system administrators are accustomed to using TCP wrappers to manage access to certain network services. Any network services managed by **`xinetd`** (as well as any program with built-in support for **`libwrap`**) can use TCP wrappers to manage access. **`xinetd`** can use the **`/etc/hosts.allow`** and **`/etc/hosts.deny`** files to configure access to system services. As the names imply, **`hosts.allow`** contains a list of rules that allow clients to access the network services controlled by **`xinetd`**, and **`hosts.deny`** contains rules to deny access. The **`hosts.allow`** file takes precedence over the **`hosts.deny`** file. Permissions to grant or deny access can be based on individual IP address (or hostnames) or on a pattern of clients. Refer to **`hosts_access`** in section 5 of the man pages (**`man 5 hosts_access`**) for details.

6.2.1. `xinetd`

To control access to Internet services, use **`xinetd`**, which is a secure replacement for **`inetd`**. The **`xinetd`** daemon conserves system resources, provides access control and logging, and can be used to start special-purpose servers. **`xinetd`** can also be used to grant or deny access to particular hosts, provide service access at specific times, limit the rate of incoming connections, limit the load created by connections, and more.

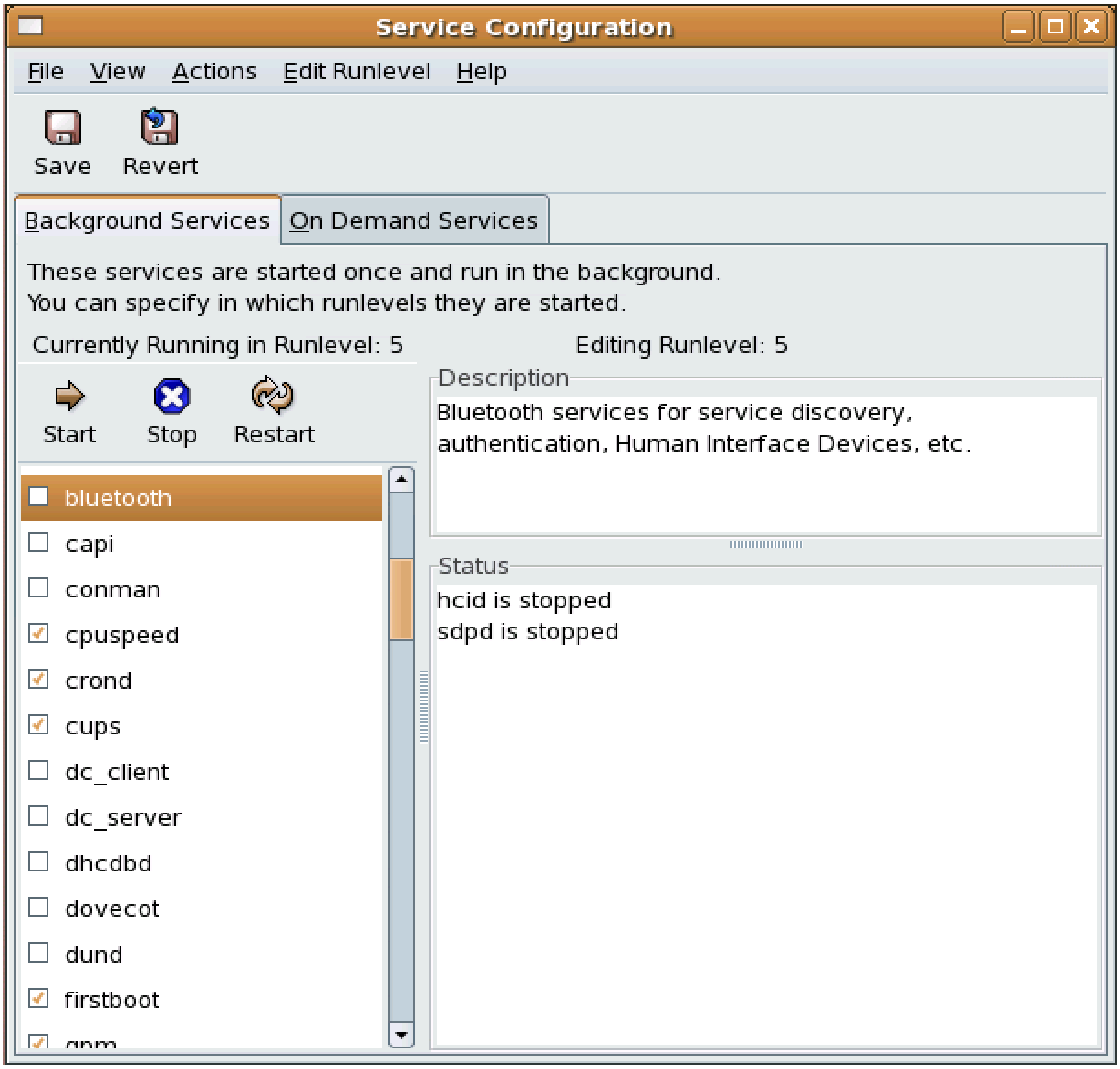
`xinetd` runs constantly and listens on all ports for the services it manages. When a connection request arrives for one of its managed services, **`xinetd`** starts up the appropriate server for that service.

The configuration file for **`xinetd`** is **`/etc/xinetd.conf`**, but the file only contains a few defaults and an instruction to include the **`/etc/xinetd.d`** directory. To enable or disable an **`xinetd`** service, edit its configuration file in the **`/etc/xinetd.d`** directory. If the `disable` attribute is set to **`yes`**, the service is disabled. If the `disable` attribute is set to **`no`**, the service is enabled. You can edit any of the **`xinetd`** configuration files or change its enabled status using the **Services Configuration Tool**, **ntsysv**, or **`chkconfig`**. For a list of network services controlled by **`xinetd`**, review the contents of the **`/etc/xinetd.d`** directory with the command **`ls /etc/xinetd.d`**.

6.3. Services Configuration Tool

The **Services Configuration Tool** is a graphical application developed by Red Hat to configure which SysV services in the **`/etc/rc.d/init.d`** directory are started at boot time (for runlevels 3, 4, and 5) and which **`xinetd`** services are enabled. It also allows you to start, stop, and restart SysV services as well as reload **`xinetd`**.

To start the **Services Configuration Tool** from the desktop, go to the Applications (the main menu on the panel) > **System Settings** > **Server Settings** > **Services** or type the command **`system-config-services`** at a shell prompt (for example, in an **XTerm** or a **GNOME terminal**).



These services are started once and run in the background.
You can specify in which runlevels they are started.

Currently Running in Runlevel: 5

Editing Runlevel: 5

- □ bluetooth
- □ capi
- □ conman
- ☑ cpuspeed
- ☑ crond
- ☑ cups
- □ dc_client
- □ dc_server
- □ dhcdbd
- □ dovecot
- □ dund
- ☑ firstboot
- ☑ gpm

Description

Bluetooth services for service discovery, authentication, Human Interface Devices, etc.

Status

hcid is stopped
sdpd is stopped

Figure 6.1. **Services Configuration Tool**

The **Services Configuration Tool** displays the current runlevel as well as the runlevel you are currently editing. To edit a different runlevel, select **Edit Runlevel** from the pulldown menu and select runlevel 3, 4, or 5. Refer to *Section 6.1, “Runlevels”* for a description of runlevels.

The **Services Configuration Tool** lists the services from the **`/etc/rc.d/init.d`** directory as well as the services controlled by **`xinetd`**. Click on the name of the service from the list on the left-hand side of the application to display a brief description of that service as well as the status of the service. If the service is not an **`xinetd`** service, the status window shows whether the service is currently running. If the service is controlled by **`xinetd`**, the status window displays the phrase **xinetd service**.

To start, stop, or restart a service immediately, select the service from the list and click the appropriate button on the toolbar (or choose the action from the **Actions** pulldown menu). If the service is an **`xinetd`** service, the action buttons are disabled because they cannot be started or stopped individually.

If you enable/disable an **`xinetd`** service by checking or unchecking the checkbox next to the service name, you must select **File** > **Save Changes** from the pulldown menu (or the **Save** button above the tabs) to reload **`xinetd`** and immediately enable/disable the **`xinetd`** service that you changed.

xinetd is also configured to remember the setting. You can enable/disable multiple **xinetd** services at a time and save the changes when you are finished.

For example, assume you check **rsync** to enable it in runlevel 3 and then save the changes. The **rsync** service is immediately enabled. The next time **xinetd** is started, **rsync** is still enabled.

> **Note**
>
> When you save changes to **xinetd** services, **xinetd** is reloaded, and the changes take place immediately. When you save changes to other services, the runlevel is reconfigured, but the changes do not take effect immediately.

To enable a non-**xinetd** service to start at boot time for the currently selected runlevel, check the box beside the name of the service in the list. After configuring the runlevel, apply the changes by selecting **File** > **Save Changes** from the pulldown menu. The runlevel configuration is changed, but the runlevel is not restarted; thus, the changes do not take place immediately.

For example, assume you are configuring runlevel 3. If you change the value for the **httpd** service from checked to unchecked and then select **Save Changes**, the runlevel 3 configuration changes so that **httpd** is not started at boot time. However, runlevel 3 is not reinitialized, so **httpd** is still running. Select one of following options at this point:

1. Stop the **httpd** service — Stop the service by selecting it from the list and clicking the **Stop** button. A message appears stating that the service was stopped successfully.

2. Reinitialize the runlevel — Reinitialize the runlevel by going to a shell prompt and typing the command **telinit *x*** (where *x* is the runlevel number; in this example, 3.). This option is recommended if you change the **Start at Boot** value of multiple services and want to activate the changes immediately.

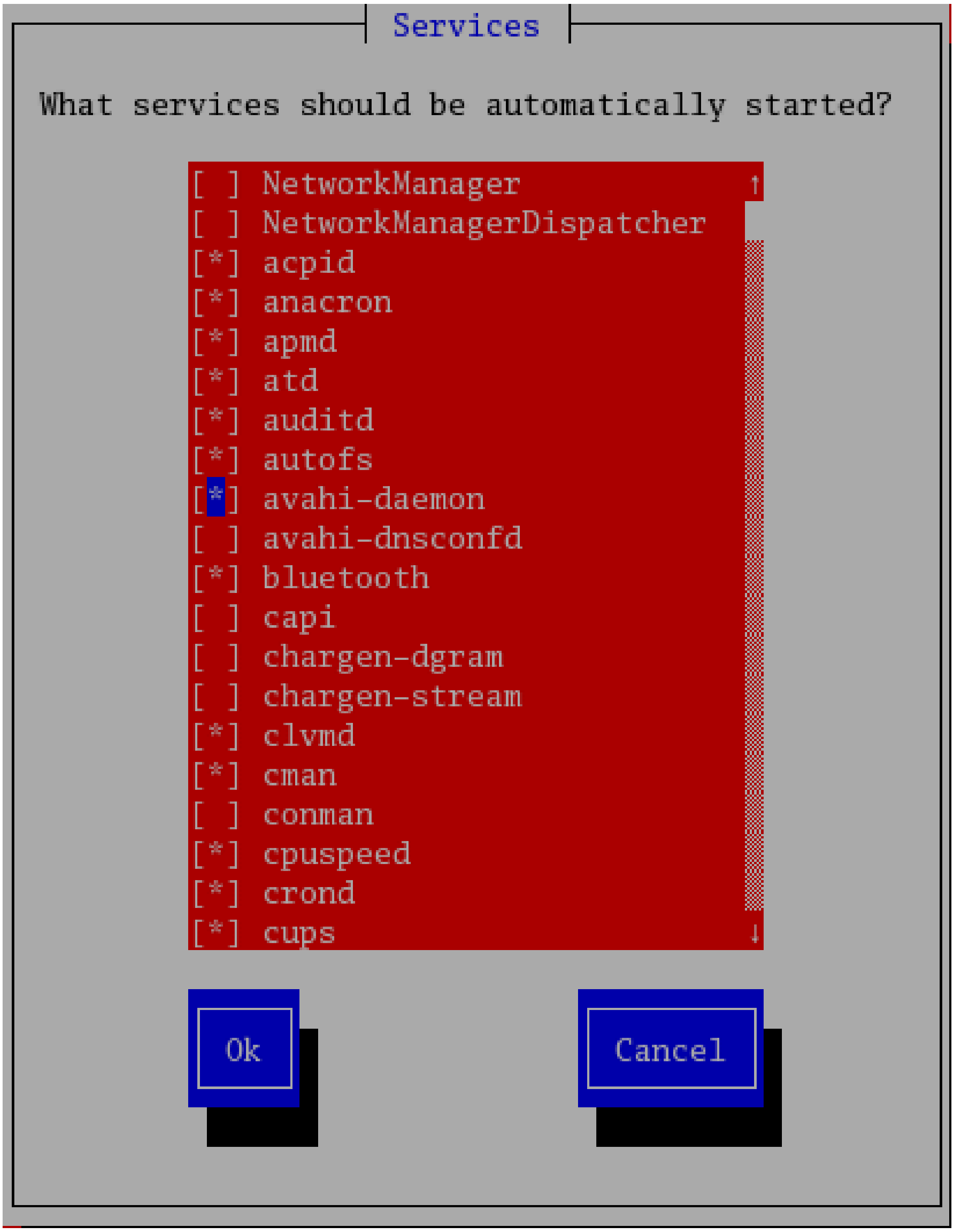
3. Do nothing else — You do not have to stop the **httpd** service. You can wait until the system is rebooted for the service to stop. The next time the system is booted, the runlevel is initialized without the **httpd** service running.

To add a service to a runlevel, select the runlevel from the **Edit Runlevel** pulldown menu, and then select **Actions** > **Add Service**. To delete a service from a runlevel, select the runlevel from the **Edit Runlevel** pulldown menu, select the service to be deleted from the list on the left, and select **Actions** > **Delete Service**.

6.4. ntsysv

The **ntsysv** utility provides a simple interface for activating or deactivating services. You can use **ntsysv** to turn an **xinetd**-managed service on or off. You can also use **ntsysv** to configure runlevels. By default, only the current runlevel is configured. To configure a different runlevel, specify one or more runlevels with the `--level` option. For example, the command **ntsysv --level 345** configures runlevels 3, 4, and 5.

The **ntsysv** interface works like the text mode installation program. Use the up and down arrows to navigate up and down the list. The space bar selects/unselects services and is also used to "press" the **Ok** and **Cancel** buttons. To move between the list of services and the **Ok** and **Cancel** buttons, use the **Tab** key. An asterisk (*****) signifies that a service is set to on. Pressing the **F1** key displays a short description of the selected service.



```
Services

What services should be automatically started?

[ ] NetworkManager
[ ] NetworkManagerDispatcher
[*] acpid
[*] anacron
[*] apmd
[*] atd
[*] auditd
[*] autofs
[*] avahi-daemon
[ ] avahi-dnsconfd
[*] bluetooth
[ ] capi
[ ] chargen-dgram
[ ] chargen-stream
[*] clvmd
[*] cman
[ ] conman
[*] cpuspeed
[*] crond
[*] cups

Ok        Cancel
```

Figure 6.2. The **ntsysv** utility

Warning

Services managed by **`xinetd`** are immediately affected by **ntsysv**. For all other services, changes do not take effect immediately. You must stop or start the individual service with the command **`service <daemon> stop`** (where *<daemon>* is the name of the service you want to stop; for example, **`httpd`**). Replace **`stop`** with **`start`** or **`restart`** to start or restart the service.

6.5. chkconfig

The **chkconfig** command can also be used to activate and deactivate services. The **chkconfig --list** command displays a list of system services and whether they are started (**on**) or stopped (**off**) in runlevels 0-6. At the end of the list is a section for the services managed by **xinetd**.

If the **chkconfig --list** command is used to query a service managed by **xinetd**, it displays whether the **xinetd** service is enabled (**on**) or disabled (**off**). For example, the command **chkconfig --list rsync** returns the following output:

```
rsync on
```

As shown, **rsync** is enabled as an **xinetd** service. If **xinetd** is running, **rsync** is enabled.

If you use **chkconfig --list** to query a service in **/etc/rc.d**, that service's settings for each runlevel are displayed. For example, the command **chkconfig --list httpd** returns the following output:

```
httpd 0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

chkconfig can also be used to configure a service to be started (or not) in a specific runlevel. For example, to turn **nscd** off in runlevels 3, 4, and 5, use the following command:

```

chkconfig --level 345 nscd off
```

Warning

Services managed by **xinetd** are immediately affected by **chkconfig**. For example, if **xinetd** is running while **rsync** is disabled, and the command **chkconfig rsync on** is executed, then **rsync** is immediately enabled without having to restart **xinetd** manually. Changes for other services do not take effect immediately after using **chkconfig**. You must stop or start the individual service with the command **service *<daemon>* stop** (where *<daemon>* is the name of the service you want to stop; for example, **httpd**). Replace **stop** with **start** or **restart** to start or restart the service.

6.6. Additional Resources

For more information, refer to the following resources.

6.6.1. Installed Documentation

- The man pages for **ntsysv**, **chkconfig**, **xinetd**, and **xinetd.conf**.
- **man 5 hosts_access** — The man page for the format of host access control files (in section 5 of the man pages).

6.6.2. Useful Websites

- *http://www.xinetd.org* — The **xinetd** webpage. It contains sample configuration files and a more detailed list of features.

Berkeley Internet Name Domain (BIND)

On most modern networks, including the Internet, users locate other computers by name. This frees users from the daunting task of remembering the numerical network address of network resources. The most effective way to configure a network to allow such name-based connections is to set up a *Domain Name Service* (*DNS*) or a *nameserver*, which resolves hostnames on the network to numerical addresses and vice versa.

This chapter reviews the nameserver included in Fedora and the *Berkeley Internet Name Domain* (*BIND*) DNS server, with an emphasis on the structure of its configuration files and how it may be administered both locally and remotely.

Note

BIND is also known as the service **`named`** in Fedora. You can manage it via the Services Configuration Tool (**`system-config-service`**).

7.1. Introduction to DNS

DNS associates hostnames with their respective IP addresses, so that when users want to connect to other machines on the network, they can refer to them by name, without having to remember IP addresses.

Use of DNS also has advantages for system administrators, allowing the flexibility to change the IP address for a host without affecting name-based queries to the machine. Conversely, administrators can shuffle which machines handle a name-based query.

DNS is normally implemented using centralized servers that are authoritative for some domains and refer to other DNS servers for other domains.

When a client host requests information from a nameserver, it usually connects to port 53. The nameserver then attempts to resolve the name requested. If the nameserver does not have an authoritative answer about the name the which host requested, or does not already have the answer cached from an earlier query, it queries other nameservers, called *root nameservers*, to determine which nameservers are authoritative for the name in question. Then, with that information, it queries the authoritative nameservers to get the requested name.

7.1.1. Nameserver Zones

In a DNS server such as BIND, all information is stored in basic data elements called *resource records*. A resource record is usually the *fully qualified domain name* (FQDN) of a host. Resource records are broken down into multiple sections. These sections are organized into a tree-like hierarchy consisting of a main trunk, primary branches, secondary branches, and so forth. Consider the following resource record:

```
bob.sales.example.com
```

When looking at how a resource record is resolved to find, for example, the IP address that relates to a particular system, read the name from right to left. Each level of the hierarchy is divided by a period (often called a "dot": "`.`"). In this example, therefore, `com` defines the *top-level domain* for this resource record. The name `example` is a sub-domain under `com`, while `sales` is a sub-domain under `example`. The name furthest to the left, `bob`, identifies a resource record which is part of the `sales.example.com` domain.

Except for the first (leftmost) part of the resource record (`bob`), each section is called a *zone*. Zone defines a specific *namespace*. A zone contains definitions of resource records, which usually contain host-to-IP address mappings and IP address-to-host mappings, which are called *reverse records*).

Zones are defined on authoritative nameservers through the use of *zone files*, which define the resource records in that zone. Zone files are stored on *primary nameservers* (also called *master nameservers*), where changes are made to the files, and *secondary nameservers* (also called *slave nameservers*), which receive zone definitions from the primary nameservers. Both primary and secondary nameservers are authoritative for the zone and look the same to clients. Any nameserver can be a primary or secondary nameserver for multiple zones at the same time. It all depends on how the nameserver is configured.

7.1.2. Nameserver Types

There are two nameserver configuration types:

authoritative
: This category includes both primary (master) and secondary (slave) servers. Those servers answer only for resource records which are part of their zones.

recursive
: Offers resolution services, but is not authoritative for any zones. Answers for all resolutions are cached in memory for a fixed period of time, which is specified by the retrieved RR.

A nameserver may be one or both of these types. For example, a nameserver can be a master for some zones, a slave for others, and offer recursive services for others. However the best practice is not to combine authoritative and recursive servers due their absolutely different requirements. Authoritative servers are available for all clients and they should be available all the time otherwise it is not possible to resolve particular subtree of the DNS database. Recursive lookups take far more time than authoritative responses thus recursive servers should be available for a restricted number of clients. Otherwise recursive server could be easy target for *distributed denial of service (DDoS)* attack.

7.1.3. BIND as a Nameserver

BIND is set of DNS related programs. It contains a monolithic nameserver called **`/usr/sbin/named`**, an administration utility called **`/usr/sbin/rndc`** and DNS debugging utility called **`/usr/bin/dig`**. More information about **`rndc`** can be found in *Section 7.4, “Using **`rndc`**”*.

BIND stores its configuration files in the following locations:

`/etc/named.conf`
: The configuration file for the **`named`** daemon

`/var/named/` directory
: The **`named`** working directory which stores zone and statistic files

Note

If you have installed the **bind-chroot** package, the BIND service will run in the **/var/named/chroot** environment. All configuration files will be moved there. As such, **named.conf** will be located in **/var/named/chroot/etc/named.conf**, and so on.

The next few sections review the BIND configuration in more detail.

7.2. /etc/named.conf

The **named.conf** file is a collection of statements using nested options surrounded by opening and closing ellipse characters, **{ }**. Administrators must be careful when editing **named.conf** to avoid syntax errors as many seemingly minor errors prevent the **named** service from starting.

A typical **named.conf** file is organized similar to the following example:

```
<statement-1> ["<statement-1-name>"] [<statement-1-class>]
 { <option-1>; <option-2>; <option-N>; }; <statement-2> ["<statement-2-
name>"] [<statement-2-class>] { <option-1>; <option-2>; <option-
N>; }; <statement-N> ["<statement-N-name>"] [<statement-N-class>]
 { <option-1>; <option-2>; <option-N>; };
```

7.2.1. Common Statement Types

The following types of statements are commonly used in **/etc/named.conf**:

7.2.1.1. acl Statement

The **acl** (Access Control List) statement defines groups of hosts which can then be permitted or denied access to the nameserver.

An **acl** statement takes the following form:

```
acl <acl-name> { <match-element>; [<match-element>; ...] };
```

In this statement, replace *<acl-name>* with the name of the access control list and replace *<match-element>* with a semi-colon separated list of IP addresses. Most of the time, an individual IP address or CIDR network notation (such as **10.0.1.0/24**) is used to identify the IP addresses within the **acl** statement.

The following access control lists are already defined as keywords to simplify configuration:

- **any** — Matches every IP address
- **localhost** — Matches any IP address in use by the local system
- **localnets** — Matches any IP address on any network to which the local system is connected
- **none** — Matches no IP addresses

When used in conjunction with other statements (such as the **`options`** statement), **`acl`** statements can be very useful in preventing the misuse of a BIND nameserver.

The following example defines two access control lists and uses an **`options`** statement to define how they are treated by the nameserver:

```
 acl black-hats {
 10.0.2.0/24;     192.168.0.0/24;     1234:5678::9abc/24;};
 acl red-hats {     10.0.1.0/24;  };
options {
 blackhole { black-hats; };
 allow-query { red-hats; };
 allow-query-cache { red-hats; };
}
```

This example contains two access control lists, **`black-hats`** and **`red-hats`**. Hosts in the **`black-hats`** list are denied access to the nameserver, while hosts in the **`red-hats`** list are given normal access.

7.2.1.2. `include` Statement

The **`include`** statement allows files to be included in a **`named.conf`** file. In this way, sensitive configuration data (such as **`keys`**) can be placed in a separate file with restrictive permissions.

An **`include`** statement takes the following form:

```
include "<file-name>"
```

In this statement, <*file-name*> is replaced with an absolute path to a file.

7.2.1.3. `options` Statement

The **`options`** statement defines global server configuration options and sets defaults for other statements. It can be used to specify the location of the **`named`** working directory, the types of queries allowed, and much more.

The **`options`** statement takes the following form:

```
options { <option>; [<option>; ...] };
```

In this statement, the <*option*> directives are replaced with a valid option.

The following are commonly used options:

`allow-query`

Specifies which hosts are allowed to query this nameserver for authoritative RRs. By default, all hosts are allowed to query. An access control lists, or collection of IP addresses or networks, may be used here to allow only particular hosts to query the nameserver.

allow-query-cache

Similar to **allow-query**, this option applies to non-authoritative data, like recursive queries. By default, only **localhost;** and **localnets;** are allowed to obtain non-authoritative data.

blackhole

Specifies which hosts are banned from the server. This option should be used when particular host or network floods the server with requests. Default is **none;**

directory

Specifies the **named** working directory if different from the default value, **/var/named/**.

forwarders

Specifies a list of valid IP addresses for nameservers where requests should be forwarded for resolution.

forward

Specifies the forwarding behavior of a **forwarders** directive.

The following options are accepted:

- **first** — Specifies that the nameservers listed in the **forwarders** directive be queried before **named** attempts to resolve the name itself.
- **only** — Specifies that **named** does not attempt name resolution itself in the event that queries to nameservers specified in the **forwarders** directive fail.

listen-on

Specifies the IPv4 network interface on which **named** listens for queries. By default, all IPv4 interfaces are used.

Using this directive on a DNS server which also acts a gateway, BIND can be configured to only answer queries that originate from one of the networks.

The following is an example of a **listen-on** directive:

```
options { listen-on { 10.0.1.1; }; };
```

In this example, server listens only on (**10.0.1.1**) address.

listen-on-v6

Same as **listen-on** except for IPv6 interfaces.

The following is an example of a **listen-on-v6** directive:

```
options { listen-on-v6 { 1234:5678::9abc; }; };
```

In this example, server listens only on (**1234:5678::9abc**) address.

max-cache-size

Specifies the maximum amount of memory to use for server caches. When the amount of data in the cache reaches this limit, the server will cause records to expire prematurely so that the limit is not exceeded. In a server with multiple views, the limit applies separately to the cache of each view. Default is 32M.

```
options { max-cache-size 256M; };
```

`notify`
Controls whether **`named`** notifies the slave servers when a zone is updated. It accepts the following options:

- **`yes`** — Notifies slave servers.
- **`no`** — Does not notify slave servers.
- **`master-only`** - Send notify only when server is a master server for the zone.
- **`explicit`** — Only notifies slave servers specified in an **`also-notify`** list within a zone statement.

`pid-file`
Specifies the location of the process ID file created by **`named`**.

`recursion`
Specifies if **`named`** acts as a recursive server. The default is **`yes`**.

```
options { recursion no; };
```

`statistics-file`
Specifies an alternate location for statistics files. By default, **`named`** statistics are saved to the **`/var/named/named.stats`** file.

There are many other options also available, many of which rely upon one another to work properly. Refer to the *BIND 9 Administrator Reference Manual* referenced in *Section 7.7.1, "Installed Documentation"* and the **`named.conf`** man page for more details.

7.2.1.4. zone Statement

A **`zone`** statement defines the characteristics of a zone, such as the location of its configuration file and zone-specific options. This statement can be used to override the global **`options`** statements.

A **`zone`** statement takes the following form:

```
zone <zone-name><zone-class> { <zone-options>; [<zone-options>; ...] };
```

In this statement, *`<zone-name>`* is the name of the zone, *`<zone-class>`* is the optional class of the zone, and *`<zone-options>`* is a list of options characterizing the zone.

The *`<zone-name>`* attribute for the zone statement is particularly important. It is the default value assigned for the **`$ORIGIN`** directive used within the corresponding zone file located in the **`/var/`**

named/ directory. The **named** daemon appends the name of the zone to any non-fully qualified domain name listed in the zone file.

For example, if a **zone** statement defines the namespace for **example.com**, use **example.com** as the <*zone-name*> so it is placed at the end of hostnames within the **example.com** zone file.

For more information about zone files, refer to *Section 7.3, “Zone Files”*.

The most common **zone** statement options include the following:

allow-query
Specifies the clients that are allowed to request information about this zone. Setting of this option overrides global **allow-query** option. The default is to allow all query requests.

allow-transfer
Specifies the slave servers that are allowed to request a transfer of the zone's information. The default is to allow all transfer requests.

allow-update
Specifies the hosts that are allowed to dynamically update information in their zone. The default is to deny all dynamic update requests.

Be careful when allowing hosts to update information about their zone. Do not set IP addresses in this option unless the server is in the trusted network. Use TSIG key instead .

file
Specifies the name of the file in the **named** working directory that contains the zone's configuration data.

masters
Specifies the IP addresses from which to request authoritative zone information and is used only if the zone is defined as **typeslave**.

notify
Specifies whether or not **named** notifies the slave servers when a zone is updated. This option has same parameters as a global **notify** parameter.

type
Defines the type of zone.

Below is a list of valid options:

- **delegation-only** — Enforces the delegation status of infrastructure zones such as COM, NET, or ORG. Any answer that is received without an explicit or implicit delegation is treated as **NXDOMAIN**. This option is only applicable in TLDs or root zone files used in recursive or caching implementations.

- **forward** — Forwards all requests for information about this zone to other nameservers.

- **hint** — A special type of zone used to point to the root nameservers which resolve queries when a zone is not otherwise known. No configuration beyond the default is necessary with a **hint** zone.

- **master** — Designates the nameserver as authoritative for this zone. A zone should be set as the **master** if the zone's configuration files reside on the system.

- **slave** — Designates the nameserver as a slave server for this zone. Master server is specified in **masters** directive.

7.2.1.5. Sample zone Statements

Most changes to the **/etc/named.conf** file of a master or slave nameserver involves adding, modifying, or deleting **zone** statements. While these **zone** statements can contain many options, most nameservers require only a small subset to function efficiently. The following **zone** statements are very basic examples illustrating a master-slave nameserver relationship.

The following is an example of a **zone** statement for the primary nameserver hosting **example.com** (**192.168.0.1**):

```
zone "example.com" IN { type master; file "example.com.zone"; allow-
transfer { 192.168.0.2; }; };
```

In the statement, the zone is identified as **example.com**, the type is set to **master**, and the **named** service is instructed to read the **/var/named/example.com.zone** file. It also allows only slave nameserver (**192.168.0.2**) to transfer the zone.

A slave server's **zone** statement for **example.com** is slightly different from the previous example. For a slave server, the type is set to **slave** and the **masters** directive is telling **named** the IP address of the master server.

The following is an example slave server **zone** statement for **example.com** zone:

```
zone "example.com" { type slave; file "slaves/example.com.zone"; masters
 { 192.168.0.1; }; };
```

This **zone** statement configures **named** on the slave server to query the master server at the **192.168.0.1** IP address for information about the **example.com** zone. The information that the slave server receives from the master server is saved to the **/var/named/slaves/example.com.zone** file. Make sure you put all slave zones to **/var/named/slaves** directory otherwise **named** will fail to transfer the zone.

7.2.2. Other Statement Types

The following is a list of lesser used statement types available within **named.conf**:

controls

Configures various security requirements necessary to use the **rndc** command to administer the **named** service.

Refer to *Section 7.4.1, “Configuring **/etc/named.conf**”* to learn more about how the **controls** statement is structured and what options are available.

key "*<key-name>*"

Defines a particular key by name. Keys are used to authenticate various actions, such as secure updates or the use of the **rndc** command. Two options are used with **key**:

- **`algorithm <*algorithm-name*>`** — The type of algorithm used, such as **`hmac-md5`**.
- **`secret "<*key-value*>"`** — The encrypted key.

Refer to *Section 7.4.2, “Configuring **`/etc/rndc.conf`**”* for instructions on how to write a **`key`** statement.

`logging`

Allows for the use of multiple types of logs, called *channels*. By using the **`channel`** option within the **`logging`** statement, a customized type of log can be constructed — with its own file name (**`file`**), size limit (**`size`**), versioning (**`version`**), and level of importance (**`severity`**). Once a customized channel is defined, a **`category`** option is used to categorize the channel and begin logging when **`named`** is restarted.

By default, **`named`** logs standard messages to the **`syslog`** daemon, which places them in **`/var/log/messages`**. This occurs because several standard channels are built into BIND with various severity levels, such as **`default_syslog`** (which handles informational logging messages) and **`default_debug`** (which specifically handles debugging messages). A default category, called **`default`**, uses the built-in channels to do normal logging without any special configuration.

Customizing the logging process can be a very detailed process and is beyond the scope of this chapter. For information on creating custom BIND logs, refer to the *BIND 9 Administrator Reference Manual* referenced in *Section 7.7.1, “Installed Documentation”*.

`server`

Specifies options that affect how **`named`** should respond to remote nameservers, especially with regard to notifications and zone transfers.

The **`transfer-format`** option controls whether one resource record is sent with each message (**`one-answer`**) or multiple resource records are sent with each message (**`many-answers`**). While **`many-answers`** is more efficient, only newer BIND nameservers understand it.

`trusted-keys`

Contains assorted public keys used for secure DNS (DNSSEC). Refer to *Section 7.5.3, “Security”* for more information concerning BIND security.

`view "<*view-name*>"`

Creates special views depending upon which network the host querying the nameserver is on. This allows some hosts to receive one answer regarding a zone while other hosts receive totally different information. Alternatively, certain zones may only be made available to particular trusted hosts while non-trusted hosts can only make queries for other zones.

Multiple views may be used, but their names must be unique. The **`match-clients`** option specifies the IP addresses that apply to a particular view. Any **`options`** statement may also be used within a view, overriding the global options already configured for **`named`**. Most **`view`** statements contain multiple **`zone`** statements that apply to the **`match-clients`** list. The order in which **`view`** statements are listed is important, as the first **`view`** statement that matches a particular client's IP address is used.

Refer to *Section 7.5.2, “Multiple Views”* for more information about the **`view`** statement.

7.2.3. Comment Tags

The following is a list of valid comment tags used within **`named.conf`**:

- `//` — When placed at the beginning of a line, that line is ignored by **`named`**.
- `#` — When placed at the beginning of a line, that line is ignored by **`named`**.
- `/*` and `*/` — When text is enclosed in these tags, the block of text is ignored by **`named`**.

7.3. Zone Files

Zone files contain information about a namespace and are stored in the **`named`** working directory (**`/var/named/`**) by default. Each zone file is named according to the **`file`** option data in the **`zone`** statement, usually in a way that relates to the domain in question and identifies the file as containing zone data, such as **`example.com.zone`**.

> **Note**
>
> If you have installed the **`bind-chroot`** package, the BIND service will run in the **`/var/named/chroot`** environment. All configuration files will be moved there. As such, you can find the zone files in **`/var/named/chroot/var/named`**.

Each zone file may contain *directives* and *resource records*. Directives tell the nameserver to perform tasks or apply special settings to the zone. Resource records define the parameters of the zone and assign identities to individual hosts. Directives are optional, but resource records are required to provide name service to a zone.

All directives and resource records should be entered on individual lines.

Comments can be placed after semicolon characters (`;`) in zone files.

7.3.1. Zone File Directives

Directives begin with the dollar sign character (**`$`**) followed by the name of the directive. They usually appear at the top of the zone file.

The following are commonly used directives:

`$INCLUDE`

Configures **named** to include another zone file in this zone file at the place where the directive appears. This allows additional zone settings to be stored apart from the main zone file.

`$ORIGIN`

Appends the domain name to unqualified records, such as those with the hostname and nothing more.

For example, a zone file may contain the following line:

```
$ORIGIN example.com.
```

Any names used in resource records that do not end in a trailing period (`.`) are appended with **`example.com`**.

Note

The use of the **$ORIGIN** directive is unnecessary if the zone is specified in **/etc/named.conf** because the zone name is used as the value for the **$ORIGIN** directive by default.

$TTL

Sets the default *Time to Live (TTL)* value for the zone. This is the length of time, in seconds, that a zone resource record is valid. Each resource record can contain its own TTL value, which overrides this directive.

Increasing this value allows remote nameservers to cache the zone information for a longer period of time, reducing the number of queries for the zone and lengthening the amount of time required to proliferate resource record changes.

7.3.2. Zone File Resource Records

The primary component of a zone file is its resource records.

There are many types of zone file resource records. The following are used most frequently:

A

This refers to the Address record, which specifies an IP address to assign to a name, as in this example:

```
<host> IN A <IP-address>
```

If the *<host>* value is omitted, then an **A** record points to a default IP address for the top of the namespace. This system is the target for all non-FQDN requests.

Consider the following **A** record examples for the **example.com** zone file:

```
server1 IN A 10.0.1.3
  IN A 10.0.1.5
```

Requests for **example.com** are pointed to 10.0.1.3 or 10.0.1.5.

CNAME

This refers to the Canonical Name record, which maps one name to another. This type of record can also be referred to as an *alias record*.

The next example tells **named** that any requests sent to the *<alias-name>* should point to the host, *<real-name>*. **CNAME** records are most commonly used to point to services that use a common naming scheme, such as **www** for Web servers.

```
<alias-name> IN CNAME <real-name>
```

In the following example, an **A** record binds a hostname to an IP address, while a **CNAME** record points the commonly used **www** hostname to it.

```
server1 IN A 10.0.1.5
              www IN CNAME server1
```

MX

This refers to the Mail eXchange record, which tells where mail sent to a particular namespace controlled by this zone should go.

```
 IN MX <preference-value><email-server-name>
```

Here, the <*preference-value*> allows numerical ranking of the email servers for a namespace, giving preference to some email systems over others. The **MX** resource record with the lowest <*preference-value*> is preferred over the others. However, multiple email servers can possess the same value to distribute email traffic evenly among them.

The <*email-server-name*> may be a hostname or FQDN.

```
IN MX 10 mail.example.com.
              IN MX 20 mail2.example.com.
```

In this example, the first **mail.example.com** email server is preferred to the **mail2.example.com** email server when receiving email destined for the **example.com** domain.

NS

This refers to the NameServer record, which announces the authoritative nameservers for a particular zone.

The following illustrates the layout of an **NS** record:

```
 IN NS <nameserver-name>
```

Here, <*nameserver-name*> should be an FQDN.

Next, two nameservers are listed as authoritative for the domain. It is not important whether these nameservers are slaves or if one is a master; they are both still considered authoritative.

```
IN NS dns1.example.com.
              IN NS dns2.example.com.
```

PTR

This refers to the PoinTeR record, which is designed to point to another part of the namespace.

PTR records are primarily used for reverse name resolution, as they point IP addresses back to a particular name. Refer to *Section 7.3.4, “Reverse Name Resolution Zone Files”* for more examples of **PTR** records in use.

SOA

This refers to the Start Of Authority resource record, which proclaims important authoritative information about a namespace to the nameserver.

Located after the directives, an **SOA** resource record is the first resource record in a zone file.

The following shows the basic structure of an **SOA** resource record:

```
@     IN     SOA    <primary-name-server>
              <hostmaster-email> (
 <serial-number>
              <time-to-refresh>
              <time-to-retry>
              <time-to-expire>
              <minimum-TTL> )
```

The `@` symbol places the **$ORIGIN** directive (or the zone's name, if the **$ORIGIN** directive is not set) as the namespace being defined by this **SOA** resource record. The hostname of the primary nameserver that is authoritative for this domain is the *<primary-name-server>* directive, and the email of the person to contact about this namespace is the *<hostmaster-email>* directive.

The *<serial-number>* directive is a numerical value incremented every time the zone file is altered to indicate it is time for **named** to reload the zone. The *<time-to-refresh>* directive is the numerical value slave servers use to determine how long to wait before asking the master nameserver if any changes have been made to the zone. The *<serial-number>* directive is a numerical value used by the slave servers to determine if it is using outdated zone data and should therefore refresh it.

The *<time-to-retry>* directive is a numerical value used by slave servers to determine the length of time to wait before issuing a refresh request in the event that the master nameserver is not answering. If the master has not replied to a refresh request before the amount of time specified in the *<time-to-expire>* directive elapses, the slave servers stop responding as an authority for requests concerning that namespace.

In BIND 4 and 8, the *<minimum-TTL>* directive is the amount of time other nameservers cache the zone's information. However, in BIND 9, the *<minimum-TTL>* directive defines how long negative answers are cached for. Caching of negative answers can be set to a maximum of 3 hours (`3H`).

When configuring BIND, all times are specified in seconds. However, it is possible to use abbreviations when specifying units of time other than seconds, such as minutes (**M**), hours (**H**), days (**D**), and weeks (**W**). The table in *Table 7.1, “Seconds compared to other time units”* shows an amount of time in seconds and the equivalent time in another format.

Seconds	Other Time Units
60	**1M**
1800	**30M**

Seconds	Other Time Units
3600	**1H**
10800	**3H**
21600	**6H**
43200	**12H**
86400	**1D**
259200	**3D**
604800	**1W**
31536000	**365D**

Table 7.1. Seconds compared to other time units

The following example illustrates the form an **SOA** resource record might take when it is populated with real values.

```
@     IN     SOA    dns1.example.com.     hostmaster.example.com. (
   2001062501 ; serial
   21600      ; refresh after 6 hours
   3600       ; retry after 1 hour
   604800     ; expire after 1 week
   86400 )    ; minimum TTL of 1 day
```

7.3.3. Example Zone File

Seen individually, directives and resource records can be difficult to grasp. However, when placed together in a single file, they become easier to understand.

The following example shows a very basic zone file.

```
$ORIGIN example.com.
$TTL 86400
@ SOA dns1.example.com. hostmaster.example.com. (
  2001062501 ; serial
  21600      ; refresh after 6 hours
  3600       ; retry after 1 hour
  604800     ; expire after 1 week
  86400 )    ; minimum TTL of 1 day
;
;
 NS dns1.example.com.
 NS dns2.example.com.
dns1 A 10.0.1.1
 AAAA aaaa:bbbb::1
dns2 A 10.0.1.2
 AAAA aaaa:bbbb::2
;
;
@ MX 10 mail.example.com.
```

```
 MX 20 mail2.example.com.
mail A 10.0.1.5
 AAAA aaaa:bbbb::5
mail2 A 10.0.1.6
 AAAA aaaa:bbbb::6
;
;
; This sample zone file illustrates sharing the same IP addresses for
 multiple services:
;
services A 10.0.1.10
  AAAA aaaa:bbbb::10
  A 10.0.1.11
  AAAA aaaa:bbbb::11

ftp CNAME services.example.com.
www CNAME services.example.com.
;
;
```

In this example, standard directives and **SOA** values are used. The authoritative nameservers are set as **dns1.example.com** and **dns2.example.com**, which have **A** records that tie them to **10.0.1.1** and **10.0.1.2**, respectively.

The email servers configured with the **MX** records point to **mail** and **mail2** via **A** records. Since the **mail** and **mail2** names do not end in a trailing period (**.**), the **$ORIGIN** domain is placed after them, expanding them to **mail.example.com** and **mail2.example.com**. Through the related **A** resource records, their IP addresses can be determined.

Services available at the standard names, such as **www.example.com** (WWW), are pointed at the appropriate servers using a **CNAME** record.

This zone file would be called into service with a **zone** statement in the **named.conf** similar to the following:

```

zone "example.com" IN {
 type master;
 file "example.com.zone";
 allow-update { none; };
};
```

7.3.4. Reverse Name Resolution Zone Files

A reverse name resolution zone file is used to translate an IP address in a particular namespace into an FQDN. It looks very similar to a standard zone file, except that **PTR** resource records are used to link the IP addresses to a fully qualified domain name.

The following illustrates the layout of a **PTR** record:

```
<last-IP-digit> IN PTR <FQDN-of-system>
```

The `<last-IP-digit>` is the last number in an IP address which points to a particular system's FQDN.

In the following example, IP addresses **`10.0.1.1`** through **`10.0.1.6`** are pointed to corresponding FQDNs. It can be located in **`/var/named/example.com.rr.zone`**.

```
$ORIGIN 1.0.10.in-addr.arpa.
$TTL 86400
@ IN SOA dns1.example.com. hostmaster.example.com. (
   2001062501 ; serial
   21600      ; refresh after 6 hours
   3600       ; retry after 1 hour
   604800     ; expire after 1 week
   86400 )    ; minimum TTL of 1 day
;
;
1 IN PTR dns1.example.com.
2 IN PTR dns2.example.com.
;
5 IN PTR    server1.example.com.
6 IN PTR    server2.example.com.
;
3 IN PTR    ftp.example.com.
4 IN PTR    ftp.example.com.
```

This zone file would be called into service with a **`zone`** statement in the **`named.conf`** file similar to the following:

```
zone "1.0.10.in-addr.arpa" IN {
 type master;
 file "example.com.rr.zone";
 allow-update { none; };
};
```

There is very little difference between this example and a standard **`zone`** statement, except for the zone name. Note that a reverse name resolution zone requires the first three blocks of the IP address reversed followed by **`.in-addr.arpa`**. This allows the single block of IP numbers used in the reverse name resolution zone file to be associated with the zone.

7.4. Using `rndc`

BIND includes a utility called **`rndc`** which allows command line administration of the **`named`** daemon from the localhost or a remote host.

In order to prevent unauthorized access to the **`named`** daemon, BIND uses a shared secret key authentication method to grant privileges to hosts. This means an identical key must be present in both **`/etc/named.conf`** and the **`rndc`** configuration file, **`/etc/rndc.conf`**.

Note

If you have installed the **bind-chroot** package, the BIND service will run in the **/var/named/chroot** environment. All configuration files will be moved there. As such, the **rndc.conf** file is located in **/var/named/chroot/etc/rndc.conf**.

Note that since the **rndc** utility does not run in a **chroot** environment, **/etc/rndc.conf** is a symlink to **/var/named/chroot/etc/rndc.conf**.

7.4.1. Configuring /etc/named.conf

In order for **rndc** to connect to a **named** service, there must be a **controls** statement in the BIND server's **/etc/named.conf** file.

The **controls** statement, shown in the following example, allows **rndc** to connect from the localhost.

```
controls {
 inet 127.0.0.1
  allow { localhost; } keys { <key-name>; };
};
```

This statement tells **named** to listen on the default TCP port 953 of the loopback address and allow **rndc** commands coming from the localhost, if the proper key is given. The *<key-name>* specifies a name in the **key** statement within the **/etc/named.conf** file. The next example illustrates a sample **key** statement.

```
key "<key-name>" {
 algorithm hmac-md5;
 secret "<key-value>";
};
```

In this case, the *<key-value>* uses the HMAC-MD5 algorithm. Use the following command to generate keys using the HMAC-MD5 algorithm:

```
dnssec-keygen -a hmac-md5 -b <bit-length> -n HOST <key-file-name>
```

A key with at least a 256-bit length is a good idea. The actual key that should be placed in the *<key-value>* area can be found in the ***<key-file-name>*** file generated by this command.

Warning

Because **/etc/named.conf** is world-readable, it is advisable to place the **key** statement in a separate file, readable only by root, and then use an **include** statement to reference it. For example:

```
include "/etc/rndc.key";
```

7.4.2. Configuring /etc/rndc.conf

The **key** is the most important statement in **/etc/rndc.conf**.

```
key "<key-name>" {
 algorithm hmac-md5;
 secret "<key-value>";
};
```

The *<key-name>* and *<key-value>* should be exactly the same as their settings in **/etc/named.conf**.

To match the keys specified in the target server's **/etc/named.conf**, add the following lines to **/etc/rndc.conf**.

```
options {
 default-server  localhost;
 default-key     "<key-name>";
};
```

This directive sets a global default key. However, the **rndc** configuration file can also specify different keys for different servers, as in the following example:

```
server localhost {
 key  "<key-name>";
};
```

Important

Make sure that only the root user can read or write to the **/etc/rndc.conf** file.

For more information about the **/etc/rndc.conf** file, refer to the **rndc.conf** man page.

7.4.3. Command Line Options

An **rndc** command takes the following form:

```
rndc <options><command><command-options>
```

When executing **rndc** on a properly configured localhost, the following commands are available:

- **`halt`** — Stops the **`named`** service immediately.
- **`querylog`** — Logs all queries made to this nameserver.
- **`refresh`** — Refreshes the nameserver's database.
- **`reload`** — Reloads the zone files but keeps all other previously cached responses. This command also allows changes to zone files without losing all stored name resolutions.

  If changes made only affect a specific zone, reload only that specific zone by adding the name of the zone after the **`reload`** command.
- **`stats`** — Dumps the current **`named`** statistics to the **`/var/named/named.stats`** file.
- **`stop`** — Stops the server gracefully, saving any dynamic update and *Incremental Zone Transfers* (*IXFR*) data before exiting.

Occasionally, it may be necessary to override the default settings in the **`/etc/rndc.conf`** file. The following options are available:

- **`-c <configuration-file>`** — Specifies the alternate location of a configuration file.
- **`-p <port-number>`** — Specifies a port number to use for the **`rndc`** connection other than the default port 953.
- **`-s <server>`** — Specifies a server other than the **`default-server`** listed in **`/etc/rndc.conf`**.
- **`-y <key-name>`** — Specifies a key other than the **`default-key`** option in **`/etc/rndc.conf`**.

Additional information about these options can be found in the **`rndc`** man page.

7.5. Advanced Features of BIND

Most BIND implementations only use **`named`** to provide name resolution services or to act as an authority for a particular domain or sub-domain. However, BIND version 9 has a number of advanced features that allow for a more secure and efficient DNS service.

> **Caution**
>
> Some of these advanced features, such as DNSSEC, TSIG, and IXFR (which are defined in the following section), should only be used in network environments with nameservers that support the features. If the network environment includes non-BIND or older BIND nameservers, verify that each advanced feature is supported before attempting to use it.

All of the features mentioned are discussed in greater detail in the *BIND 9 Administrator Reference Manual* referenced in *Section 7.7.1, “Installed Documentation”*.

7.5.1. DNS Protocol Enhancements

BIND supports Incremental Zone Transfers (IXFR), where a slave nameserver only downloads the updated portions of a zone modified on a master nameserver. The standard transfer process requires that the entire zone be transferred to each slave nameserver for even the smallest change. For very popular domains with very lengthy zone files and many slave nameservers, IXFR makes the notification and update process much less resource-intensive.

Note that IXFR is only available when using *dynamic updating* to make changes to master zone records. If manually editing zone files to make changes, Automatic Zone Transfer (AXFR) is used. More information on dynamic updating is available in the *BIND 9 Administrator Reference Manual* referenced in *Section 7.7.1, "Installed Documentation"*.

7.5.2. Multiple Views

Through the use of the **`view`** statement in **`named.conf`**, BIND can present different information depending on which network a request originates from.

This is primarily used to deny sensitive DNS entries from clients outside of the local network, while allowing queries from clients inside the local network.

The **`view`** statement uses the **`match-clients`** option to match IP addresses or entire networks and give them special options and zone data.

7.5.3. Security

BIND supports a number of different methods to protect the updating and transfer of zones, on both master and slave nameservers:

DNSSEC

Short for *DNS SECurity*, this feature allows for zones to be cryptographically signed with a *zone key*.

In this way, the information about a specific zone can be verified as coming from a nameserver that has signed it with a particular private key, as long as the recipient has that nameserver's public key.

BIND version 9 also supports the SIG(0) public/private key method of message authentication.

TSIG

Short for *Transaction SIGnatures*, this feature allows a transfer from master to slave only after verifying that a shared secret key exists on both nameservers.

This feature strengthens the standard IP address-based method of transfer authorization. An attacker would not only need to have access to the IP address to transfer the zone, but they would also need to know the secret key.

BIND version 9 also supports *TKEY*, which is another shared secret key method of authorizing zone transfers.

7.5.4. IP version 6

BIND version 9 supports name service in IP version 6 (IPv6) environments through the use of **`A6`** zone records.

If the network environment includes both IPv4 and IPv6 hosts, use the **`lwresd`** lightweight resolver daemon on all network clients. This daemon is a very efficient, caching-only nameserver which understands the new **`A6`** and **`DNAME`** records used under IPv6. Refer to the **`lwresd`** man page for more information.

7.6. Common Mistakes to Avoid

It is very common for beginners to make mistakes when editing BIND configuration files. Be sure to avoid the following issues:

- *Take care to increment the serial number when editing a zone file.*

  If the serial number is not incremented, the master nameserver has the correct, new information, but the slave nameservers are never notified of the change and do not attempt to refresh their data of that zone.

- *Be careful to use ellipses and semi-colons correctly in the* ***`/etc/named.conf`*** *file.*

  An omitted semi-colon or unclosed ellipse section can cause **`named`** to refuse to start.

- *Remember to place periods (`.`) in zone files after all FQDNs and omit them on hostnames.*

  A period at the end of a domain name denotes a fully qualified domain name. If the period is omitted, then **`named`** appends the name of the zone or the **`$ORIGIN`** value to complete it.

- *If a firewall is blocking connections from the* ***`named`*** *program to other nameservers, edit its configuration file.*

  By default, BIND version 9 uses random ports above 1024 to query other nameservers. Some firewalls, however, expect all nameservers to communicate using only port 53. To force **`named`** to use port 53, add the following line to the **`options`** statement of **`/etc/named.conf`**:

  ```
  query-source address * port 53;
  ```

7.7. Additional Resources

The following sources of information provide additional resources regarding BIND.

7.7.1. Installed Documentation

BIND features a full range of installed documentation covering many different topics, each placed in its own subject directory. For each item below, replace <*`version-number`*> with the version of **`bind`** installed on the system:

`/usr/share/doc/bind-<`*`version-number`*`>/`
: This directory lists the most recent features.

`/usr/share/doc/bind-<`*`version-number`*`>/arm/`
: This directory contains the *BIND 9 Administrator Reference Manual* in HTML and SGML formats, which details BIND resource requirements, how to configure different types of nameservers, how to perform load balancing, and other advanced topics. For most new users of BIND, this is the best place to start.

`/usr/share/doc/bind-<`*`version-number`*`>/draft/`
: This directory contains assorted technical documents that review issues related to DNS service and propose some methods to address them.

`/usr/share/doc/bind-<version-number>/misc/`

This directory contains documents designed to address specific advanced issues. Users of BIND version 8 should consult the **`migration`** document for specific changes they must make when moving to BIND 9. The **`options`** file lists all of the options implemented in BIND 9 that are used in **`/etc/named.conf`**.

`/usr/share/doc/bind-<version-number>/rfc/`

This directory provides every RFC document related to BIND.

There are also a number of man pages for the various applications and configuration files involved with BIND. The following lists some of the more important man pages.

Administrative Applications

- **`man rndc`** — Explains the different options available when using the **`rndc`** command to control a BIND nameserver.

Server Applications

- **`man named`** — Explores assorted arguments that can be used to control the BIND nameserver daemon.
- **`man lwresd`** — Describes the purpose of and options available for the lightweight resolver daemon.

Configuration Files

- **`man named.conf`** — A comprehensive list of options available within the **`named`** configuration file.
- **`man rndc.conf`** — A comprehensive list of options available within the **`rndc`** configuration file.

7.7.2. Useful Websites

- *http://www.isc.org/index.pl?/sw/bind/* — The home page of the BIND project containing information about current releases as well as a PDF version of the *BIND 9 Administrator Reference Manual*.
- *http://www.redhat.com/mirrors/LDP/HOWTO/DNS-HOWTO.html* — Covers the use of BIND as a resolving, caching nameserver and the configuration of various zone files necessary to serve as the primary nameserver for a domain.

7.7.3. Related Books

- *DNS and BIND* by Paul Albitz and Cricket Liu; O'Reilly & Associates — A popular reference that explains both common and esoteric BIND configuration options, as well as providing strategies for securing a DNS server.
- *The Concise Guide to DNS and BIND* by Nicolai Langfeldt; Que — Looks at the connection between multiple network services and BIND, with an emphasis on task-oriented, technical topics.

OpenSSH

SSH™ (or *S*ecure *SH*ell) is a protocol which facilitates secure communications between two systems using a client/server architecture and allows users to log into server host systems remotely. Unlike other remote communication protocols, such as FTP or Telnet, SSH encrypts the login session, rendering the connection difficult for intruders to collect unencrypted passwords.

SSH is designed to replace older, less secure terminal applications used to log into remote hosts, such as **`telnet`** or **`rsh`**. A related program called **`scp`** replaces older programs designed to copy files between hosts, such as **`rcp`**. Because these older applications do not encrypt passwords transmitted between the client and the server, avoid them whenever possible. Using secure methods to log into remote systems decreases the risks for both the client system and the remote host.

8.1. Features of SSH

The SSH protocol provides the following safeguards:

- After an initial connection, the client can verify that it is connecting to the same server it had connected to previously.
- The client transmits its authentication information to the server using strong, 128-bit encryption.
- All data sent and received during a session is transferred using 128-bit encryption, making intercepted transmissions extremely difficult to decrypt and read.
- The client can forward X11[1] applications from the server. This technique, called *X11 forwarding*, provides a secure means to use graphical applications over a network.

Because the SSH protocol encrypts everything it sends and receives, it can be used to secure otherwise insecure protocols. Using a technique called *port forwarding*, an SSH server can become a conduit to securing otherwise insecure protocols, like POP, and increasing overall system and data security.

The OpenSSH server and client can also be configured to create a tunnel similar to a virtual private network for traffic between server and client machines.

Finally, OpenSSH servers and clients can be configured to authenticate using the GSSAPI implementation of the Kerberos network authentication protocol. For more information on configuring Kerberos authentication services, refer to .

Fedora includes the general OpenSSH package (**`openssh`**) as well as the OpenSSH server (**`openssh-server`**) and client (**`openssh-clients`**) packages. Note, the OpenSSH packages require the OpenSSL package (**`openssl`**) which installs several important cryptographic libraries, enabling OpenSSH to provide encrypted communications.

8.1.1. Why Use SSH?

Nefarious computer users have a variety of tools at their disposal enabling them to disrupt, intercept, and re-route network traffic in an effort to gain access to a system. In general terms, these threats can be categorized as follows:

- *Interception of communication between two systems* — In this scenario, the attacker can be somewhere on the network between the communicating parties, copying any information passed between them. The attacker may intercept and keep the information, or alter the information and send it on to the intended recipient.

This attack can be mounted through the use of a packet sniffer — a common network utility.

- *Impersonation of a particular host* — Using this strategy, an attacker's system is configured to pose as the intended recipient of a transmission. If this strategy works, the user's system remains unaware that it is communicating with the wrong host.

  This attack can be mounted through techniques known as DNS poisoning[2] or IP spoofing[3].

Both techniques intercept potentially sensitive information and, if the interception is made for hostile reasons, the results can be disastrous.

If SSH is used for remote shell login and file copying, these security threats can be greatly diminished. This is because the SSH client and server use digital signatures to verify their identity. Additionally, all communication between the client and server systems is encrypted. Attempts to spoof the identity of either side of a communication does not work, since each packet is encrypted using a key known only by the local and remote systems.

8.2. SSH Protocol Versions

The SSH protocol allows any client and server programs built to the protocol's specifications to communicate securely and to be used interchangeably.

Two varieties of SSH (version 1 and version 2) currently exist. The OpenSSH suite under Fedora uses SSH version 2 which has an enhanced key exchange algorithm not vulnerable to the exploit in version 1. However, the OpenSSH suite does support version 1 connections.

> **Important**
>
> It is recommended that only SSH version 2-compatible servers and clients are used whenever possible.

8.3. Event Sequence of an SSH Connection

The following series of events help protect the integrity of SSH communication between two hosts.

1. A cryptographic handshake is made so that the client can verify that it is communicating with the correct server.

2. The transport layer of the connection between the client and remote host is encrypted using a symmetric cipher.

3. The client authenticates itself to the server.

4. The remote client interacts with the remote host over the encrypted connection.

8.3.1. Transport Layer

The primary role of the transport layer is to facilitate safe and secure communication between the two hosts at the time of authentication and during subsequent communication. The transport layer accomplishes this by handling the encryption and decryption of data, and by providing integrity protection of data packets as they are sent and received. The transport layer also provides compression, speeding the transfer of information.

Once an SSH client contacts a server, key information is exchanged so that the two systems can correctly construct the transport layer. The following steps occur during this exchange:

- Keys are exchanged
- The public key encryption algorithm is determined
- The symmetric encryption algorithm is determined
- The message authentication algorithm is determined
- The hash algorithm is determined

During the key exchange, the server identifies itself to the client with a unique *host key*. If the client has never communicated with this particular server before, the server's host key is unknown to the client and it does not connect. OpenSSH gets around this problem by accepting the server's host key. This is done after the user is notified and has both accepted and verified the new host key. In subsequent connections, the server's host key is checked against the saved version on the client, providing confidence that the client is indeed communicating with the intended server. If, in the future, the host key no longer matches, the user must remove the client's saved version before a connection can occur.

Caution

It is possible for an attacker to masquerade as an SSH server during the initial contact since the local system does not know the difference between the intended server and a false one set up by an attacker. To help prevent this, verify the integrity of a new SSH server by contacting the server administrator before connecting for the first time or in the event of a host key mismatch.

SSH is designed to work with almost any kind of public key algorithm or encoding format. After an initial key exchange creates a hash value used for exchanges and a shared secret value, the two systems immediately begin calculating new keys and algorithms to protect authentication and future data sent over the connection.

After a certain amount of data has been transmitted using a given key and algorithm (the exact amount depends on the SSH implementation), another key exchange occurs, generating another set of hash values and a new shared secret value. Even if an attacker is able to determine the hash and shared secret value, this information is only useful for a limited period of time.

8.3.2. Authentication

Once the transport layer has constructed a secure tunnel to pass information between the two systems, the server tells the client the different authentication methods supported, such as using a private key-encoded signature or typing a password. The client then tries to authenticate itself to the server using one of these supported methods.

SSH servers and clients can be configured to allow different types of authentication, which gives each side the optimal amount of control. The server can decide which encryption methods it supports based on its security model, and the client can choose the order of authentication methods to attempt from the available options.

8.3.3. Channels

After a successful authentication over the SSH transport layer, multiple channels are opened via a technique called *multiplexing*[4]. Each of these channels handles communication for different terminal sessions and for forwarded X11 sessions.

Both clients and servers can create a new channel. Each channel is then assigned a different number on each end of the connection. When the client attempts to open a new channel, the clients sends the channel number along with the request. This information is stored by the server and is used to direct communication to that channel. This is done so that different types of sessions do not affect one another and so that when a given session ends, its channel can be closed without disrupting the primary SSH connection.

Channels also support *flow-control*, which allows them to send and receive data in an orderly fashion. In this way, data is not sent over the channel until the client receives a message that the channel is open.

The client and server negotiate the characteristics of each channel automatically, depending on the type of service the client requests and the way the user is connected to the network. This allows great flexibility in handling different types of remote connections without having to change the basic infrastructure of the protocol.

8.4. Configuring an OpenSSH Server

To run an OpenSSH server, you must first make sure that you have the proper RPM packages installed. The **openssh-server** package is required and is dependent on the **openssh** package.

The OpenSSH daemon uses the configuration file **/etc/ssh/sshd_config**. The default configuration file should be sufficient for most purposes. If you want to configure the daemon in ways not provided by the default **sshd_config**, read the **sshd** man page for a list of the keywords that can be defined in the configuration file.

To start the OpenSSH service, use the command **/sbin/service sshd start**. To stop the OpenSSH server, use the command **/sbin/service sshd stop**. If you want the daemon to start automatically at boot time, refer to *Chapter 6, Controlling Access to Services* for information on how to manage services.

If you reinstall, the reinstalled system creates a new set of identification keys. Any clients who had connected to the system with any of the OpenSSH tools before the reinstall will see the following message:

```

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
```

[4] A multiplexed connection consists of several signals being sent over a shared, common medium. With SSH, different channels are sent over a common secure connection.

```
It is also possible that the RSA host key has just been changed.
```

If you want to keep the host keys generated for the system, backup the **`/etc/ssh/ssh_host*key*`** files and restore them after the reinstall. This process retains the system's identity, and when clients try to connect to the system after the reinstall, they will not receive the warning message.

8.4.1. Requiring SSH for Remote Connections

For SSH to be truly effective, using insecure connection protocols, such as Telnet and FTP, should be prohibited. Otherwise, a user's password may be protected using SSH for one session, only to be captured later while logging in using Telnet.

Some services to disable include:

- **`telnet`**
- **`rsh`**
- **`rlogin`**
- **`vsftpd`**

To disable insecure connection methods to the system, use the command line program **`chkconfig`**, the ncurses-based program **/usr/sbin/ntsysv**, or the **Services Configuration Tool** (**`system-config-services`**) graphical application. All of these tools require root level access.

For more information on runlevels and configuring services with **`chkconfig`**, **/usr/sbin/ntsysv**, and the **Services Configuration Tool**, refer to *Chapter 6, Controlling Access to Services*.

8.5. OpenSSH Configuration Files

OpenSSH has two different sets of configuration files: one for client programs (**`ssh`**, **`scp`**, and **`sftp`**) and one for the server daemon (**`sshd`**).

System-wide SSH configuration information is stored in the **`/etc/ssh/`** directory:

- **`moduli`** — Contains Diffie-Hellman groups used for the Diffie-Hellman key exchange which is critical for constructing a secure transport layer. When keys are exchanged at the beginning of an SSH session, a shared, secret value is created which cannot be determined by either party alone. This value is then used to provide host authentication.
- **`ssh_config`** — The system-wide default SSH client configuration file. It is overridden if one is also present in the user's home directory (**`~/.ssh/config`**).
- **`sshd_config`** — The configuration file for the **`sshd`** daemon.
- **`ssh_host_dsa_key`** — The DSA private key used by the **`sshd`** daemon.
- **`ssh_host_dsa_key.pub`** — The DSA public key used by the **`sshd`** daemon.
- **`ssh_host_key`** — The RSA private key used by the **`sshd`** daemon for version 1 of the SSH protocol.
- **`ssh_host_key.pub`** — The RSA public key used by the **`sshd`** daemon for version 1 of the SSH protocol.

- **`ssh_host_rsa_key`** — The RSA private key used by the **`sshd`** daemon for version 2 of the SSH protocol.
- **`ssh_host_rsa_key.pub`** — The RSA public key used by the **`sshd`** for version 2 of the SSH protocol.

User-specific SSH configuration information is stored in the user's home directory within the **`~/.ssh/`** directory:

- **`authorized_keys`** — This file holds a list of authorized public keys for servers. When the client connects to a server, the server authenticates the client by checking its signed public key stored within this file.
- **`id_dsa`** — Contains the DSA private key of the user.
- **`id_dsa.pub`** — The DSA public key of the user.
- **`id_rsa`** — The RSA private key used by **`ssh`** for version 2 of the SSH protocol.
- **`id_rsa.pub`** — The RSA public key used by **`ssh`** for version 2 of the SSH protocol
- **`identity`** — The RSA private key used by **`ssh`** for version 1 of the SSH protocol.
- **`identity.pub`** — The RSA public key used by **`ssh`** for version 1 of the SSH protocol.
- **`known_hosts`** — This file contains DSA host keys of SSH servers accessed by the user. This file is very important for ensuring that the SSH client is connecting the correct SSH server.

> **Important**
>
> If an SSH server's host key has changed, the client notifies the user that the connection cannot proceed until the server's host key is deleted from the **`known_hosts`** file using a text editor. Before doing this, however, contact the system administrator of the SSH server to verify the server is not compromised.

Refer to the **`ssh_config`** and **`sshd_config`** man pages for information concerning the various directives available in the SSH configuration files.

8.6. Configuring an OpenSSH Client

To connect to an OpenSSH server from a client machine, you must have the **`openssh-clients`** and **`openssh`** packages installed on the client machine.

8.6.1. Using the ssh Command

The **`ssh`** command is a secure replacement for the **`rlogin`**, **`rsh`**, and **`telnet`** commands. It allows you to log in to a remote machine as well as execute commands on a remote machine.

Logging in to a remote machine with **`ssh`** is similar to using **`telnet`**. To log in to a remote machine named penguin.example.net, type the following command at a shell prompt:

```
ssh penguin.example.net
```

The first time you **ssh** to a remote machine, you will see a message similar to the following:

```
The authenticity of host 'penguin.example.net' can't be established.
DSA key fingerprint is 94:68:3a:3a:bc:f3:9a:9b:01:5d:b3:07:38:e2:11:0c.
Are you sure you want to continue connecting (yes/no)?
```

Type **yes** to continue. This will add the server to your list of known hosts (**~/.ssh/known_hosts**) as seen in the following message:

```
Warning: Permanently added 'penguin.example.net' (RSA) to the list of known
 hosts.
```

Next, you will see a prompt asking for your password for the remote machine. After entering your password, you will be at a shell prompt for the remote machine. If you do not specify a username the username that you are logged in as on the local client machine is passed to the remote machine. If you want to specify a different username, use the following command:

```
ssh username@penguin.example.net
```

You can also use the syntax **ssh -l *username* penguin.example.net**.

The **ssh** command can be used to execute a command on the remote machine without logging in to a shell prompt. The syntax is **ssh *hostnamecommand***. For example, if you want to execute the command **ls /usr/share/doc** on the remote machine penguin.example.net, type the following command at a shell prompt:

```
ssh penguin.example.net ls /usr/share/doc
```

After you enter the correct password, the contents of the remote directory **/usr/share/doc** will be displayed, and you will return to your local shell prompt.

8.6.2. Using the scp Command

The **scp** command can be used to transfer files between machines over a secure, encrypted connection. It is similar to **rcp**.

The general syntax to transfer a local file to a remote system is as follows:

```
scp <localfile>
        username@tohostname:<remotefile>
```

The <*localfile*> specifies the source including path to the file, such as **/var/log/maillog**. The <*remotefile*> specifies the destination, which can be a new filename such as **/tmp/hostname-**

maillog. For the remote system, if you do not have a preceding **/**, the path will be relative to the home directory of *username*, typically **/home/username/**.

To transfer the local file **shadowman** to the home directory of your account on penguin.example.net, type the following at a shell prompt (replace *username* with your username):

```
scp shadowman username@penguin.example.net:shadowman
```

This will transfer the local file **shadowman** to **/home/*username*/shadowman** on penguin.example.net. Alternately, you can leave off the final shadowman in the **scp** command.

The general syntax to transfer a remote file to the local system is as follows:

```
scp username@tohostname:<remotefile>
        <newlocalfile>
```

The <*remotefile*> specifies the source including path, and <*newlocalfile*> specifies the destination including path.

Multiple files can be specified as the source files. For example, to transfer the contents of the directory **downloads/** to an existing directory called **uploads/** on the remote machine penguin.example.net, type the following at a shell prompt:

```
scp downloads/* username@penguin.example.net:uploads/
```

8.6.3. Using the sftp Command

The **sftp** utility can be used to open a secure, interactive FTP session. It is similar to **ftp** except that it uses a secure, encrypted connection. The general syntax is **sftp *username@hostname.com***. Once authenticated, you can use a set of commands similar to those used by FTP. Refer to the **sftp** man page for a list of these commands. To read the man page, execute the command **man sftp** at a shell prompt. The **sftp** utility is only available in OpenSSH version 2.5.0p1 and higher.

8.7. More Than a Secure Shell

A secure command line interface is just the beginning of the many ways SSH can be used. Given the proper amount of bandwidth, X11 sessions can be directed over an SSH channel. Or, by using TCP/IP forwarding, previously insecure port connections between systems can be mapped to specific SSH channels.

8.7.1. X11 Forwarding

Opening an X11 session over an SSH connection is as easy as connecting to the SSH server using the -Y option and running an X program on a local machine.

```
ssh -Y <user>@example.com
```

When an X program is run from the secure shell prompt, the SSH client and server create a new secure channel, and the X program data is sent over that channel to the client machine transparently.

X11 forwarding can be very useful. For example, X11 forwarding can be used to create a secure, interactive session of the **Printer Configuration Tool**. To do this, connect to the server using **ssh** and type:

```
system-config-printer &
```

After supplying the root password for the server, the **Printer Configuration Tool** appears and allows the remote user to safely configure printing on the remote system.

8.7.2. Port Forwarding

SSH can secure otherwise insecure TCP/IP protocols via port forwarding. When using this technique, the SSH server becomes an encrypted conduit to the SSH client.

Port forwarding works by mapping a local port on the client to a remote port on the server. SSH can map any port from the server to any port on the client; port numbers do not need to match for this technique to work.

To create a TCP/IP port forwarding channel which listens for connections on the localhost, use the following command:

```
ssh -L local-port:remote-hostname:remote-port
        username@hostname
```

Note

Setting up port forwarding to listen on ports below 1024 requires root level access.

To check email on a server called **mail.example.com** using POP3 through an encrypted connection, use the following command:

```
ssh -L 1100:mail.example.com:110 mail.example.com
```

Once the port forwarding channel is in place between the client machine and the mail server, direct a POP3 mail client to use port 1100 on the localhost to check for new mail. Any requests sent to port 1100 on the client system are directed securely to the **mail.example.com** server.

If **mail.example.com** is not running an SSH server, but another machine on the same network is, SSH can still be used to secure part of the connection. However, a slightly different command is necessary:

```
ssh -L 1100:mail.example.com:110 other.example.com
```

In this example, POP3 requests from port 1100 on the client machine are forwarded through the SSH connection on port 22 to the SSH server, **other.example.com**. Then, **other.example.com** connects to port 110 on **mail.example.com** to check for new mail. Note, when using this technique only the connection between the client system and **other.example.com** SSH server is secure.

Port forwarding can also be used to get information securely through network firewalls. If the firewall is configured to allow SSH traffic via its standard port (22) but blocks access to other ports, a connection between two hosts using the blocked ports is still possible by redirecting their communication over an established SSH connection.

Note

Using port forwarding to forward connections in this manner allows any user on the client system to connect to that service. If the client system becomes compromised, the attacker also has access to forwarded services.

System administrators concerned about port forwarding can disable this functionality on the server by specifying a **No** parameter for the **AllowTcpForwarding** line in **/etc/ssh/sshd_config** and restarting the **sshd** service.

8.7.3. Generating Key Pairs

If you do not want to enter your password every time you use **ssh**, **scp**, or **sftp** to connect to a remote machine, you can generate an authorization key pair.

Keys must be generated for each user. To generate keys for a user, use the following steps as the user who wants to connect to remote machines. If you complete the steps as root, only root will be able to use the keys.

Starting with OpenSSH version 3.0, **~/.ssh/authorized_keys2**, **~/.ssh/known_hosts2**, and **/etc/ssh_known_hosts2** are obsolete. SSH Protocol 1 and 2 share the **~/.ssh/authorized_keys**, **~/.ssh/known_hosts**, and **/etc/ssh/ssh_known_hosts** files.

Fedora 12 uses SSH Protocol 2 and RSA keys by default.

Tip

If you reinstall and want to save your generated key pair, backup the **.ssh** directory in your home directory. After reinstalling, copy this directory back to your home directory. This process can be done for all users on your system, including root.

8.7.3.1. Generating an RSA Key Pair for Version 2

Use the following steps to generate an RSA key pair for version 2 of the SSH protocol. This is the default starting with OpenSSH 2.9.

1. To generate an RSA key pair to work with version 2 of the protocol, type the following command at a shell prompt:

   ```
   ssh-keygen -t rsa
   ```

Accept the default file location of **~/.ssh/id_rsa**. Enter a passphrase different from your account password and confirm it by entering it again.

The public key is written to **~/.ssh/id_rsa.pub**. The private key is written to **~/.ssh/id_rsa**. Never distribute your private key to anyone.

2. Change the permissions of the **.ssh** directory using the following command:

   ```
   chmod 755 ~/.ssh
   ```

3. Copy the contents of **~/.ssh/id_rsa.pub** into the file **~/.ssh/authorized_keys** on the machine to which you want to connect. If the file **~/.ssh/authorized_keys** exist, append the contents of the file **~/.ssh/id_rsa.pub** to the file **~/.ssh/authorized_keys** on the other machine.

4. Change the permissions of the **authorized_keys** file using the following command:

   ```
   chmod 644 ~/.ssh/authorized_keys
   ```

5. If you are running GNOME or are running in a graphical desktop with GTK2+ libraries installed, skip to *Section 8.7.3.4, “Configuring **ssh-agent** with a GUI”*. If you are not running the X Window System, skip to *Section 8.7.3.5, “Configuring **ssh-agent**”*.

8.7.3.2. Generating a DSA Key Pair for Version 2

Use the following steps to generate a DSA key pair for version 2 of the SSH Protocol.

1. To generate a DSA key pair to work with version 2 of the protocol, type the following command at a shell prompt:

   ```
   ssh-keygen -t dsa
   ```

   Accept the default file location of **~/.ssh/id_dsa**. Enter a passphrase different from your account password and confirm it by entering it again.

   > **Tip**
   >
   > A passphrase is a string of words and characters used to authenticate a user. Passphrases differ from passwords in that you can use spaces or tabs in the passphrase. Passphrases are generally longer than passwords because they are usually phrases instead of a single word.

   The public key is written to **~/.ssh/id_dsa.pub**. The private key is written to **~/.ssh/id_dsa**. It is important never to give anyone the private key.

2. Change the permissions of the **.ssh** directory with the following command:

```
chmod 755 ~/.ssh
```

3. Copy the contents of **~/.ssh/id_dsa.pub** into the file **~/.ssh/authorized_keys** on the machine to which you want to connect. If the file **~/.ssh/authorized_keys** exist, append the contents of the file **~/.ssh/id_dsa.pub** to the file **~/.ssh/authorized_keys** on the other machine.

4. Change the permissions of the **authorized_keys** file using the following command:

   ```
   chmod 644 ~/.ssh/authorized_keys
   ```

5. If you are running GNOME or a graphical desktop environment with the GTK2+ libraries installed, skip to *Section 8.7.3.4, “Configuring **ssh-agent** with a GUI”*. If you are not running the X Window System, skip to *Section 8.7.3.5, “Configuring **ssh-agent**”*.

8.7.3.3. Generating an RSA Key Pair for Version 1.3 and 1.5

Use the following steps to generate an RSA key pair, which is used by version 1 of the SSH Protocol. If you are only connecting between systems that use DSA, you do not need an RSA version 1.3 or RSA version 1.5 key pair.

1. To generate an RSA (for version 1.3 and 1.5 protocol) key pair, type the following command at a shell prompt:

   ```
   ssh-keygen -t rsa1
   ```

   Accept the default file location (**~/.ssh/identity**). Enter a passphrase different from your account password. Confirm the passphrase by entering it again.

   The public key is written to **~/.ssh/identity.pub**. The private key is written to **~/.ssh/identity**. Do not give anyone the private key.

2. Change the permissions of your **.ssh** directory and your key with the commands **chmod 755 ~/.ssh** and **chmod 644 ~/.ssh/identity.pub**.

3. Copy the contents of **~/.ssh/identity.pub** into the file **~/.ssh/authorized_keys** on the machine to which you wish to connect. If the file **~/.ssh/authorized_keys** does not exist, you can copy the file **~/.ssh/identity.pub** to the file **~/.ssh/authorized_keys** on the remote machine.

4. If you are running GNOME, skip to *Section 8.7.3.4, “Configuring **ssh-agent** with a GUI”*. If you are not running GNOME, skip to *Section 8.7.3.5, “Configuring **ssh-agent**”*.

8.7.3.4. Configuring ssh-agent with a GUI

The **ssh-agent** utility can be used to save your passphrase so that you do not have to enter it each time you initiate an **ssh** or **scp** connection. If you are using GNOME, the **gnome-ssh-askpass** package contains the application used to prompt you for your passphrase when you log in to GNOME and save it until you log out of GNOME. You will not have to enter your password or passphrase for any **`ssh`** or **`scp`** connection made during that GNOME session. If you are not using GNOME, refer to *Section 8.7.3.5, “Configuring **`ssh-agent`**”*.

To save your passphrase during your GNOME session, follow the following steps:

1. You will need to have the package **`gnome-ssh-askpass`** installed; you can use the command **`rpm -q openssh-askpass`** to determine if it is installed or not. If it is not installed, install it from your Fedora CD-ROM set, from a Red Hat FTP mirror site, or using Red Hat Network.

2. Select **Main Menu Button** (on the Panel) > **Preferences** > **More Preferences** > **Sessions**, and click on the **Startup Programs** tab. Click **Add** and enter **`/usr/bin/ssh-add`** in the **Startup Command** text area. Set it a priority to a number higher than any existing commands to ensure that it is executed last. A good priority number for **`ssh-add`** is 70 or higher. The higher the priority number, the lower the priority. If you have other programs listed, this one should have the lowest priority. Click **Close** to exit the program.

3. Log out and then log back into GNOME; in other words, restart X. After GNOME is started, a dialog box will appear prompting you for your passphrase(s). Enter the passphrase requested. If you have both DSA and RSA key pairs configured, you will be prompted for both. From this point on, you should not be prompted for a password by **`ssh`**, **`scp`**, or **`sftp`**.

8.7.3.5. Configuring ssh-agent

The **`ssh-agent`** can be used to store your passphrase so that you do not have to enter it each time you make a **`ssh`** or **`scp`** connection. If you are not running the X Window System, follow these steps from a shell prompt. If you are running GNOME but you do not want to configure it to prompt you for your passphrase when you log in (refer to *Section 8.7.3.4, “Configuring **`ssh-agent`** with a GUI”*), this procedure will work in a terminal window, such as an XTerm. If you are running X but not GNOME, this procedure will work in a terminal window. However, your passphrase will only be remembered for that terminal window; it is not a global setting.

1. At a shell prompt, type the following command:

   ```
   exec /usr/bin/ssh-agent $SHELL
   ```

2. Then type the command:

   ```
   ssh-add
   ```

   and enter your passphrase(s). If you have more than one key pair configured, you will be prompted for each one.

3. When you log out, your passphrase(s) will be forgotten. You must execute these two commands each time you log in to a virtual console or open a terminal window.

8.8. Additional Resources

The OpenSSH and OpenSSL projects are in constant development, and the most up-to-date information for them is available from their websites. The man pages for OpenSSH and OpenSSL tools are also good sources of detailed information.

8.8.1. Installed Documentation

- The **`ssh`**, **`scp`**, **`sftp`**, **`sshd`**, and **`ssh-keygen`** man pages — These man pages include information on how to use these commands as well as all the parameters that can be used with them.

8.8.2. Useful Websites

- *http://www.openssh.com/* — The OpenSSH FAQ page, bug reports, mailing lists, project goals, and a more technical explanation of the security features.

- *http://www.openssl.org/* — The OpenSSL FAQ page, mailing lists, and a description of the project goal.

- *http://www.freessh.org/* — SSH client software for other platforms.

Samba

Samba is an open source implementation of the Server Message Block (SMB) protocol. It allows the networking of Microsoft Windows®, Linux, UNIX, and other operating systems together, enabling access to Windows-based file and printer shares. Samba's use of SMB allows it to appear as a Windows server to Windows clients.

9.1. Introduction to Samba

The third major release of Samba, version 3.0.0, introduced numerous improvements from prior versions, including:

- The ability to join an Active Directory domain by means of LDAP and Kerberos
- Built in Unicode support for internationalization
- Support for Microsoft Windows XP Professional client connections to Samba servers without needing local registry hacking
- Two new documents developed by the Samba.org team, which include a 400+ page reference manual, and a 300+ page implementation and integration manual. For more information about these published titles, refer to *Section 9.12.2, “Related Books”*.

9.1.1. Samba Features

Samba is a powerful and versatile server application. Even seasoned system administrators must know its abilities and limitations before attempting installation and configuration.

What Samba can do:

- Serve directory trees and printers to Linux, UNIX, and Windows clients
- Assist in network browsing (with or without NetBIOS)
- Authenticate Windows domain logins
- Provide Windows Internet Name Service (WINS) name server resolution
- Act as a Windows NT®-style Primary Domain Controller (PDC)
- Act as a Backup Domain Controller (BDC) for a Samba-based PDC
- Act as an Active Directory domain member server
- Join a Windows NT/2000/2003 PDC

What Samba cannot do:

- Act as a BDC for a Windows PDC (and vice versa)
- Act as an Active Directory domain controller

9.2. Samba Daemons and Related Services

The following is a brief introduction to the individual Samba daemons and services.

9.2.1. Samba Daemons

Samba is comprised of three daemons (**`smbd`**, **`nmbd`**, and **`winbindd`**). Two services (**`smb`** and **`windbind`**) control how the daemons are started, stopped, and other service-related features. Each daemon is listed in detail, as well as which specific service has control over it.

`smbd`

The **`smbd`** server daemon provides file sharing and printing services to Windows clients. In addition, it is responsible for user authentication, resource locking, and data sharing through the SMB protocol. The default ports on which the server listens for SMB traffic are TCP ports 139 and 445.

The **`smbd`** daemon is controlled by the **`smb`** service.

`nmbd`

The **`nmbd`** server daemon understands and replies to NetBIOS name service requests such as those produced by SMB/CIFS in Windows-based systems. These systems include Windows 95/98/ ME, Windows NT, Windows 2000, Windows XP, and LanManager clients. It also participates in the browsing protocols that make up the Windows **Network Neighborhood** view. The default port that the server listens to for NMB traffic is UDP port 137.

The **`nmbd`** daemon is controlled by the **`smb`** service.

`winbindd`

The **`winbind`** service resolves user and group information on a server running Windows NT 2000 or Windows Server 2003. This makes Windows user / group information understandable by UNIX platforms. This is achieved by using Microsoft RPC calls, Pluggable Authentication Modules (PAM), and the Name Service Switch (NSS). This allows Windows NT domain users to appear and operate as UNIX users on a UNIX machine. Though bundled with the Samba distribution, the **`winbind`** service is controlled separately from the **`smb`** service.

The **`winbindd`** daemon is controlled by the **`winbind`** service and does not require the **`smb`** service to be started in order to operate. Winbindd is also used when Samba is an Active Directory member, and may also be used on a Samba domain controller (to implement nested groups and/or interdomain trust). Because **`winbind`** is a client-side service used to connect to Windows NT-based servers, further discussion of **`winbind`** is beyond the scope of this manual.

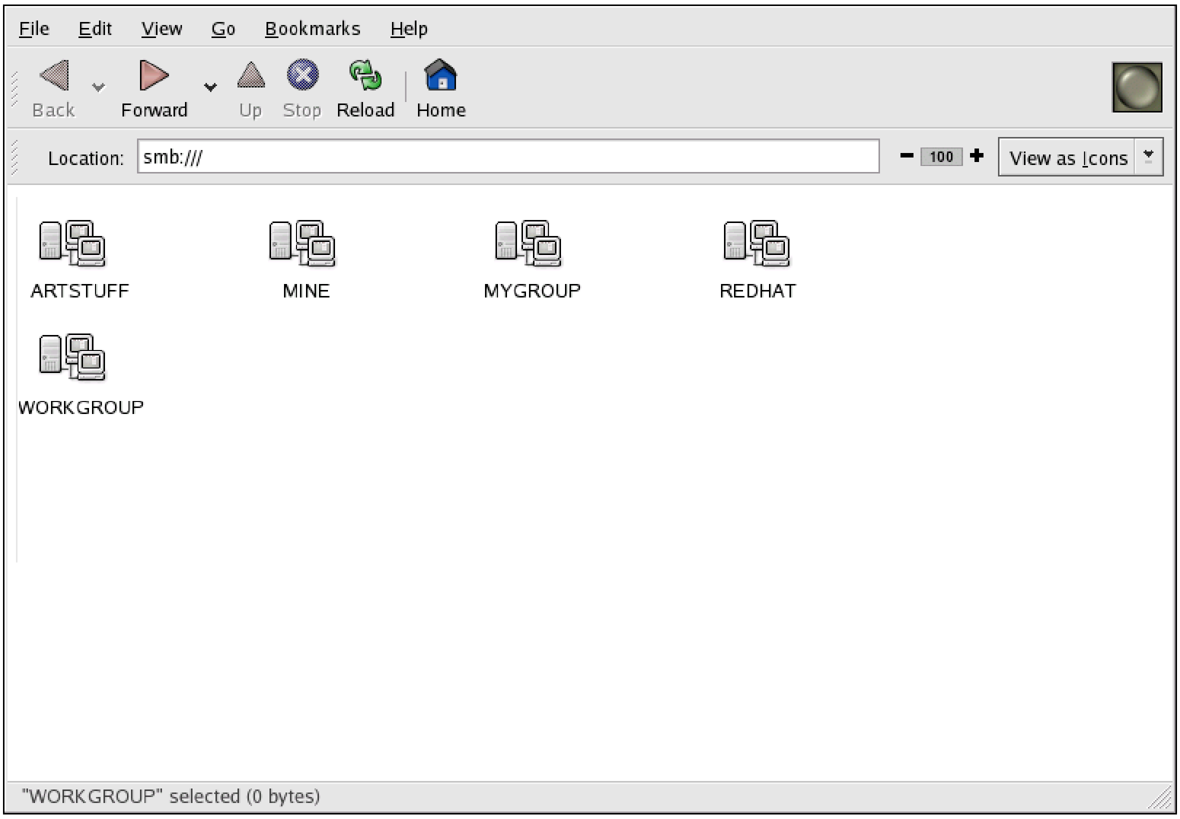
Note

You may refer to *Section 9.11, “Samba Distribution Programs”* for a list of utilities included in the Samba distribution.

9.3. Connecting to a Samba Share

You can use **Nautilus** to view available Samba shares on your network. Select **Places** (on the Panel) > **Network Servers** to view a list of Samba workgroups on your network. You can also type **`smb:`** in the **File** > **Open Location** bar of Nautilus to view the workgroups.

As shown in *Figure 9.1, “SMB Workgroups in Nautilus”*, an icon appears for each available SMB workgroup on the network.

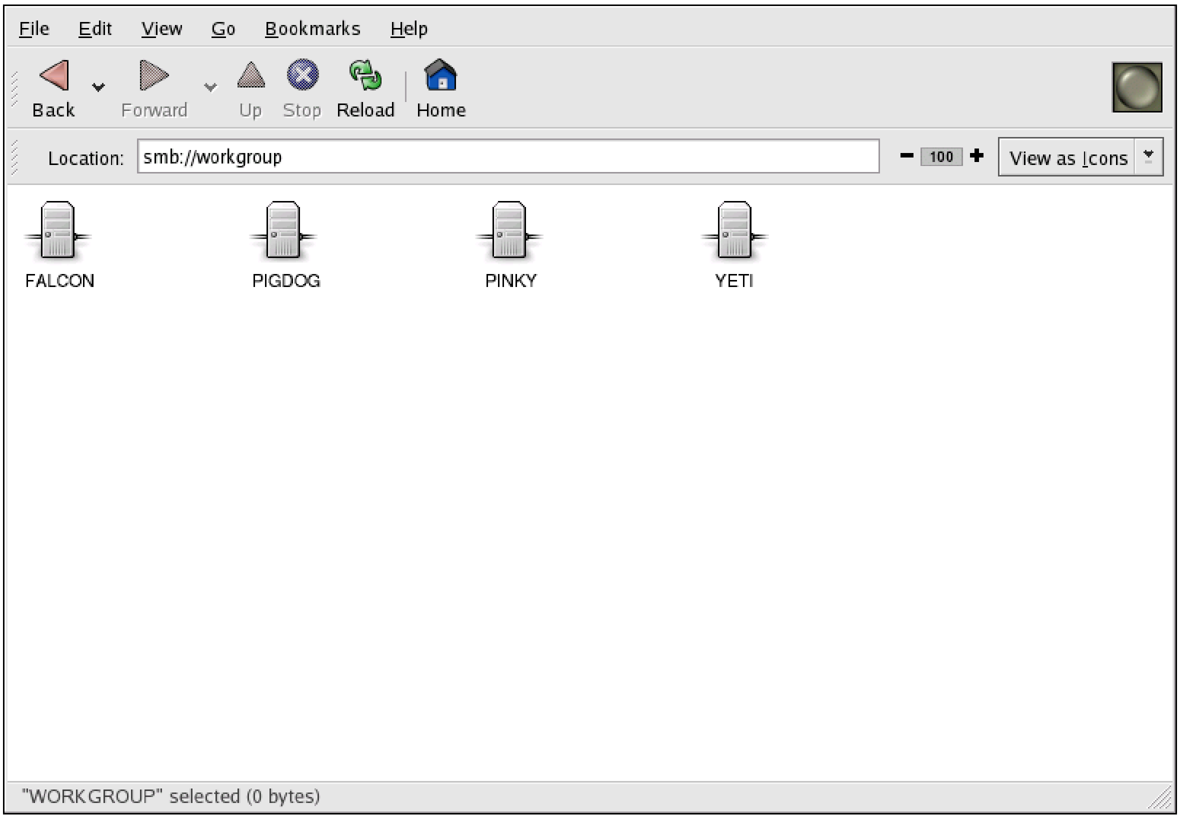


Location: smb:///

"WORKGROUP" selected (0 bytes)

Figure 9.1. SMB Workgroups in Nautilus

Double-click one of the workgroup icons to view a list of computers within the workgroup.



Location: smb://workgroup

"WORKGROUP" selected (0 bytes)

Figure 9.2. SMB Machines in Nautilus

As you can see from *Figure 9.2, “SMB Machines in Nautilus”*, there is an icon for each machine within the workgroup. Double-click on an icon to view the Samba shares on the machine. If a username and password combination is required, you are prompted for them.

Alternately, you can also specify the Samba server and sharename in the **Location:** bar for **Nautilus** using the following syntax (replace <*servername*> and <*sharename*> with the appropriate values):

```
smb://<servername>/<sharename>
```

9.3.1. Command Line

To query the network for Samba servers, use the **findsmb** command. For each server found, it displays its IP address, NetBIOS name, workgroup name, operating system, and SMB server version.

To connect to a Samba share from a shell prompt, type the following command:

```
smbclient //<hostname>/<sharename> -U <username>
```

Replace <*hostname*> with the hostname or IP address of the Samba server you want to connect to, <*sharename*> with the name of the shared directory you want to browse, and <*username*> with the Samba username for the system. Enter the correct password or press **Enter** if no password is required for the user.

If you see the `smb:\>` prompt, you have successfully logged in. Once you are logged in, type **help** for a list of commands. If you wish to browse the contents of your home directory, replace *sharename* with your username. If the **-U** switch is not used, the username of the current user is passed to the Samba server.

To exit **smbclient**, type **exit** at the `smb:\>` prompt.

9.3.2. Mounting the Share

Sometimes it is useful to mount a Samba share to a directory so that the files in the directory can be treated as if they are part of the local file system.

To mount a Samba share to a directory, create create a directory to mount it to (if it does not already exist), and execute the following command as root:

```
mount -t cifs -o <username>,<password> //<servername>/<sharename>/mnt/
point/
```

This command mounts <*sharename*> from <*servername*> in the local directory */mnt/point/*. For more information about mounting a samba share, refer to **man mount.cifs**.

9.4. Configuring a Samba Server

The default configuration file (**/etc/samba/smb.conf**) allows users to view their home directories as a Samba share. It also shares all printers configured for the system as Samba shared printers. In other words, you can attach a printer to the system and print to it from the Windows machines on your network.

9.4.1. Graphical Configuration

To configure Samba using a graphical interface, use the **Samba Server Configuration Tool**. For command line configuration, skip to *Section 9.4.2, “Command Line Configuration”*.

The **Samba Server Configuration Tool** is a graphical interface for managing Samba shares, users, and basic server settings. It modifies the configuration files in the **/etc/samba/** directory. Any changes to these files not made using the application are preserved.

To use this application, you must be running the X Window System, have root privileges, and have the **system-config-samba** RPM package installed. To start the **Samba Server Configuration Tool** from the desktop, go to the **System** (on the Panel) > **Administration** > **Server Settings** > **Samba** or type the command **system-config-samba** at a shell prompt (for example, in an XTerm or a GNOME terminal).

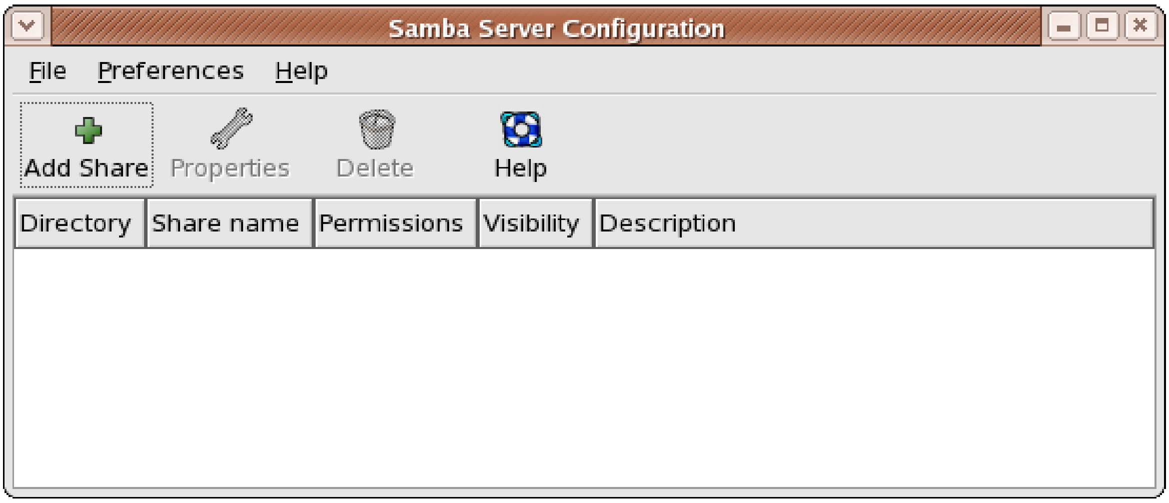


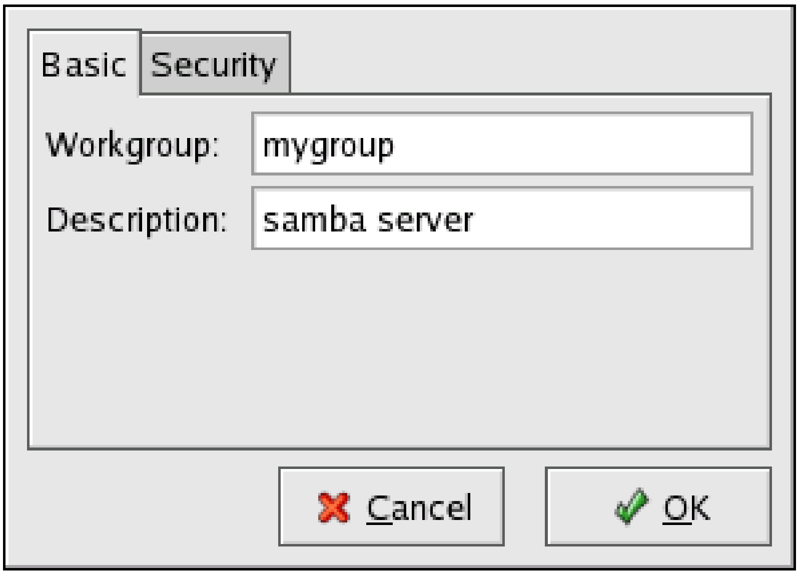
Figure 9.3. **Samba Server Configuration Tool**

Note

The **Samba Server Configuration Tool** does not display shared printers or the default stanza that allows users to view their own home directories on the Samba server.

9.4.1.1. Configuring Server Settings

The first step in configuring a Samba server is to configure the basic settings for the server and a few security options. After starting the application, select **Preferences** > **Server Settings** from the pulldown menu. The **Basic** tab is displayed as shown in *Figure 9.4, "Configuring Basic Server Settings"*.

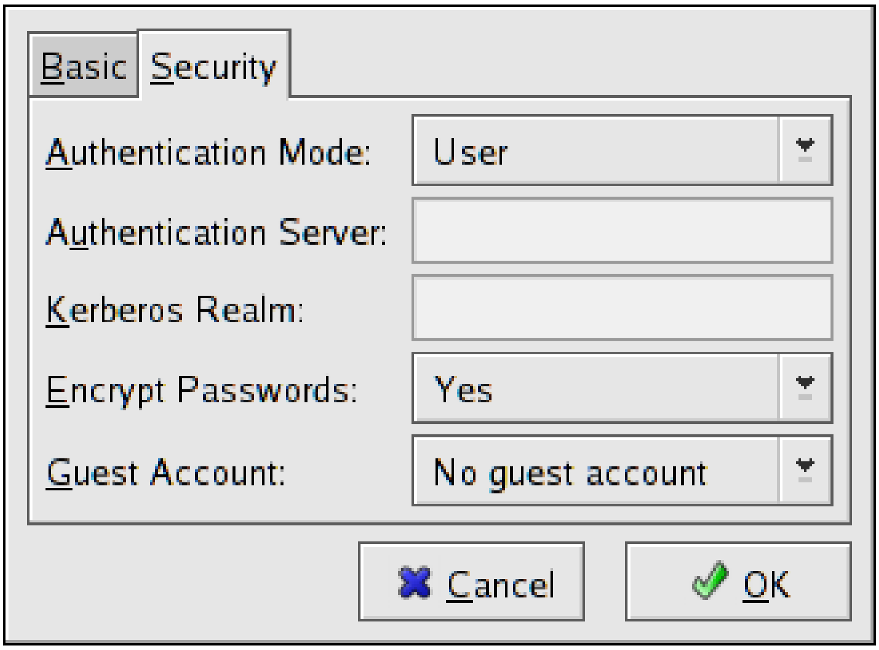


Workgroup: mygroup

Description: samba server

Figure 9.4. Configuring Basic Server Settings

On the **Basic** tab, specify which workgroup the computer should be in as well as a brief description of the computer. They correspond to the `workgroup` and `server string` options in `smb.conf`.



Authentication Mode: User

Authentication Server:

Kerberos Realm:

Encrypt Passwords: Yes

Guest Account: No guest account

Figure 9.5. Configuring Security Server Settings

The **Security** tab contains the following options:

- **Authentication Mode** — This corresponds to the **`security`** option. Select one of the following types of authentication.
  - **ADS** — The Samba server acts as a domain member in an Active Directory Domain (ADS) realm. For this option, Kerberos must be installed and configured on the server, and Samba must become a member of the ADS realm using the **`net`** utility, which is part of the **`samba-client`** package. Refer to the **`net`** man page for details. This option does not configure Samba to be an ADS Controller. Specify the realm of the Kerberos server in the **Kerberos Realm** field.

    > **Note**
    >
    > The **Kerberos Realm** field must be supplied in all uppercase letters, such as **`EXAMPLE.COM`**.
    >
    > Using a Samba server as a domain member in an ADS realm assumes proper configuration of Kerberos, including the **`/etc/krb5.conf`** file.

  - **Domain** — The Samba server relies on a Windows NT Primary or Backup Domain Controller to verify the user. The server passes the username and password to the Controller and waits for it to return. Specify the NetBIOS name of the Primary or Backup Domain Controller in the **Authentication Server** field.

    The **Encrypted Passwords** option must be set to **Yes** if this is selected.

  - **Server** — The Samba server tries to verify the username and password combination by passing them to another Samba server. If it can not, the server tries to verify using the user authentication mode. Specify the NetBIOS name of the other Samba server in the **Authentication Server** field.

  - **Share** — Samba users do not have to enter a username and password combination on a per Samba server basis. They are not prompted for a username and password until they try to connect to a specific shared directory from a Samba server.

  - **User** — (Default) Samba users must provide a valid username and password on a per Samba server basis. Select this option if you want the **Windows Username** option to work. Refer to *Section 9.4.1.2, "Managing Samba Users"* for details.

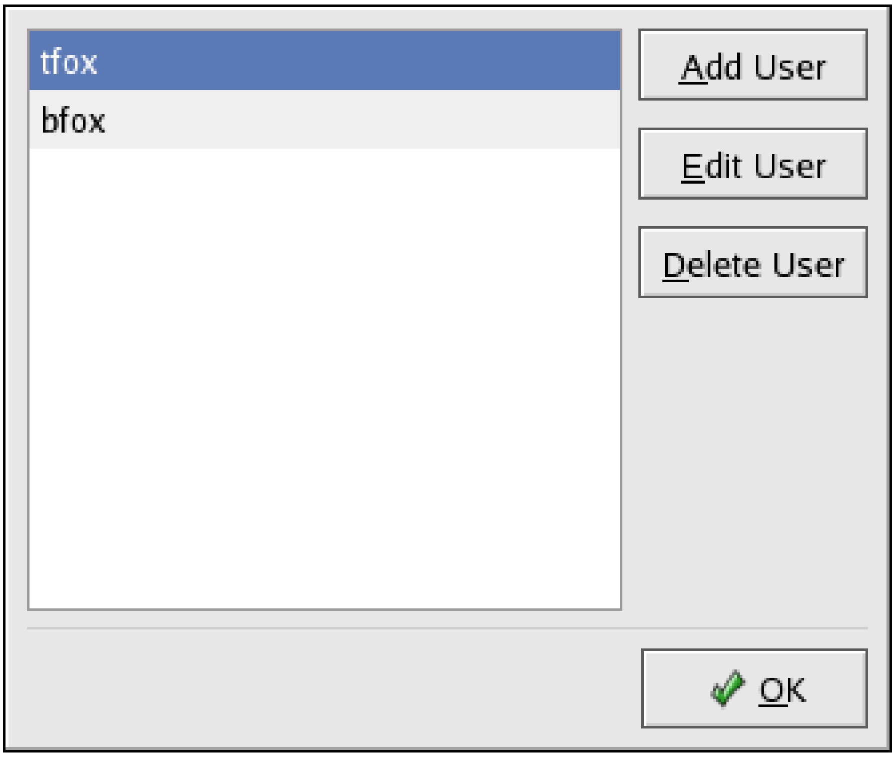
- **Encrypt Passwords** — This option must be enabled if the clients are connecting from a system with Windows 98, Windows NT 4.0 with Service Pack 3, or other more recent versions of Microsoft Windows. The passwords are transfered between the server and the client in an encrypted format instead of as a plain-text word that can be intercepted. This corresponds to the **`encrypted passwords`** option. Refer to *Section 9.4.3, "Encrypted Passwords"* for more information about encrypted Samba passwords.

- **Guest Account** — When users or guest users log into a Samba server, they must be mapped to a valid user on the server. Select one of the existing usernames on the system to be the guest Samba account. When guests log in to the Samba server, they have the same privileges as this user. This corresponds to the **`guest account`** option.

After clicking **OK**, the changes are written to the configuration file and the daemon is restarted; thus, the changes take effect immediately.

9.4.1.2. Managing Samba Users

The **Samba Server Configuration Tool** requires that an existing user account be active on the system acting as the Samba server before a Samba user can be added. The Samba user is associated with the existing user account.



tfox
bfox

Figure 9.6. Managing Samba Users

To add a Samba user, select **Preferences** > **Samba Users** from the pulldown menu, and click the **Add User** button. In the **Create New Samba User** window select a **Unix Username** from the list of existing users on the local system.

If the user has a different username on a Windows machine and needs to log into the Samba server from the Windows machine, specify that Windows username in the **Windows Username** field. The **Authentication Mode** on the **Security** tab of the **Server Settings** preferences must be set to **User** for this option to work.

Also, configure a **Samba Password** for the Samba User and confirm it by typing it again. Even if you opt to use encrypted passwords for Samba, it is recommended that the Samba passwords for all users are different from their system passwords.

To edit an existing user, select the user from the list, and click **Edit User**. To delete an existing Samba user, select the user, and click the **Delete User** button. Deleting a Samba user does not delete the associated system user account.

The users are modified immediately after clicking the **OK** button.

9.4.1.3. Adding a Share

To create a Samba share, click the **Add** button from the main Samba configuration window.

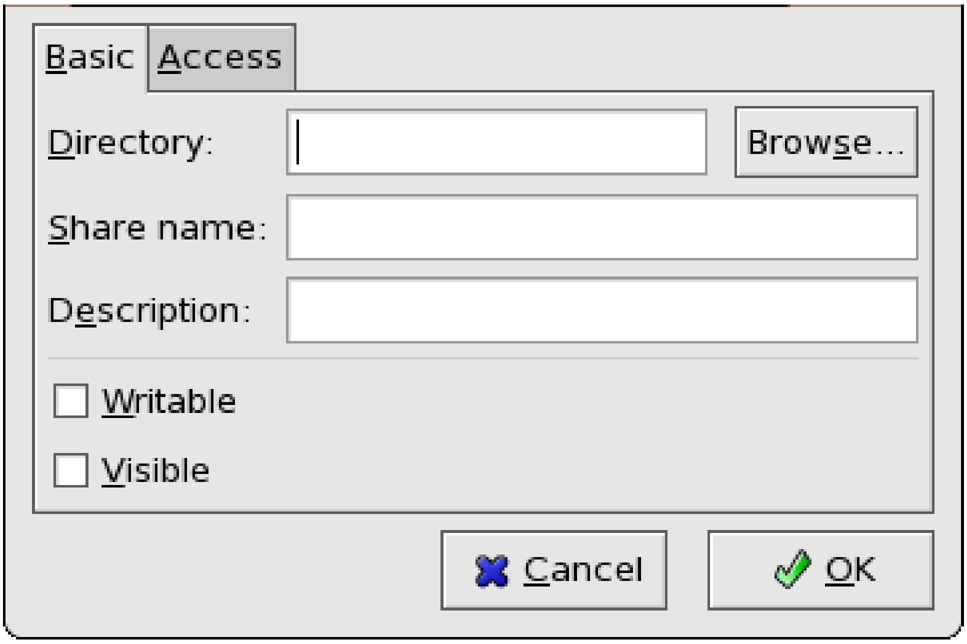


Figure 9.7. Adding a Share

The **Basic** tab configures the following options:

- **Directory** — The directory to share via Samba. The directory must exist before it can be entered here.

- **Share name** — The actual name of the share that is seen from remote machines. By default, it is the same value as **Directory**, but can be configured.

- **Descriptions** — A brief description of the share.

- **Writable** — Enables users to read and write to the shared directory
- **Visible** — Grants read-only rights to users for the shared directory.

On the **Access** tab, select whether to allow only specified users to access the share or whether to allow all Samba users to access the share. If you select to allow access to specific users, select the users from the list of available Samba users.

The share is added immediately after clicking **OK**.

9.4.2. Command Line Configuration

Samba uses **`/etc/samba/smb.conf`** as its configuration file. If you change this configuration file, the changes do not take effect until you restart the Samba daemon with the command **`service smb restart`**.

To specify the Windows workgroup and a brief description of the Samba server, edit the following lines in your **`smb.conf`** file:

```
workgroup = WORKGROUPNAME
server string = BRIEF COMMENT ABOUT SERVER
```

Replace *`WORKGROUPNAME`* with the name of the Windows workgroup to which this machine should belong. The *`BRIEF COMMENT ABOUT SERVER`* is optional and is used as the Windows comment about the Samba system.

To create a Samba share directory on your Linux system, add the following section to your **`smb.conf`** file (after modifying it to reflect your needs and your system):

```
[sharename]
comment = Insert a comment here
path = /home/share/
valid users = tfox carole
public = no
writable = yes
printable = no
create mask = 0765
```

The above example allows the users **`tfox`** and **`carole`** to read and write to the directory **`/home/share`**, on the Samba server, from a Samba client.

9.4.3. Encrypted Passwords

Encrypted passwords are enabled by default because it is more secure to do so. To create a user with an encrypted password, use the command **`smbpasswd -a <username>`**.

9.5. Starting and Stopping Samba

To start a Samba server, type the following command in a shell prompt while logged in as root:

```
/sbin/service smb start
```

Important

To set up a domain member server, you must first join the domain or Active Directory using the **`net join`** command *before* starting the **`smb`** service.

To stop the server, type the following command in a shell prompt while logged in as root:

```
/sbin/service smb stop
```

The `restart` option is a quick way of stopping and then starting Samba. This is the most reliable way to make configuration changes take effect after editing the configuration file for Samba. Note that the restart option starts the daemon even if it was not running originally.

To restart the server, type the following command in a shell prompt while logged in as root:

```
 /sbin/service smb restart
```

The `condrestart` (*conditional restart*) option only starts **`smb`** on the condition that it is currently running. This option is useful for scripts, because it does not start the daemon if it is not running.

Note

When the **`smb.conf`** file is changed, Samba automatically reloads it after a few minutes. Issuing a manual **`restart`** or **`reload`** is just as effective.

To conditionally restart the server, type the following command as root:

```
 /sbin/service smb condrestart
```

A manual reload of the **`smb.conf`** file can be useful in case of a failed automatic reload by the **`smb`** service. To ensure that the Samba server configuration file is reloaded without restarting the service, type the following command as root:

```
 /sbin/service smb reload
```

By default, the **`smb`** service does *not* start automatically at boot time. To configure Samba to start at boot time, use an initscript utility, such as **`/sbin/chkconfig`**, **`/usr/sbin/ntsysv`**, or the **Services Configuration Tool** program. Refer to *Chapter 6, Controlling Access to Services* for more information regarding these tools.

9.6. Samba Server Types and the `smb.conf` File

Samba configuration is straightforward. All modifications to Samba are done in the **`/etc/samba/smb.conf`** configuration file. Although the default **`smb.conf`** file is well documented, it does not address complex topics such as LDAP, Active Directory, and the numerous domain controller implementations.

The following sections describe the different ways a Samba server can be configured. Keep in mind your needs and the changes required to the **`smb.conf`** file for a successful configuration.

9.6.1. Stand-alone Server

A stand-alone server can be a workgroup server or a member of a workgroup environment. A stand-alone server is not a domain controller and does not participate in a domain in any way. The following examples include several anonymous share-level security configurations and one user-level security configuration. For more information on share-level and user-level security modes, refer to *Section 9.7, “Samba Security Modes”*.

9.6.1.1. Anonymous Read-Only

The following **`smb.conf`** file shows a sample configuration needed to implement anonymous read-only file sharing. The **`security = share`** parameter makes a share anonymous. Note, security levels for a single Samba server cannot be mixed. The **`security`** directive is a global Samba parameter located in the **`[global]`** configuration section of the **`smb.conf`** file.

```
[global]
workgroup = DOCS
netbios name = DOCS_SRV
security = share
[data]
comment = Documentation Samba Server
path = /export
read only = Yes
guest only = Yes
```

9.6.1.2. Anonymous Read/Write

The following **`smb.conf`** file shows a sample configuration needed to implement anonymous read/write file sharing. To enable anonymous read/write file sharing, set the **`read only`** directive to **`no`**. The **`force user`** and **`force group`** directives are also added to enforce the ownership of any newly placed files specified in the share.

> **Note**
>
> Although having an anonymous read/write server is possible, it is not recommended. Any files placed in the share space, regardless of user, are assigned the user/group combination as specified by a generic user (**`force user`**) and group (**`force group`**) in the **`smb.conf`** file.

```
[global]
workgroup = DOCS
netbios name = DOCS_SRV
security = share
[data]
comment = Data
path = /export
force user = docsbot
force group = users
read only = No
guest ok = Yes
```

9.6.1.3. Anonymous Print Server

The following **smb.conf** file shows a sample configuration needed to implement an anonymous print server. Setting **browseable** to **no** as shown does not list the printer in Windows **Network Neighborhood**. Although hidden from browsing, configuring the printer explicitly is possible. By connecting to **DOCS_SRV** using NetBIOS, the client can have access to the printer if the client is also part of the **DOCS** workgroup. It is also assumed that the client has the correct local printer driver installed, as the **use client driver** directive is set to **Yes**. In this case, the Samba server has no responsibility for sharing printer drivers to the client.

```

[global]
workgroup = DOCS
netbios name = DOCS_SRV
security = share
printcap name = cups
disable spools= Yes
show add printer wizard = No
printing = cups
[printers]
comment = All Printers
path = /var/spool/samba
guest ok = Yes
printable = Yes
use client driver = Yes
browseable = Yes
```

9.6.1.4. Secure Read/Write File and Print Server

The following **smb.conf** file shows a sample configuration needed to implement a secure read/write print server. Setting the **security** directive to **user** forces Samba to authenticate client connections. Notice the **[homes]** share does not have a **force user** or **force group** directive as the **[public]** share does. The **[homes]** share uses the authenticated user details for any files created as opposed to the **force user** and **force group** in **[public]**.

```

[global]
workgroup = DOCS
```

```
netbios name = DOCS_SRV
security = user
printcap name = cups
disable spools = Yes
show add printer wizard = No
printing = cups
[homes]
comment = Home Directories
valid users = %S
read only = No
browseable = No
[public]
comment = Data
path = /export
force user = docsbot
force group = users
guest ok = Yes
[printers]
comment = All Printers
path = /var/spool/samba
printer admin = john, ed, @admins
create mask = 0600
guest ok = Yes
printable = Yes
use client driver = Yes
browseable = Yes
```

9.6.2. Domain Member Server

A domain member, while similar to a stand-alone server, is logged into a domain controller (either Windows or Samba) and is subject to the domain's security rules. An example of a domain member server would be a departmental server running Samba that has a machine account on the Primary Domain Controller (PDC). All of the department's clients still authenticate with the PDC, and desktop profiles and all network policy files are included. The difference is that the departmental server has the ability to control printer and network shares.

9.6.2.1. Active Directory Domain Member Server

The following **`smb.conf`** file shows a sample configuration needed to implement an Active Directory domain member server. In this example, Samba authenticates users for services being run locally but is also a client of the Active Directory. Ensure that your kerberos **`realm`** parameter is shown in all caps (for example **`realm = EXAMPLE.COM`**). Since Windows 2000/2003 requires Kerberos for Active Directory authentication, the **`realm`** directive is required. If Active Directory and Kerberos are running on different servers, the **`password server`** directive may be required to help the distinction.

```

[global]
realm = EXAMPLE.COM
security = ADS
encrypt passwords = yes
```

```
# Optional. Use only if Samba cannot determine the Kerberos server
 automatically.
password server = kerberos.example.com
```

In order to join a member server to an Active Directory domain, the following steps must be completed:

- Configuration of the **smb.conf** file on the member server
- Configuration of Kerberos, including the **/etc/krb5.conf** file, on the member server
- Creation of the machine account on the Active Directory domain server
- Association of the member server to the Active Directory domain

To create the machine account and join the Windows 2000/2003 Active Directory, Kerberos must first be initialized for the member server wishing to join the Active Directory domain. To create an administrative Kerberos ticket, type the following command as root on the member server:

```
kinit administrator@EXAMPLE.COM
```

The **kinit** command is a Kerberos initialization script that references the Active Directory administrator account and Kerberos realm. Since Active Directory requires Kerberos tickets, **kinit** obtains and caches a Kerberos ticket-granting ticket for client/server authentication. For more information on Kerberos, the **/etc/krb5.conf** file, and the **kinit** command, refer to .

To join an Active Directory server (windows1.example.com), type the following command as root on the member server:

```
net ads join -S windows1.example.com -U administrator%password
```

Since the machine **windows1** was automatically found in the corresponding Kerberos realm (the **kinit** command succeeded), the **net** command connects to the Active Directory server using its required administrator account and password. This creates the appropriate machine account on the Active Directory and grants permissions to the Samba domain member server to join the domain.

Note

Since **security = ads** and not **security = user** is used, a local password backend such as **smbpasswd** is not needed. Older clients that do not support **security = ads** are authenticated as if **security = domain** had been set. This change does not affect functionality and allows local users not previously in the domain.

9.6.2.2. Windows NT4-based Domain Member Server

The following **smb.conf** file shows a sample configuration needed to implement a Windows NT4-based domain member server. Becoming a member server of an NT4-based domain is similar to connecting to an Active Directory. The main difference is NT4-based domains do not use Kerberos in their authentication method, making the **smb.conf** file simpler. In this instance, the Samba member server functions as a pass through to the NT4-based domain server.

```
[global]
workgroup = DOCS
netbios name = DOCS_SRV
security = domain
[homes]
comment = Home Directories
valid users = %S
read only = No
browseable = No
[public]
comment = Data
path = /export
force user = docsbot
force group = users
guest ok = Yes
```

Having Samba as a domain member server can be useful in many situations. There are times where the Samba server can have other uses besides file and printer sharing. It may be beneficial to make Samba a domain member server in instances where Linux-only applications are required for use in the domain environment. Administrators appreciate keeping track of all machines in the domain, even if not Windows-based. In the event the Windows-based server hardware is deprecated, it is quite easy to modify the **smb.conf** file to convert the server to a Samba-based PDC. If Windows NT-based servers are upgraded to Windows 2000/2003, the **smb.conf** file is easily modifiable to incorporate the infrastructure change to Active Directory if needed.

Important

After configuring the **smb.conf** file, join the domain *before* starting Samba by typing the following command as root:

```
net rpc join -U administrator%password
```

Note that the -S option, which specifies the domain server hostname, does not need to be stated in the **net rpc join** command. Samba uses the hostname specified by the **workgroup** directive in the **smb.conf** file instead of it being stated explicitly.

9.6.3. Domain Controller

A domain controller in Windows NT is functionally similar to a Network Information Service (NIS) server in a Linux environment. Domain controllers and NIS servers both host user/group information databases as well as related services. Domain controllers are mainly used for security, including the authentication of users accessing domain resources. The service that maintains the user/group database integrity is called the *Security Account Manager* (SAM). The SAM database is stored differently between Windows and Linux Samba-based systems, therefore SAM replication cannot be achieved and platforms cannot be mixed in a PDC/BDC environment.

In a Samba environment, there can be only one PDC and zero or more BDCs.

Important

Samba cannot exist in a mixed Samba/Windows domain controller environment (Samba cannot be a BDC of a Windows PDC or vice versa). Alternatively, Samba PDCs and BDCs *can* coexist.

9.6.3.1. Primary Domain Controller (PDC) using `tdbsam`

The simplest and most common implementation of a Samba PDC uses the **`tdbsam`** password database backend. Planned to replace the aging **`smbpasswd`** backend, **`tdbsam`** has numerous improvements that are explained in more detail in *Section 9.8, “Samba Account Information Databases”*. The **`passdb backend`** directive controls which backend is to be used for the PDC.

```
[global]
workgroup = DOCS
netbios name = DOCS_SRV
passdb backend = tdbsam
security = user
add user script = /usr/sbin/useradd -m %u
delete user script = /usr/sbin/userdel -r %u
add group script = /usr/sbin/groupadd %g
delete group script = /usr/sbin/groupdel %g
add user to group script = /usr/sbin/usermod -G %g %u
add machine script = /usr/sbin/useradd -s /bin/false -d /dev/null  -g
 machines %u
# The following specifies the default logon script
# Per user logon scripts can be specified in the user
# account using pdbedit logon script = logon.bat
# This sets the default profile path.
# Set per user paths with pdbedit
logon drive = H:
domain logons = Yes
os level = 35
preferred master = Yes
domain master = Yes
[homes]
 comment = Home Directories
 valid users = %S
 read only = No
[netlogon]
 comment = Network Logon Service
 path = /var/lib/samba/netlogon/scripts
 browseable = No
 read only = No
# For profiles to work, create a user directory under the
# path shown.
mkdir -p /var/lib/samba/profiles/john
[Profiles]
 comment = Roaming Profile Share
 path = /var/lib/samba/profiles
 read only = No
 browseable = No
 guest ok = Yes
 profile acls = Yes
# Other resource shares ... ...
```

Note

If you need more than one domain controller or have more than 250 users, do *not* use a **`tdbsam`** authentication backend. LDAP is recommended in these cases.

9.6.3.2. Primary Domain Controller (PDC) with Active Directory

Although it is possible for Samba to be a member of an Active Directory, it is not possible for Samba to operate as an Active Directory domain controller.

9.7. Samba Security Modes

There are only two types of security modes for Samba, *share-level* and *user-level*, which are collectively known as *security levels*. Share-level security can only be implemented in one way, while user-level security can be implemented in one of four different ways. The different ways of implementing a security level are called *security modes*.

9.7.1. User-Level Security

User-level security is the default setting for Samba. Even if the **`security = user`** directive is not listed in the **`smb.conf`** file, it is used by Samba. If the server accepts the client's username/password, the client can then mount multiple shares without specifying a password for each instance. Samba can also accept session-based username/password requests. The client maintains multiple authentication contexts by using a unique UID for each logon.

In **`smb.conf`**, the **`security = user`** directive that sets user-level security is:

```

[GLOBAL]
...
security = user
...
```

The following sections describe other implementations of user-level security.

9.7.1.1. Domain Security Mode (User-Level Security)

In domain security mode, the Samba server has a machine account (domain security trust account) and causes all authentication requests to be passed through to the domain controllers. The Samba server is made into a domain member server by using the following directives in **`smb.conf`**:

```
[GLOBAL]
...
security = domain
workgroup = MARKETING
...
```

9.7.1.2. Active Directory Security Mode (User-Level Security)

If you have an Active Directory environment, it is possible to join the domain as a native Active Directory member. Even if a security policy restricts the use of NT-compatible authentication protocols, the Samba server can join an ADS using Kerberos. Samba in Active Directory member mode can accept Kerberos tickets.

In **`smb.conf`**, the following directives make Samba an Active Directory member server:

```

[GLOBAL]
...
security = ADS
realm = EXAMPLE.COM
password server = kerberos.example.com
...
```

9.7.1.3. Server Security Mode (User-Level Security)

Server security mode was previously used when Samba was not capable of acting as a domain member server.

> **Note**
>
> It is highly recommended to *not* use this mode since there are numerous security drawbacks.

In **`smb.conf`**, the following directives enable Samba to operate in server security mode:

```

[GLOBAL]
...
encrypt passwords = Yes
security = server
password server = "NetBIOS_of_Domain_Controller"
...
```

9.7.2. Share-Level Security

With share-level security, the server accepts only a password without an explicit username from the client. The server expects a password for each share, independent of the username. There have been recent reports that Microsoft Windows clients have compatibility issues with share-level security servers. Samba developers strongly discourage use of share-level security.

In **`smb.conf`**, the **`security = share`** directive that sets share-level security is:

```
[GLOBAL]
...
security = share
...
```

9.8. Samba Account Information Databases

The latest release of Samba offers many new features including new password database backends not previously available. Samba version 3.0.0 fully supports all databases used in previous versions of Samba. However, although supported, many backends may not be suitable for production use.

The following is a list different backends you can use with Samba. Other backends not listed here may also be available.

Plain Text

> Plain text backends are nothing more than the **`/etc/passwd`** type backends. With a plain text backend, all usernames and passwords are sent unencrypted between the client and the Samba server. This method is very unsecure and is not recommended for use by any means. It is possible that different Windows clients connecting to the Samba server with plain text passwords cannot support such an authentication method.

`smbpasswd`

> A popular backend used in previous Samba packages, the **`smbpasswd`** backend utilizes a plain ASCII text layout that includes the MS Windows LanMan and NT account, and encrypted password information. The **`smbpasswd`** backend lacks the storage of the Windows NT/2000/2003 SAM extended controls. The **`smbpasswd`** backend is not recommended because it does not scale well or hold any Windows information, such as RIDs for NT-based groups. The **`tdbsam`** backend solves these issues for use in a smaller database (250 users), but is still not an enterprise-class solution.

`ldapsam_compat`

> The **`ldapsam_compat`** backend allows continued OpenLDAP support for use with upgraded versions of Samba. This option normally used when migrating to Samba 3.0.

`tdbsam`

> The **`tdbsam`** backend provides an ideal database backend for local servers, servers that do not need built-in database replication, and servers that do not require the scalability or complexity of LDAP. The **`tdbsam`** backend includes all of the **`smbpasswd`** database information as well as the previously-excluded SAM information. The inclusion of the extended SAM data allows Samba to implement the same account and system access controls as seen with Windows NT/2000/2003-based systems.
>
> The **`tdbsam`** backend is recommended for 250 users at most. Larger organizations should require Active Directory or LDAP integration due to scalability and possible network infrastructure concerns.

`ldapsam`

> The **`ldapsam`** backend provides an optimal distributed account installation method for Samba. LDAP is optimal because of its ability to replicate its database to any number of servers using the

OpenLDAP **`slurpd`** daemon. LDAP databases are light-weight and scalable, and as such are preferred by large enterprises.

If you are upgrading from a previous version of Samba to 3.0, note that the **`/usr/share/doc/samba-<version>/LDAP/samba.schema`** has changed. This file contains the *attribute syntax definitions* and *objectclass definitions* that the **`ldapsam`** backend will need in order to function properly.

As such, if you are using the **`ldapsam`** backend for your Samba server, you will need to configure **`slapd`** to include this schema file. Refer to *Section 14.5, “The `/etc/openldap/schema/` Directory”* for directions on how to do this.

> **Note**
>
> You will need to have the **`openldap-server`** package installed if you want to use the **`ldapsam`** backend.

`mysqlsam`

The **`mysqlsam`** backend uses a MySQL-based database backend. This is useful for sites that already implement MySQL. At present, **`mysqlsam`** is now packed in a module separate from Samba, and as such is not officially supported by Samba.

9.9. Samba Network Browsing

Network browsing enables Windows and Samba servers to appear in the Windows **Network Neighborhood**. Inside the **Network Neighborhood**, icons are represented as servers and if opened, the server's shares and printers that are available are displayed.

Network browsing capabilities require NetBIOS over TCP/IP. NetBIOS-based networking uses broadcast (UDP) messaging to accomplish browse list management. Without NetBIOS and WINS as the primary method for TCP/IP hostname resolution, other methods such as static files (**`/etc/hosts`**) or DNS, must be used.

A domain master browser collates the browse lists from local master browsers on all subnets so that browsing can occur between workgroups and subnets. Also, the domain master browser should preferably be the local master browser for its own subnet.

9.9.1. Domain Browsing

By default, a Windows server PDC for a domain is also the domain master browser for that domain. A Samba server must *not* be set up as a domain master server in this type of situation

For subnets that do not include the Windows server PDC, a Samba server can be implemented as a local master browser. Configuring the **`smb.conf`** for a local master browser (or no browsing at all) in a domain controller environment is the same as workgroup configuration.

9.9.2. WINS (Windows Internetworking Name Server)

Either a Samba server or a Windows NT server can function as a WINS server. When a WINS server is used with NetBIOS enabled, UDP unicasts can be routed which allows name resolution across networks. Without a WINS server, the UDP broadcast is limited to the local subnet and therefore cannot be routed to other subnets, workgroups, or domains. If WINS replication is necessary, do not use Samba as your primary WINS server, as Samba does not currently support WINS replication.

In a mixed NT/2000/2003 server and Samba environment, it is recommended that you use the Microsoft WINS capabilities. In a Samba-only environment, it is recommended that you use *only one* Samba server for WINS.

The following is an example of the **`smb.conf`** file in which the Samba server is serving as a WINS server:

```
[global]
wins support = Yes
```

Tip

All servers (including Samba) should connect to a WINS server to resolve NetBIOS names. Without WINS, browsing only occurs on the local subnet. Furthermore, even if a domain-wide list is somehow obtained, hosts cannot be resolved for the client without WINS.

9.10. Samba with CUPS Printing Support

Samba allows client machines to share printers connected to the Samba server. In addition, Samba also allows client machines to send documents built in Linux to Windows printer shares. Although there are other printing systems that function with Fedora, CUPS (Common UNIX Print System) is the recommended printing system due to its close integration with Samba.

9.10.1. Simple `smb.conf` Settings

The following example shows a very basic **`smb.conf`** configuration for CUPS support:

```
[global]
load printers = Yes
printing = cups
printcap name = cups
[printers]
comment = All Printers
path = /var/spool/samba/print
printer = IBMInfoP
browseable = No
public = Yes
guest ok = Yes
writable = No
printable = Yes
printer admin = @ntadmins
[print$]
comment = Printer Drivers Share
path = /var/lib/samba/drivers
```

```
write list = ed, john
printer admin = ed, john
```

Other printing configurations are also possible. To add additional security and privacy for printing confidential documents, users can have their own print spooler not located in a public path. If a job fails, other users would not have access to the file.

The **`print$`** share contains printer drivers for clients to access if not available locally. The **`print$`** share is optional and may not be required depending on the organization.

Setting **`browseable`** to **`Yes`** enables the printer to be viewed in the Windows Network Neighborhood, provided the Samba server is set up correctly in the domain/workgroup.

9.11. Samba Distribution Programs

`findsmb`

`findsmb <`*`subnet_broadcast_address`*`>`

The **`findsmb`** program is a Perl script which reports information about SMB-aware systems on a specific subnet. If no subnet is specified the local subnet is used. Items displayed include IP address, NetBIOS name, workgroup or domain name, operating system, and version.

The following example shows the output of executing **`findsmb`** as any valid user on a system:

```

findsmb
IP ADDR       NETBIOS NAME  WORKGROUP/OS/VERSION
------------------------------------------------------------------
10.1.59.25    VERVE          [MYGROUP] [Unix] [Samba 3.0.0-15]
10.1.59.26    STATION22      [MYGROUP] [Unix] [Samba 3.0.2-7.FC1]
10.1.56.45    TREK          +[WORKGROUP] [Windows 5.0] [Windows 2000 LAN
 Manager]
10.1.57.94    PIXEL          [MYGROUP] [Unix] [Samba 3.0.0-15]
10.1.57.137   MOBILE001      [WORKGROUP] [Windows 5.0] [Windows 2000 LAN
 Manager]
10.1.57.141   JAWS          +[KWIKIMART] [Unix] [Samba 2.2.7a-security-
rollup-fix]
10.1.56.159   FRED          +[MYGROUP] [Unix] [Samba 3.0.0-14.3E]
10.1.59.192   LEGION        *[MYGROUP] [Unix] [Samba 2.2.7-security-rollup-
fix]
10.1.56.205   NANCYN        +[MYGROUP] [Unix] [Samba 2.2.7a-security-rollup-
fix]
```

`net`

`net <`*`protocol`*`> <`*`function`*`> <`*`misc_options`*`> <`*`target_options`*`>`

The **`net`** utility is similar to the **`net`** utility used for Windows and MS-DOS. The first argument is used to specify the protocol to use when executing a command. The **`<`*`protocol`*`>`** option can be **`ads`**, **`rap`**, or **`rpc`** for specifying the type of server connection. Active Directory uses **`ads`**, Win9x/NT3 uses **`rap`**, and Windows NT4/2000/2003 uses **rpc**. If the protocol is omitted, **net** automatically tries to determine it.

The following example displays a list the available shares for a host named **wakko**:

```
net -l share -S wakko
Password:
Enumerating shared resources (exports) on remote server:
Share name   Type     Description
----------   ----     -----------
data         Disk     Wakko data share
tmp          Disk     Wakko tmp share
IPC$         IPC      IPC Service (Samba Server)
ADMIN$       IPC      IPC Service (Samba Server)
```

The following example displays a list of Samba users for a host named **wakko**:

```
net -l user -S wakko
root password:
User name             Comment
-----------------------------
andriusb              Documentation
joe                   Marketing
lisa                  Sales
```

nmblookup

nmblookup <*options*> <*netbios_name*>

The **nmblookup** program resolves NetBIOS names into IP addresses. The program broadcasts its query on the local subnet until the target machine replies.

Here is an example:

```
nmblookup trek
querying trek on 10.1.59.255
10.1.56.45 trek<00>
```

pdbedit

pdbedit <*options*>

The **pdbedit** program manages accounts located in the SAM database. All backends are supported including **smbpasswd**, LDAP, NIS+, and the **tdb** database library.

The following are examples of adding, deleting, and listing users:

```
pdbedit -a kristin
```

```
new password:
retype new password:
Unix username:        kristin
NT username:
Account Flags:        [U          ]
User SID:             S-1-5-21-1210235352-3804200048-1474496110-2012
Primary Group SID:    S-1-5-21-1210235352-3804200048-1474496110-2077
Full Name: Home Directory:       \\wakko\kristin
HomeDir Drive:
Logon Script:
Profile Path:         \\wakko\kristin\profile
Domain:               WAKKO
Account desc:
Workstations: Munged
dial:
Logon time:           0
Logoff time:          Mon, 18 Jan 2038 22:14:07 GMT
Kickoff time:         Mon, 18 Jan 2038 22:14:07 GMT
Password last set:    Thu, 29 Jan 2004 08:29:28
GMT Password can change:  Thu, 29 Jan 2004 08:29:28 GMT
Password must change: Mon, 18 Jan 2038 22:14:07 GMT
pdbedit -v -L kristin
Unix username:        kristin
NT username:
Account Flags:        [U          ]
User SID:             S-1-5-21-1210235352-3804200048-1474496110-2012
Primary Group SID:    S-1-5-21-1210235352-3804200048-1474496110-2077
Full Name:
Home Directory:       \\wakko\kristin
HomeDir Drive:
Logon Script:
Profile Path:         \\wakko\kristin\profile
Domain:               WAKKO
Account desc:
Workstations: Munged
dial:
Logon time:           0
Logoff time:          Mon, 18 Jan 2038 22:14:07 GMT
Kickoff time:         Mon, 18 Jan 2038 22:14:07 GMT
Password last set:    Thu, 29 Jan 2004 08:29:28 GMT
Password can change:  Thu, 29 Jan 2004 08:29:28 GMT
Password must change: Mon, 18 Jan 2038 22:14:07 GMT
pdbedit -L
andriusb:505:
joe:503:
lisa:504:
kristin:506:
pdbedit -x joe
pdbedit -L
andriusb:505: lisa:504: kristin:506:
```

rpcclient

rpcclient ***<server> <options>***

The **rpcclient** program issues administrative commands using Microsoft RPCs, which provide access to the Windows administration graphical user interfaces (GUIs) for systems management. This is most often used by advanced users that understand the full complexity of Microsoft RPCs.

smbcacls

smbcacls ***<//server/share> <filename> <options>***

The **smbcacls** program modifies Windows ACLs on files and directories shared by the Samba server.

smbclient

smbclient ***<//server/share> <password> <options>***

The **smbclient** program is a versatile UNIX client which provides functionality similar to **ftp**.

smbcontrol

smbcontrol -i ***<options>***

smbcontrol ***<options> <destination> <messagetype> <parameters>***

The **smbcontrol** program sends control messages to running **smbd** or **nmbd** daemons. Executing **smbcontrol -i** runs commands interactively until a blank line or a 'q' is entered.

smbpasswd

smbpasswd ***<options> <username> <password>***

The **smbpasswd** program manages encrypted passwords. This program can be run by a superuser to change any user's password as well as by an ordinary user to change their own Samba password.

smbspool

smbspool ***<job> <user> <title> <copies> <options> <filename>***

The **smbspool** program is a CUPS-compatible printing interface to Samba. Although designed for use with CUPS printers, **smbspool** can work with non-CUPS printers as well.

smbstatus

smbstatus ***<options>***

The **smbstatus** program displays the status of current connections to a Samba server.

smbtar

smbtar ***<options>***

The **smbtar** program performs backup and restores of Windows-based share files and directories to a local tape archive. Though similar to the **tar** command, the two are not compatible.

testparm

testparm *<options> <filename> <hostname IP_address>*

The **testparm** program checks the syntax of the **smb.conf** file. If your **smb.conf** file is in the default location (**/etc/samba/smb.conf**) you do not need to specify the location. Specifying the hostname and IP address to the **testparm** program verifies that the **hosts.allow** and **host.deny** files are configured correctly. The **testparm** program also displays a summary of your **smb.conf** file and the server's role (stand-alone, domain, etc.) after testing. This is convenient when debugging as it excludes comments and concisely presents information for experienced administrators to read.

For example:

```
testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[tmp]"
Processing section "[html]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
<enter>
# Global parameters
[global]
 workgroup = MYGROUP
 server string = Samba Server
 security = SHARE
 log file = /var/log/samba/%m.log
 max log size = 50
 socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
 dns proxy = No
[homes]
 comment = Home Directories
 read only = No
 browseable = No
[printers]
 comment = All Printers
 path = /var/spool/samba
 printable = Yes
 browseable = No
[tmp]
 comment = Wakko tmp
 path = /tmp
 guest only = Yes
[html]
 comment = Wakko www
 path = /var/www/html
 force user = andriusb
 force group = users
 read only = No
```

```
guest only = Yes
```

`wbinfo`

`wbinfo <options>`

The **`wbinfo`** program displays information from the **`winbindd`** daemon. The **`winbindd`** daemon must be running for **`wbinfo`** to work.

9.12. Additional Resources

The following sections give you the means to explore Samba in greater detail.

9.12.1. Installed Documentation

- **`/usr/share/doc/samba-<version-number>/`** — All additional files included with the Samba distribution. This includes all helper scripts, sample configuration files, and documentation.

  This directory also contains online versions of *The Official Samba-3 HOWTO-Collection* and *Samba-3 by Example*, both of which are cited below.

9.12.2. Related Books

- *The Official Samba-3 HOWTO-Collection* by John H. Terpstra and Jelmer R. Vernooij; Prentice Hall — The official Samba-3 documentation as issued by the Samba development team. This is more of a reference guide than a step-by-step guide.

- *Samba-3 by Example* by John H. Terpstra; Prentice Hall — This is another official release issued by the Samba development team which discusses detailed examples of OpenLDAP, DNS, DHCP, and printing configuration files. This has step-by-step related information that helps in real-world implementations.

- *Using Samba, 2nd Edition* by Jay T's, Robert Eckstein, and David Collier-Brown; O'Reilly — A good resource for novice to advanced users, which includes comprehensive reference material.

9.12.3. Useful Websites

- *http://www.samba.org/* — Homepage for the Samba distribution and all official documentation created by the Samba development team. Many resources are available in HTML and PDF formats, while others are only available for purchase. Although many of these links are not Fedora specific, some concepts may apply.

- *http://samba.org/samba/archives.html* [1] — Active email lists for the Samba community. Enabling digest mode is recommended due to high levels of list activity.

- Samba newsgroups — Samba threaded newsgroups, such as gmane.org, that use the NNTP protocol are also available. This an alternative to receiving mailing list emails.

- *hhttp://sourceforge.net/projects/smbldap-tools/* [2] — These are highly recommended for assisting in managing LDAP related resources. The scripts can be found at **`/usr/share/doc/samba-version_number/LDAP/smbldap-tools`** or can be downloaded from Sourceforge.

Chapter 10.

Dynamic Host Configuration Protocol (DHCP)

Dynamic Host Configuration Protocol (DHCP) is a network protocol that automatically assigns TCP/IP information to client machines. Each DHCP client connects to the centrally located DHCP server, which returns that client's network configuration (including the IP address, gateway, and DNS servers).

10.1. Why Use DHCP?

DHCP is useful for automatic configuration of client network interfaces. When configuring the client system, the administrator chooses DHCP instead of specifying an IP address, netmask, gateway, or DNS servers. The client retrieves this information from the DHCP server. DHCP is also useful if an administrator wants to change the IP addresses of a large number of systems. Instead of reconfiguring all the systems, he can just edit one DHCP configuration file on the server for the new set of IP addresses. If the DNS servers for an organization changes, the changes are made on the DHCP server, not on the DHCP clients. When the administrator restarts the network or reboots the clients, the changes will go into effect.

If an organization has a functional DHCP server properly connected to a network, laptops and other mobile computer users can move these devices from office to office.

10.2. Configuring a DHCP Server

The **dhcp** package contains an ISC DHCP server. First, install the package as the superuser:

```
~]# yum install dhcp
```

Installing the **dhcp** package creates a file, **`/etc/dhcpd.conf`**, which is merely an empty configuration file:

```
~]# cat /etc/dhcpd.conf
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.sample
```

The sample configuration file can be found at **`/usr/share/doc/dhcp-<`*`version`*`>/dhcpd.conf.sample`**. You should use this file to help you configure **`/etc/dhcpd.conf`**, which is explained in detail below.

DHCP also uses the file **`/var/lib/dhcpd/dhcpd.leases`** to store the client lease database. Refer to *Section 10.2.2, “Lease Database”* for more information.

10.2.1. Configuration File

The first step in configuring a DHCP server is to create the configuration file that stores the network information for the clients. Use this file to declare options and global options for client systems.

The configuration file can contain extra tabs or blank lines for easier formatting. Keywords are case-insensitive and lines beginning with a hash mark (#) are considered comments.

Two DNS update schemes are currently implemented — the ad-hoc DNS update mode and the interim DHCP-DNS interaction draft update mode. If and when these two are accepted as part of the Internet Engineering Task Force (IETF) standards process, there will be a third mode — the standard DNS update method. You must configure the DNS server for compatibility with these schemes. Version 3.0b2pl11 and previous versions used the ad-hoc mode; however, it has been deprecated. To keep the same behavior, add the following line to the top of the configuration file:

```
ddns-update-style ad-hoc;
```

To use the recommended mode, add the following line to the top of the configuration file:

```
ddns-update-style interim;
```

Refer to the **dhcpd.conf** man page for details about the different modes.

There are two types of statements in the configuration file:

- Parameters — State how to perform a task, whether to perform a task, or what network configuration options to send to the client.
- Declarations — Describe the topology of the network, describe the clients, provide addresses for the clients, or apply a group of parameters to a group of declarations.

The parameters that start with the keyword option are reffered to as *options*. These options control DHCP options; whereas, parameters configure values that are not optional or control how the DHCP server behaves.

Parameters (including options) declared before a section enclosed in curly brackets ({ }) are considered global parameters. Global parameters apply to all the sections below it.

Important

If the configuration file is changed, the changes do not take effect until the DHCP daemon is restarted with the command **service dhcpd restart**.

Tip

Instead of changing a DHCP configuration file and restarting the service each time, using the **omshell** command provides an interactive way to connect to, query, and change the configuration of a DHCP server. By using **omshell**, all changes can be made while the server is running. For more information on **omshell**, refer to the **omshell** man page.

In *Example 10.1, “Subnet Declaration”*, the **routers**, **subnet-mask**, **domain-name**, **domain-name-servers**, and **time-offset** options are used for any **host** statements declared below it.

Additionally, a **subnet** can be declared, a **subnet** declaration must be included for every subnet in the network. If it is not, the DHCP server fails to start.

In this example, there are global options for every DHCP client in the subnet and a **range** declared. Clients are assigned an IP address within the **range**.

```
subnet 192.168.1.0 netmask 255.255.255.0 {
        option routers                  192.168.1.254;
        option subnet-mask              255.255.255.0;

        option domain-name              "example.com";
        option domain-name-servers       192.168.1.1;

        option time-offset              -18000;     # Eastern Standard Time

 range 192.168.1.10 192.168.1.100;
}
```

Example 10.1. Subnet Declaration

All subnets that share the same physical network should be declared within a **`shared-network`** declaration as shown in *Example 10.2, “Shared-network Declaration”*. Parameters within the **`shared-network`**, but outside the enclosed **`subnet`** declarations, are considered to be global parameters. The name of the **`shared-network`** must be a descriptive title for the network, such as using the title 'test-lab' to describe all the subnets in a test lab environment.

```
shared-network name {
    option domain-name              "test.redhat.com";
    option domain-name-servers      ns1.redhat.com, ns2.redhat.com;
    option routers                  192.168.0.254;
    more parameters for EXAMPLE shared-network
    subnet 192.168.1.0 netmask 255.255.252.0 {
        parameters for subnet
        range 192.168.1.1 192.168.1.254;
    }
    subnet 192.168.2.0 netmask 255.255.252.0 {
        parameters for subnet
        range 192.168.2.1 192.168.2.254;
    }
}
```

Example 10.2. Shared-network Declaration

As demonstrated in *Example 10.3, “Group Declaration”*, the **`group`** declaration is used to apply global parameters to a group of declarations. For example, shared networks, subnets, and hosts can be grouped.

```
group {
   option routers                  192.168.1.254;
   option subnet-mask              255.255.255.0;

   option domain-name              "example.com";
   option domain-name-servers       192.168.1.1;

   option time-offset              -18000;     # Eastern Standard Time

   host apex {
      option host-name "apex.example.com";
      hardware ethernet 00:A0:78:8E:9E:AA;
      fixed-address 192.168.1.4;
   }

   host raleigh {
      option host-name "raleigh.example.com";
      hardware ethernet 00:A1:DD:74:C3:F2;
      fixed-address 192.168.1.6;
   }
}
```

Example 10.3. Group Declaration

To configure a DHCP server that leases a dynamic IP address to a system within a subnet, modify *Example 10.4, “Range Parameter”* with your values. It declares a default lease time, maximum lease time, and network configuration values for the clients. This example assigns IP addresses in the **`range`** 192.168.1.10 and 192.168.1.100 to client systems.

```
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option domain-name "example.com";

subnet 192.168.1.0 netmask 255.255.255.0 {
   range 192.168.1.10 192.168.1.100;
}
```

Example 10.4. Range Parameter

To assign an IP address to a client based on the MAC address of the network interface card, use the **`hardware ethernet`** parameter within a **`host`** declaration. As demonstrated in *Example 10.5, “Static IP Address using DHCP”*, the **`host apex`** declaration specifies that the network interface card with the MAC address 00:A0:78:8E:9E:AA always receives the IP address 192.168.1.4.

Note that the optional parameter **`host-name`** can also be used to assign a host name to the client.

```
host apex {
   option host-name "apex.example.com";
   hardware ethernet 00:A0:78:8E:9E:AA;
   fixed-address 192.168.1.4;
}
```

Example 10.5. Static IP Address using DHCP

Tip

The sample configuration file provided can be used as a starting point and custom configuration options can be added to it. To copy it to the proper location, use the following command:

```
cp /usr/share/doc/dhcp-<version-number>/dhcpd.conf.sample /etc/
dhcpd.conf
```

(where <*version-number*> is the DHCP version number).

For a complete list of option statements and what they do, refer to the **dhcp-options** man page.

10.2.2. Lease Database

On the DHCP server, the file **/var/lib/dhcpd/dhcpd.leases** stores the DHCP client lease database. Do not change this file. DHCP lease information for each recently assigned IP address is automatically stored in the lease database. The information includes the length of the lease, to whom the IP address has been assigned, the start and end dates for the lease, and the MAC address of the network interface card that was used to retrieve the lease.

All times in the lease database are in Coordinated Universal Time (UTC), not local time.

The lease database is recreated from time to time so that it is not too large. First, all known leases are saved in a temporary lease database. The **dhcpd.leases** file is renamed **dhcpd.leases~** and the temporary lease database is written to **dhcpd.leases**.

The DHCP daemon could be killed or the system could crash after the lease database has been renamed to the backup file but before the new file has been written. If this happens, the **dhcpd.leases** file does not exist, but it is required to start the service. Do not create a new lease file. If you do, all old leases are lost which causes many problems. The correct solution is to rename the **dhcpd.leases~** backup file to **dhcpd.leases** and then start the daemon.

10.2.3. Starting and Stopping the Server

Important

When the DHCP server is started for the first time, it fails unless the **dhcpd.leases** file exists. Use the command **touch /var/lib/dhcpd/dhcpd.leases** to create the file if it does not exist.

If the same server is also running BIND as a DNS server, this step is not necessary, as starting the **named** service automatically checks for a **dhcpd.leases** file.

To start the DHCP service, use the command **/sbin/service dhcpd start**. To stop the DHCP server, use the command **/sbin/service dhcpd stop**.

By default, the DHCP service does not start at boot time. To configure the daemon to start automatically at boot time, refer to *Chapter 6, Controlling Access to Services*.

If more than one network interface is attached to the system, but the DHCP server should only be started on one of the interfaces, configure the DHCP server to start only on that device. In **/etc/sysconfig/dhcpd**, add the name of the interface to the list of **DHCPDARGS**:

```
# Command line options here
DHCPDARGS=eth0
```

This is useful for a firewall machine with two network cards. One network card can be configured as a DHCP client to retrieve an IP address to the Internet. The other network card can be used as a DHCP server for the internal network behind the firewall. Specifying only the network card connected to the internal network makes the system more secure because users can not connect to the daemon via the Internet.

Other command line options that can be specified in **/etc/sysconfig/dhcpd** include:

- **-p *<portnum>*** — Specifies the UDP port number on which **dhcpd** should listen. The default is port 67. The DHCP server transmits responses to the DHCP clients at a port number one greater than the UDP port specified. For example, if the default port 67 is used, the server listens on port 67 for requests and responses to the client on port 68. If a port is specified here and the DHCP relay agent is used, the same port on which the DHCP relay agent should listen must be specified. Refer to *Section 10.2.4, “DHCP Relay Agent”* for details.

- **-f** — Runs the daemon as a foreground process. This is mostly used for debugging.

- **-d** — Logs the DHCP server daemon to the standard error descriptor. This is mostly used for debugging. If this is not specified, the log is written to **/var/log/messages**.

- **-cf *<filename>*** — Specifies the location of the configuration file. The default location is **/etc/dhcpd.conf**.

- **-lf *<filename>*** — Specifies the location of the lease database file. If a lease database file already exists, it is very important that the same file be used every time the DHCP server is started. It is strongly recommended that this option only be used for debugging purposes on non-production machines. The default location is **/var/lib/dhcpd/dhcpd.leases**.

- **-q** — Do not print the entire copyright message when starting the daemon.

10.2.4. DHCP Relay Agent

The DHCP Relay Agent (**dhcrelay**) allows for the relay of DHCP and BOOTP requests from a subnet with no DHCP server on it to one or more DHCP servers on other subnets.

When a DHCP client requests information, the DHCP Relay Agent forwards the request to the list of DHCP servers specified when the DHCP Relay Agent is started. When a DHCP server returns a reply, the reply is broadcast or unicast on the network that sent the original request.

The DHCP Relay Agent listens for DHCP requests on all interfaces unless the interfaces are specified in **`/etc/sysconfig/dhcrelay`** with the `INTERFACES` directive.

To start the DHCP Relay Agent, use the command **`service dhcrelay start`**.

10.3. Configuring a DHCP Client

The first step for configuring a DHCP client is to make sure the kernel recognizes the network interface card. Most cards are recognized during the installation process and the system is configured to use the correct kernel module for the card. If a card is added after installation, **Kudzu**[1] will recognize it and prompt you for the proper kernel module (Be sure to check the Hardware Compatibility List at *http://hardware.redhat.com/hcl/*). If either the installation program or kudzu does not recognize the network card, you can load the correct kernel module (refer to *Chapter 30, General Parameters and Modules* for details).

To configure a DHCP client manually, modify the **`/etc/sysconfig/network`** file to enable networking and the configuration file for each network device in the **`/etc/sysconfig/network-scripts`** directory. In this directory, each device should have a configuration file named **`ifcfg-eth0`**, where **`eth0`** is the network device name.

The **`/etc/sysconfig/network`** file should contain the following line:

```
NETWORKING=yes
```

The `NETWORKING` variable must be set to `yes` if you want networking to start at boot time.

The **`/etc/sysconfig/network-scripts/ifcfg-eth0`** file should contain the following lines:

```

DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

A configuration file is needed for each device to be configured to use DHCP.

Other options for the network script includes:

- **`DHCP_HOSTNAME`** — Only use this option if the DHCP server requires the client to specify a hostname before receiving an IP address. (The DHCP server daemon in Fedora does not support this feature.)

- **`PEERDNS=<answer>`**, where **`<answer>`** is one of the following:

  - **`yes`** — Modify **`/etc/resolv.conf`** with information from the server. If using DHCP, then **`yes`** is the default.

  - **`no`** — Do not modify **`/etc/resolv.conf`**.

[1] **Kudzu** is a hardware probing tool run at system boot time to determine what hardware has been added or removed from the system.

- **SRCADDR=*<address>***, where ***<address>*** is the specified source IP address for outgoing packets.
- **USERCTL=*<answer>***, where ***<answer>*** is one of the following:
  - **yes** — Non-root users are allowed to control this device.
  - **no** — Non-root users are not allowed to control this device.

If you prefer using a graphical interface, refer to *Chapter 5, Network Configuration* for instructions on using the **Network Administration Tool** to configure a network interface to use DHCP.

> **Tip**
>
> For advanced configurations of client DHCP options such as protocol timing, lease requirements and requests, dynamic DNS support, aliases, as well as a wide variety of values to override, prepend, or append to client-side configurations, refer to the **dhclient** and **dhclient.conf** man pages.

10.4. Configuring a Multihomed DHCP Server

A multihomed DHCP server serves multiple networks, that is, multiple subnets. The examples in these sections detail how to configure a DHCP server to serve multiple networks, select which network interfaces to listen on, and how to define network settings for systems that move networks.

Before making any changes, back up the existing **/etc/sysconfig/dhcpd** and **/etc/dhcpd.conf** files.

The DHCP daemon listens on all network interfaces unless otherwise specified. Use the **/etc/sysconfig/dhcpd** file to specify which network interfaces the DHCP daemon listens on. The following **/etc/sysconfig/dhcpd** example specifies that the DHCP daemon listens on the **eth0** and **eth1** interfaces:

```
DHCPDARGS="eth0 eth1";
```

If a system has three network interfaces cards -- **eth0**, **eth1**, and **eth2** -- and it is only desired that the DHCP daemon listens on **eth0**, then only specify `eth0` in **/etc/sysconfig/dhcpd**:

```
DHCPDARGS="eth0";
```

The following is a basic **/etc/dhcpd.conf** file, for a server that has two network interfaces, **eth0** in a 10.0.0.0/24 network, and **eth1** in a 172.16.0.0/24 network. Multiple `subnet` declarations allow different settings to be defined for multiple networks:

```
ddns-update-style interim;
default-lease-time 600;
max-lease-time 7200;
```

```
subnet 10.0.0.0 netmask 255.255.255.0 {
 option subnet-mask 255.255.255.0;
 option routers 10.0.0.1;
 range 10.0.0.5 10.0.0.15;
}

subnet 172.16.0.0 netmask 255.255.255.0 {
 option subnet-mask 255.255.255.0;
 option routers 172.16.0.1;
 range 172.16.0.5 172.16.0.15;

}
```

subnet *10.0.0.0* netmask *255.255.255.0*

A subnet declaration is required for every network your DHCP server is serving. Multiple subnets require multiple subnet declarations. If the DHCP server does not have a network interface in a range of a subnet declaration, the DHCP server does not serve that network.

If there is only one subnet declaration, and no network interfaces are in the range of that subnet, the DHCP daemon fails to start, and an error such as the following is logged to **/var/log/messages**:

```
dhcpd: No subnet declaration for eth0 (0.0.0.0).
dhcpd: ** Ignoring requests on eth0.  If this is not what
dhcpd:    you want, please write a subnet declaration
dhcpd:    in your dhcpd.conf file for the network segment
dhcpd:    to which interface eth1 is attached. **
dhcpd:
dhcpd:
dhcpd: Not configured to listen on any interfaces!
```

option subnet-mask *255.255.255.0*;

The option subnet-mask option defines a subnet mask, and overrides the netmask value in the subnet declaration. In simple cases, the subnet and netmask values are the same.

option routers *10.0.0.1*;

The option routers option defines the default gateway for the subnet. This is required for systems to reach internal networks on a different subnet, as well as external networks.

range *10.0.0.5 10.0.0.15*;

The range option specifies the pool of available IP addresses. Systems are assigned an address from the range of specified IP addresses.

For further information, refer to the dhcpd.conf(5) man page.

Alias Interfaces

Alias interfaces are not supported by DHCP. If an alias interface is the only interface, in the only subnet specified in **/etc/dhcpd.conf**, the DHCP daemon fails to start.

10.4.1. Host Configuration

Before making any changes, back up the existing **/etc/sysconfig/dhcpd** and **/etc/dhcpd.conf** files.

Configuring a single system for multiple networks

The following **/etc/dhcpd.conf** example creates two subnets, and configures an IP address for the same system, depending on which network it connects to:

```
ddns-update-style interim;
default-lease-time 600;
max-lease-time 7200;

subnet 10.0.0.0 netmask 255.255.255.0 {
 option subnet-mask 255.255.255.0;
 option routers 10.0.0.1;
 range 10.0.0.5 10.0.0.15;
}

subnet 172.16.0.0 netmask 255.255.255.0 {
 option subnet-mask 255.255.255.0;
 option routers 172.16.0.1;
 range 172.16.0.5 172.16.0.15;

}

host example0 {
 hardware ethernet 00:1A:6B:6A:2E:0B;
 fixed-address 10.0.0.20;
}

host example1 {
 hardware ethernet 00:1A:6B:6A:2E:0B;
 fixed-address 172.16.0.20;
}
```

host *example0*

The host declaration defines specific parameters for a single system, such as an IP address. To configure specific parameters for multiple hosts, use multiple host declarations.

Most DHCP clients ignore the name in host declarations, and as such, this name can anything, as long as it is unique to other host declarations. To configure the same system for multiple networks, use a different name for each host declaration, otherwise the DHCP daemon fails to start. Systems are identified by the hardware ethernet option, not the name in the host declaration.

hardware ethernet *00:1A:6B:6A:2E:0B*;

The hardware ethernet option identifies the system. To find this address, run the **ifconfig** command on the desired system, and look for the HWaddr address.

`fixed-address` *`10.0.0.20`*`;`

The `fixed-address` option assigns a valid IP address to the system specified by the `hardware ethernet` option. This address must be outside the IP address pool specified with the `range` option.

If `option` statements do not end with a semicolon, the DHCP daemon fails to start, and an error such as the following is logged to **`/var/log/messages`**:

```
/etc/dhcpd.conf line 20: semicolon expected.
dhcpd: }
dhcpd: ^
dhcpd: /etc/dhcpd.conf line 38: unexpected end of file
dhcpd:
dhcpd: ^
dhcpd: Configuration file errors encountered -- exiting
```

Configuring systems with multiple network interfaces

The following `host` declarations configure a single system, that has multiple network interfaces, so that each interface receives the same IP address. This configuration will not work if both network interfaces are connected to the same network at the same time:

```
host interface0 {
 hardware ethernet 00:1a:6b:6a:2e:0b;
 fixed-address 10.0.0.18;
}

host interface1 {
 hardware ethernet 00:1A:6B:6A:27:3A;
 fixed-address 10.0.0.18;
}
```

For this example, `interface0` is the first network interface, and `interface1` is the second interface. The different `hardware ethernet` options identify each interface.

If such a system connects to another network, add more `host` declarations, remembering to:

- assign a valid `fixed-address` for the network the host is connecting to.

- make the name in the `host` declaration unique.

When a name given in a `host` declaration is not unique, the DHCP daemon fails to start, and an error such as the following is logged to **`/var/log/messages`**:

```
dhcpd: /etc/dhcpd.conf line 31: host interface0: already exists
dhcpd: }
dhcpd: ^
dhcpd: Configuration file errors encountered -- exiting
```

This error was caused by having multiple `host interface0` declarations defined in **`/etc/dhcpd.conf`**.

10.5. Additional Resources

For additional configuration options, refer to the following resources.

10.5.1. Installed Documentation

- **`dhcpd`** man page — Describes how the DHCP daemon works.
- **`dhcpd.conf`** man page — Explains how to configure the DHCP configuration file; includes some examples.
- **`dhcpd.leases`** man page — Explains how to configure the DHCP leases file; includes some examples.
- **`dhcp-options`** man page — Explains the syntax for declaring DHCP options in **`dhcpd.conf`**; includes some examples.
- **`dhcrelay`** man page — Explains the DHCP Relay Agent and its configuration options.
- **`/usr/share/doc/dhcp-<`*`version`*`>/`** — Contains sample files, README files, and release notes for current versions of the DHCP service.

Apache HTTP Server

The Apache HTTP Server is a robust, commercial-grade open source Web server developed by the Apache Software Foundation (*http://www.apache.org/*). Fedora 12 includes the Apache HTTP Server 2.2 as well as a number of server modules designed to enhance its functionality.

The default configuration file installed with the Apache HTTP Server works without alteration for most situations. This chapter outlines many of the directives found within its configuration file (**`/etc/httpd/conf/httpd.conf`**) to aid those who require a custom configuration or need to convert a configuration file from the older Apache HTTP Server 1.3 format.

> **Warning**
>
> If using the graphical **HTTP Configuration Tool** (*system-config-httpd*), *do not* hand edit the Apache HTTP Server's configuration file as the **HTTP Configuration Tool** regenerates this file whenever it is used.

11.1. Apache HTTP Server 2.2

There are important differences between the Apache HTTP Server 2.2 and version 2.0.

This section reviews some of the features of Apache HTTP Server 2.2 and outlines important changes. If you are upgrading from version 1.3, you should also read the instructions on migrating from version 1.3 to version 2.0. For instructions on migrating a version 1.3 configuration file to the 2.0 format, refer to *Section 11.2.2, “Migrating Apache HTTP Server 1.3 Configuration Files to 2.0”*.

11.1.1. Features of Apache HTTP Server 2.2

Apache HTTP Server 2.2 features the following improvements over version 2.0 :

- Improved caching modules (mod_cache, mod_disk_cache, mod_mem_cache).
- A new structure for authentication and authorization support, replacing the authentication modules provided in previous versions.
- Support for proxy load balancing (mod_proxy_balancer)
- support for handling large files (namely, greater than 2GB) on 32-bit platforms

The following changes have been made to the default httpd configuration:

- The mod_cern_meta and mod_asis modules are no longer loaded by default.
- The mod_ext_filter module is now loaded by default.

If upgrading from a previous release of Fedora, the httpd configuration will need to be updated for httpd 2.2. For more information, refer to http://httpd.apache.org/docs/2.2/upgrading.html

11.2. Migrating Apache HTTP Server Configuration Files

11.2.1. Migrating Apache HTTP Server 2.0 Configuration Files

This section outlines migration from version 2.0 to 2.2. If you are migrating from version 1.3, please refer to *Section 11.2.2, “Migrating Apache HTTP Server 1.3 Configuration Files to 2.0”*.

- Configuration files and startup scripts from version 2.0 need minor adjustments particularly in module names which may have changed. Third party modules which worked in version 2.0 can also work in version 2.2 but need to be recompiled before you load them. Key modules that need to be noted are authentication and authorization modules. For each of the modules which has been renamed the ***`LoadModule`***[1] line will need to be updated.

- The **`mod_userdir`** module will only act on requests if you provide a **`UserDir`** directive indicating a directory name. If you wish to maintain the procedures used in version 2.0, add the directive **`UserDir public_html`** in your configuration file.

- To enable SSL, edit the **`httpd.conf`** file adding the necessary **`mod_ssl`** directives. Use **`apachectl start`** as **`apachectl startssl`** is unavailable in version 2.2. You can view an example of SSL configuration for httpd in **`conf/extra/httpd-ssl.conf`**.

- To test your configuration it is advisable to use **`service httpd configtest`** which will detect configuration errors.

More information on upgrading from version 2.0 to 2.2 can be found on *http://httpd.apache.org/docs/2.2/upgrading.html*.

11.2.2. Migrating Apache HTTP Server 1.3 Configuration Files to 2.0

This section details migrating an Apache HTTP Server 1.3 configuration file to be utilized by Apache HTTP Server 2.0.

If the **`/etc/httpd/conf/httpd.conf`** file is a modified version of the newly installed default and a saved a copy of the original configuration file is available, it may be easiest to invoke the **`diff`** command, as in the following example (logged in as root):

```
diff -u httpd.conf.orig httpd.conf | less
```

This command highlights any modifications made. If a copy of the original file is not available, extract it from an RPM package using the **`rpm2cpio`** and **`cpio`** commands, as in the following example:

```
rpm2cpio apache-<version-number>.i386.rpm | cpio -i --make
```

In the above command, replace <*`version-number`*> with the version number for the **`apache`** package.

Finally, it is useful to know that the Apache HTTP Server has a testing mode to check for configuration errors. To use access it, type the following command:

```
apachectl configtest
```

11.2.2.1. Global Environment Configuration

The global environment section of the configuration file contains directives which affect the overall operation of the Apache HTTP Server, such as the number of concurrent requests it can handle and the locations of the various files. This section requires a large number of changes and should be based on the Apache HTTP Server 2.0 configuration file, while migrating the old settings into it.

11.2.2.1.1. Interface and Port Binding

The **`BindAddress`** and **`Port`** directives no longer exist; their functionality is now provided by a more flexible **`Listen`** directive.

If **`Port 80`** was set in the 1.3 version configuration file, change it to **`Listen 80`** in the 2.0 configuration file. If **`Port`** was set to some value *other than 80*, then append the port number to the contents of the **`ServerName`** directive.

For example, the following is a sample Apache HTTP Server 1.3 directive:

```

Port 123 ServerName www.example.com
```

To migrate this setting to Apache HTTP Server 2.0, use the following structure:

```

Listen 123 ServerName www.example.com:123
```

For more on this topic, refer to the following documentation on the Apache Software Foundation's website:

- *http://httpd.apache.org/docs-2.0/mod/mpm_common.html#listen*
- *http://httpd.apache.org/docs-2.0/mod/core.html#servername*

11.2.2.1.2. Server-Pool Size Regulation

When the Apache HTTP Server accepts requests, it dispatches child processes or threads to handle them. This group of child processes or threads is known as a *server-pool*. Under Apache HTTP Server 2.0, the responsibility for creating and maintaining these server-pools has been abstracted to a group of modules called *Multi-Processing Modules* (*MPMs*). Unlike other modules, only one module from the MPM group can be loaded by the Apache HTTP Server. There are three MPM modules that ship with 2.0: **`prefork`**, **`worker`**, and **`perchild`**. Currently only the **`prefork`** and **`worker`** MPMs are available, although the **`perchild`** MPM may be available at a later date.

The original Apache HTTP Server 1.3 behavior has been moved into the **`prefork`** MPM. The **`prefork`** MPM accepts the same directives as Apache HTTP Server 1.3, so the following directives may be migrated directly:

- **`StartServers`**
- **`MinSpareServers`**
- **`MaxSpareServers`**

- **`MaxClients`**
- **`MaxRequestsPerChild`**

The **`worker`** MPM implements a multi-process, multi-threaded server providing greater scalability. When using this MPM, requests are handled by threads, conserving system resources and allowing large numbers of requests to be served efficiently. Although some of the directives accepted by the **`worker`** MPM are the same as those accepted by the **`prefork`** MPM, the values for those directives should not be transfered directly from an Apache HTTP Server 1.3 installation. It is best to instead use the default values as a guide, then experiment to determine what values work best.

Important

To use the **`worker`** MPM, create the file **`/etc/sysconfig/httpd`** and add the following directive:

```
HTTPD=/usr/sbin/httpd.worker
```

For more on the topic of MPMs, refer to the following documentation on the Apache Software Foundation's website:

- *http://httpd.apache.org/docs-2.0/mpm.html*

11.2.2.1.3. Dynamic Shared Object (DSO) Support

There are many changes required here, and it is highly recommended that anyone trying to modify an Apache HTTP Server 1.3 configuration to suit version 2.0 (as opposed to migrating the changes into the version 2.0 configuration) copy this section from the stock Apache HTTP Server 2.0 configuration file.

Those who do not want to copy the section from the stock Apache HTTP Server 2.0 configuration should note the following:

- The **`AddModule`** and **`ClearModuleList`** directives no longer exist. These directives where used to ensure that modules could be enabled in the correct order. The Apache HTTP Server 2.0 API allows modules to specify their ordering, eliminating the need for these two directives.
- The order of the **`LoadModule`** lines are no longer relevant in most cases.
- Many modules have been added, removed, renamed, split up, or incorporated into others.
- **`LoadModule`** lines for modules packaged in their own RPMs (**`mod_ssl`**, **`php`**, **`mod_perl`**, and the like) are no longer necessary as they can be found in their relevant files within the **`/etc/httpd/conf.d/`** directory.
- The various **`HAVE_XXX`** definitions are no longer defined.

Important

If modifying the original file, note that it is of paramount importance that the **`httpd.conf`** contains the following directive:

```
Include conf.d/*.conf
```

Omission of this directive results in the failure of all modules packaged in their own RPMs (such as **mod_perl**, **php**, and **mod_ssl**).

11.2.2.1.4. Other Global Environment Changes

The following directives have been removed from Apache HTTP Server 2.0's configuration:

- ***ServerType*** — The Apache HTTP Server can only be run as **ServerType standalone** making this directive irrelevant.
- ***AccessConfig*** and ***ResourceConfig*** — These directives have been removed as they mirror the functionality of the **Include** directive. If the **AccessConfig** and **ResourceConfig** directives are set, replace them with **Include** directives.

  To ensure that the files are read in the order implied by the older directives, the **Include** directives should be placed at the end of the **httpd.conf**, with the one corresponding to **ResourceConfig** preceding the one corresponding to **AccessConfig**. If using the default values, include them explicitly as **conf/srm.conf** and **conf/access.conf** files.

11.2.2.2. Main Server Configuration

The main server configuration section of the configuration file sets up the main server, which responds to any requests that are not handled by a virtual host defined within a **<VirtualHost>** container. Values here also provide defaults for any **<VirtualHost>** containers defined.

The directives used in this section have changed little between Apache HTTP Server 1.3 and version 2.0. If the main server configuration is heavily customized, it may be easier to modify the existing configuration file to suit Apache HTTP Server 2.0. Users with only lightly customized main server sections should migrate their changes into the default 2.0 configuration.

11.2.2.2.1. UserDir Mapping

The **UserDir** directive is used to enable URLs such as **http://example.com/~bob/** to map to a subdirectory within the home directory of the user **bob**, such as **/home/bob/public_html/**. A side-effect of this feature allows a potential attacker to determine whether a given username is present on the system. For this reason, the default configuration for Apache HTTP Server 2.0 disables this directive.

To enable **UserDir** mapping, change the directive in **httpd.conf** from:

```
UserDir disable
```

to the following:

```
UserDir public_html
```

For more on this topic, refer to the following documentation on the Apache Software Foundation's website:

- *http://httpd.apache.org/docs-2.0/mod/mod_userdir.html#userdir*

11.2.2.2.2. Logging

The following logging directives have been removed:

- **AgentLog**
- **RefererLog**
- **RefererIgnore**

However, agent and referrer logs are still available using the **CustomLog** and **LogFormat** directives.

For more on this topic, refer to the following documentation on the Apache Software Foundation's website:

- *http://httpd.apache.org/docs-2.0/mod/mod_log_config.html#customlog*
- *http://httpd.apache.org/docs-2.0/mod/mod_log_config.html#logformat*

11.2.2.2.3. Directory Indexing

The deprecated **FancyIndexing** directive has now been removed. The same functionality is available through the **FancyIndexing***option* within the **IndexOptions** directive.

The **VersionSort** option to the **IndexOptions** directive causes files containing version numbers to be sorted in a more natural way. For example, **httpd-2.0.6.tar** appears before **httpd-2.0.36.tar** in a directory index page.

The defaults for the **ReadmeName** and **HeaderName** directives have changed from **README** and **HEADER** to **README.html** and **HEADER.html**.

For more on this topic, refer to the following documentation on the Apache Software Foundation's website:

- *http://httpd.apache.org/docs-2.0/mod/mod_autoindex.html#indexoptions*
- *http://httpd.apache.org/docs-2.0/mod/mod_autoindex.html#readmename*
- *http://httpd.apache.org/docs-2.0/mod/mod_autoindex.html#headername*

11.2.2.2.4. Content Negotiation

The **CacheNegotiatedDocs** directive now takes the argument on or off. Existing instances of **CacheNegotiatedDocs** should be replaced with **CacheNegotiatedDocs on**.

For more on this topic, refer to the following documentation on the Apache Software Foundation's website:

- *http://httpd.apache.org/docs-2.0/mod/mod_negotiation.html#cachenegotiateddocs*

11.2.2.2.5. Error Documents

To use a hard-coded message with the **ErrorDocument** directive, the message should be enclosed in a pair of double quotation marks **"**, rather than just preceded by a double quotation mark as required in Apache HTTP Server 1.3.

For example, the following is a sample Apache HTTP Server 1.3 directive:

```
ErrorDocument 404 "The document was not found
```

To migrate an **ErrorDocument** setting to Apache HTTP Server 2.0, use the following structure:

```
ErrorDocument 404 "The document was not found"
```

Note the trailing double quote in the previous **ErrorDocument** directive example.

For more on this topic, refer to the following documentation on the Apache Software Foundation's website:

- *http://httpd.apache.org/docs-2.0/mod/core.html#errordocument*

11.2.2.3. Virtual Host Configuration

The contents of all **<VirtualHost>** containers should be migrated in the same way as the main server section as described in *Section 11.2.2.2, “Main Server Configuration”*.

> **Important**
>
> Note that SSL/TLS virtual host configuration has been moved out of the main server configuration file and into **/etc/httpd/conf.d/ssl.conf**.

- *http://httpd.apache.org/docs-2.0/vhosts/*

11.2.2.4. Modules and Apache HTTP Server 2.0

In Apache HTTP Server 2.0, the module system has been changed to allow modules to be chained together or combined in new and interesting ways. *Common Gateway Interface* (*CGI*) scripts, for example, can generate server-parsed HTML documents which can then be processed by **mod_include**. This opens up a tremendous number of possibilities with regards to how modules can be combined to achieve a specific goal.

The way this works is that each request is served by exactly one *handler* module followed by zero or more *filter* modules.

Under Apache HTTP Server 1.3, for example, a Perl script would be handled in its entirety by the Perl module (**mod_perl**). Under Apache HTTP Server 2.0, the request is initially *handled* by the core module — which serves static files — and is then *filtered* by **mod_perl**.

Exactly how to use this, and all other new features of Apache HTTP Server 2.0, is beyond the scope of this document; however, the change has ramifications if the **PATH_INFO** directive is used for a document which is handled by a module that is now implemented as a filter, as each contains trailing path information after the true file name. The core module, which initially handles the request, does not by default understand **PATH_INFO** and returns 404 Not Found errors for requests that contain such information. As an alternative, use the **AcceptPathInfo** directive to coerce the core module into accepting requests with **PATH_INFO**.

The following is an example of this directive:

```
AcceptPathInfo on
```

For more on this topic, refer to the following documentation on the Apache Software Foundation's website:

- *http://httpd.apache.org/docs-2.0/mod/core.html#acceptpathinfo*
- *http://httpd.apache.org/docs-2.0/handler.html*
- *http://httpd.apache.org/docs-2.0/filter.html*

11.2.2.4.1. The suexec Module

In Apache HTTP Server 2.0, the **mod_suexec** module uses the **SuexecUserGroup** directive, rather than the **User** and **Group** directives, which is used for configuring virtual hosts. The **User** and **Group** directives can still be used in general, but are deprecated for configuring virtual hosts.

For example, the following is a sample Apache HTTP Server 1.3 directive:

```
<VirtualHost vhost.example.com:80> User someone Group somegroup </
VirtualHost>
```

To migrate this setting to Apache HTTP Server 2.0, use the following structure:

```
<VirtualHost vhost.example.com:80> SuexecUserGroup someone somegroup </
VirtualHost>
```

11.2.2.4.2. The mod_ssl Module

The configuration for **mod_ssl** has been moved from the **httpd.conf** file into the **/etc/httpd/conf.d/ssl.conf** file. For this file to be loaded, and for **mod_ssl** to work, the statement **Include conf.d/*.conf** must be in the **httpd.conf** file as described in *Section 11.2.2.1.3, “Dynamic Shared Object (DSO) Support”*.

ServerName directives in SSL virtual hosts must explicitly specify the port number.

For example, the following is a sample Apache HTTP Server 1.3 directive:

```
<VirtualHost _default_:443> # General setup for the virtual host ServerName
 ssl.example.name ... </VirtualHost>
```

To migrate this setting to Apache HTTP Server 2.0, use the following structure:

```
<VirtualHost _default_:443> # General setup for the virtual host ServerName
 ssl.host.name:443 ... </VirtualHost>
```

It is also important to note that both the **SSLLog** and **SSLLogLevel** directives have been removed. The **mod_ssl** module now obeys the **ErrorLog** and **LogLevel** directives. Refer to *ErrorLog* and *LogLevel* for more information about these directives.

For more on this topic, refer to the following documentation on the Apache Software Foundation's website:

- *http://httpd.apache.org/docs-2.0/mod/mod_ssl.html*
- *http://httpd.apache.org/docs-2.0/vhosts/*

11.2.2.4.3. The mod_proxy Module

Proxy access control statements are now placed inside a **<Proxy>** block rather than a **<Directory proxy:>**.

The caching functionality of the old **mod_proxy** has been split out into the following three modules:

- **mod_cache**
- **mod_disk_cache**
- **mod_mem_cache**

These generally use directives similar to the older versions of the **mod_proxy** module, but it is advisable to verify each directive before migrating any cache settings.

For more on this topic, refer to the following documentation on the Apache Software Foundation's website:

- *http://httpd.apache.org/docs-2.0/mod/mod_proxy.html*

11.2.2.4.4. The mod_include Module

The **mod_include** module is now implemented as a filter and is therefore enabled differently. Refer to *Section 11.2.2.4, “Modules and Apache HTTP Server 2.0”* for more about filters.

For example, the following is a sample Apache HTTP Server 1.3 directive:

```
AddType text/html .shtml AddHandler server-parsed .shtml
```

To migrate this setting to Apache HTTP Server 2.0, use the following structure:

```
AddType text/html .shtml AddOutputFilter INCLUDES .shtml
```

Note that the **Options +Includes** directive is still required for the **<Directory>** container or in a **.htaccess** file.

For more on this topic, refer to the following documentation on the Apache Software Foundation's website:

- *http://httpd.apache.org/docs-2.0/mod/mod_include.html*

11.2.2.4.5. The mod_auth_dbm and mod_auth_db Modules

Apache HTTP Server 1.3 supported two authentication modules, **mod_auth_db** and **mod_auth_dbm**, which used Berkeley Databases and DBM databases respectively. These modules have been combined into a single module named **mod_auth_dbm** in Apache HTTP Server 2.0, which can access several different database formats. To migrate from **mod_auth_db**, configuration files should be adjusted by replacing **AuthDBUserFile** and **AuthDBGroupFile** with the **mod_auth_dbm** equivalents, **AuthDBMUserFile** and **AuthDBMGroupFile**. Also, the directive **AuthDBMType DB** must be added to indicate the type of database file in use.

The following example shows a sample **mod_auth_db** configuration for Apache HTTP Server 1.3:

```
<Location /private/> AuthType Basic AuthName "My Private Files"
 AuthDBUserFile /var/www/authdb require valid-user </Location>
```

To migrate this setting to version 2.0 of Apache HTTP Server, use the following structure:

```
<Location /private/> AuthType Basic AuthName "My Private
 Files" AuthDBMUserFile /var/www/authdb AuthDBMType DB require valid-user
 </Location>
```

Note that the **AuthDBMUserFile** directive can also be used in **.htaccess** files.

The **dbmmanage** Perl script, used to manipulate username and password databases, has been replaced by **htdbm** in Apache HTTP Server 2.0. The **htdbm** program offers equivalent functionality and, like **mod_auth_dbm**, can operate a variety of database formats; the `-T` option can be used on the command line to specify the format to use.

*Table 11.1, “Migrating from **dbmmanage** to **htdbm**”* shows how to migrate from a DBM-format database to **htdbm** format using **dbmmanage**.

Action	dbmmanage command (1.3)	Equivalent htdbm command (2.0)
Add user to database (using given password)	**dbmmanage authdb add username password**	**htdbm -b -TDB authdb username password**

Action	dbmmanage command (1.3)	Equivalent htdbm command (2.0)
Add user to database (prompts for password)	**`dbmmanage authdb adduser username`**	**`htdbm -TDB authdb username`**
Remove user from database	**`dbmmanage authdb delete username`**	**`htdbm -x -TDB authdb username`**
List users in database	**`dbmmanage authdb view`**	**`htdbm -l -TDB authdb`**
Verify a password	**`dbmmanage authdb check username`**	**`htdbm -v -TDB authdb username`**

Table 11.1. Migrating from **dbmmanage** to **htdbm**

The `-m` and `-s` options work with both **dbmmanage** and **htdbm**, enabling the use of the MD5 or SHA1 algorithms for hashing passwords, respectively.

When creating a new database with **htdbm**, the **-c** option must be used.

For more on this topic, refer to the following documentation on the Apache Software Foundation's website:

- *http://httpd.apache.org/docs-2.0/mod/mod_auth_dbm.html*

11.2.2.4.6. The mod_perl Module

The configuration for **mod_perl** has been moved from **httpd.conf** into the file **/etc/httpd/conf.d/perl.conf**. For this file to be loaded, and hence for **mod_perl** to work, the statement **Include conf.d/*.conf** must be included in **httpd.conf** as described in *Section 11.2.2.1.3, “Dynamic Shared Object (DSO) Support”*.

Occurrences of **Apache::** in **httpd.conf** must be replaced with **ModPerl::**. Additionally, the manner in which handlers are registered has been changed.

This is a sample Apache HTTP Server 1.3 **mod_perl** configuration:

```
<Directory /var/www/perl> SetHandler perl-script PerlHandler
 Apache::Registry Options +ExecCGI </Directory>
```

This is the equivalent **mod_perl** for Apache HTTP Server 2.0:

```
<Directory /var/www/perl> SetHandler perl-script PerlResponseHandler
 ModPerl::Registry Options +ExecCGI </Directory>
```

Most modules for **mod_perl** 1.x should work without modification with **mod_perl** 2.x. XS modules require recompilation and may require minor Makefile modifications.

11.2.2.4.7. The mod_python Module

Configuration for **mod_python** has moved from **httpd.conf** to the **/etc/httpd/conf.d/python.conf** file. For this file to be loaded, and hence for **mod_python** to work, the statement

Include conf.d/*.conf must be in **httpd.conf** as described in *Section 11.2.2.1.3, “Dynamic Shared Object (DSO) Support”*.

11.2.2.4.8. PHP

The configuration for PHP has been moved from **httpd.conf** into the file **/etc/httpd/conf.d/php.conf**. For this file to be loaded, the statement **Include conf.d/*.conf** must be in **httpd.conf** as described in *Section 11.2.2.1.3, “Dynamic Shared Object (DSO) Support”*.

> **Note**
>
> Any PHP configuration directives used in Apache HTTP Server 1.3 are now fully compatible, when migrating to Apache HTTP Server 2.0 on Fedora 12.

In PHP version 4.2.0 and later the default set of predefined variables which are available in the global scope has changed. Individual input and server variables are, by default, no longer placed directly into the global scope. This change may cause scripts to break. Revert to the old behavior by setting **register_globals** to **On** in the file **/etc/php.ini**.

For more on this topic, refer to the following URL for details concerning the global scope changes:

- *http://www.php.net/release_4_1_0.php*

11.2.2.4.9. The mod_authz_ldap Module

Fedora ships with the **mod_authz_ldap** module for the Apache HTTP Server. This module uses the short form of the distinguished name for a subject and the issuer of the client SSL certificate to determine the distinguished name of the user within an LDAP directory. It is also capable of authorizing users based on attributes of that user's LDAP directory entry, determining access to assets based on the user and group privileges of the asset, and denying access for users with expired passwords. The **mod_ssl** module is required when using the **mod_authz_ldap** module.

> **Important**
>
> The **mod_authz_ldap** module does not authenticate a user to an LDAP directory using an encrypted password hash. This functionality is provided by the experimental **mod_auth_ldap** module. Refer to the **mod_auth_ldap** module documentation online at *http://httpd.apache.org/docs-2.0/mod/mod_auth_ldap.html* for details on the status of this module.

The **/etc/httpd/conf.d/authz_ldap.conf** file configures the **mod_authz_ldap** module.

Refer to **/usr/share/doc/mod_authz_ldap-<*version*>/index.html** (replacing <*version*> with the version number of the package) or *http://authzldap.othello.ch/* for more information on configuring the **mod_authz_ldap** third party module.

11.3. Starting and Stopping httpd

After installing the **httpd** package, review the Apache HTTP Server's documentation available online at *http://httpd.apache.org/docs/2.2/*.

The **`httpd`** RPM installs the **`/etc/init.d/httpd`** script, which can be accessed using the **`/sbin/service`** command.

Starting **`httpd`** using the **`apachectl`** control script sets the environmental variables in **`/etc/sysconfig/httpd`** and starts **`httpd`**. You can also set the environment variables using the init script.

To start the server using the **`apachectl`** control script as root type:

```
apachectl start
```

You can also start **`httpd`** using **`/sbin/service httpd start`**. This starts **`httpd`** but does not set the environment variables. If you are using the default **`Listen`** directive in **`httpd.conf`**, which is port 80, you will need to have root privileges to start the apache server.

To stop the server, as root type:

```
apachectl stop
```

You can also stop **`httpd`** using **`/sbin/service httpd stop`**. The `restart` option is a shorthand way of stopping and then starting the Apache HTTP Server.

You can restart the server as root by typing:

```
apachectl restart
or:
/sbin/service httpd restart
```

Apache will display a message on the console or in the **`ErrorLog`** if it encounters an error while starting.

By default, the **`httpd`** service does *not* start automatically at boot time. If you would wish to have Apache startup at boot time, you will need to add a call to **`apachectl`** in your startup files within the **`rc.N`** directory. A typical file used is **`rc.local`**. As this starts Apache as root, it is recommended to properly configure your security and authentication before adding this call.

You can also configure the **`httpd`** service to start up at boot time, using an initscript utility, such as **`/sbin/chkconfig`**, **/usr/sbin/ntsysv**, or the **Services Configuration Tool** program.

You can also display the status of your httpd server by typing:

```
apachectl status
```

The status module **`mod_status`** however needs to be enabled in your **`httpd.conf`** configuration file for this to work. For more details on **`mod_status`** can be found on *http://httpd.apache.org/docs/2.2/mod/mod_status.html*.

Note

If running the Apache HTTP Server as a secure server, the secure server's password is required after the machine boots when using an encrypted private SSL key.

You can find more information on *http://httpd.apache.org/docs/2.2/ssl*

11.4. Apache HTTP Server Configuration

The **HTTP Configuration Tool** allows you to configure the **`/etc/httpd/conf/httpd.conf`** configuration file for the Apache HTTP Server. It does not use the old **`srm.conf`** or **`access.conf`** configuration files; leave them empty. Through the graphical interface, you can configure directives such as virtual hosts, logging attributes, and maximum number of connections. To start the HTTD Configuration Tool, click on **`System > Administration > Server Settings > HTTP`**.

Only modules provided with Fedora can be configured with the **HTTP Configuration Tool**. If additional modules are installed, they can not be configured using this tool.

Caution

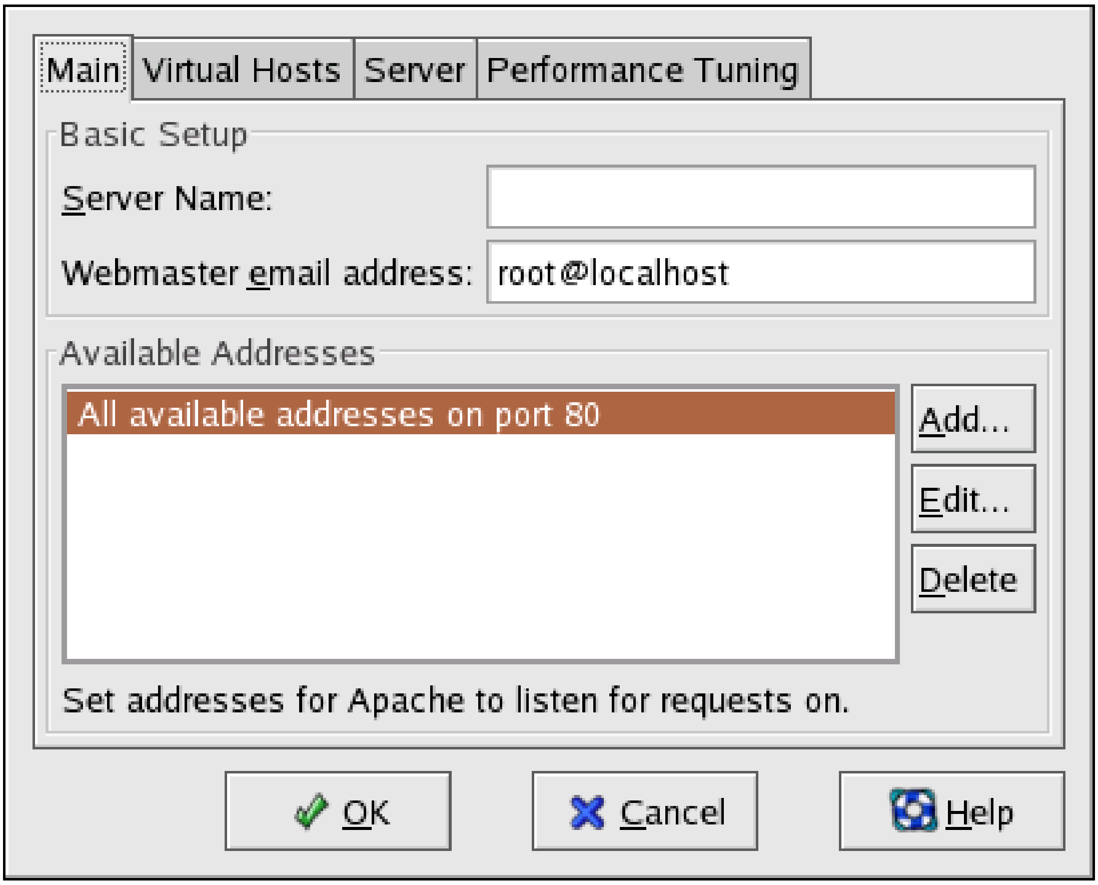
Do not edit the **`/etc/httpd/conf/httpd.conf`** configuration file by hand if you wish to use this tool. The **HTTP Configuration Tool** generates this file after you save your changes and exit the program. If you want to add additional modules or configuration options that are not available in **HTTP Configuration Tool**, you cannot use this tool.

The general steps for configuring the Apache HTTP Server using the **HTTP Configuration Tool** are as follows:

1. Configure the basic settings under the **Main** tab.
2. Click on the **Virtual Hosts** tab and configure the default settings.
3. Under the **Virtual Hosts** tab, configure the Default Virtual Host.
4. To serve more than one URL or virtual host, add any additional virtual hosts.
5. Configure the server settings under the **Server** tab.
6. Configure the connections settings under the **Performance Tuning** tab.
7. Copy all necessary files to the **`DocumentRoot`** and **`cgi-bin`** directories.
8. Exit the application and select to save your settings.

11.4.1. Basic Settings

Use the **Main** tab to configure the basic server settings.



Server Name:

Webmaster email address: root@localhost

All available addresses on port 80

Set addresses for Apache to listen for requests on.

Figure 11.1. Basic Settings

Enter a fully qualified domain name that you have the right to use in the **Server Name** text area. This option corresponds to the ***`ServerName`***[2] directive in **`httpd.conf`**. The **`ServerName`** directive sets the hostname of the Web server. It is used when creating redirection URLs. If you do not define a server name, the Web server attempts to resolve it from the IP address of the system. The server name does not have to be the domain name resolved from the IP address of the server. For example, you might set the server name to www.example.com while the server's real DNS name is foo.example.com.

Enter the email address of the person who maintains the Web server in the **Webmaster email address** text area. This option corresponds to the ***`ServerAdmin`***[3] directive in **`httpd.conf`**. If you configure the server's error pages to contain an email address, this email address is used so that users can report a problem to the server's administrator. The default value is root@localhost.

Use the **Available Addresses** area to define the ports on which the server accepts incoming requests. This option corresponds to the ***`Listen`***[4] directive in **`httpd.conf`**. By default, Red Hat configures the Apache HTTP Server to listen to port 80 for non-secure Web communications.

Click the **Add** button to define additional ports on which to accept requests. A window as shown in *Figure 11.2, “Available Addresses”* appears. Either choose the **Listen to all addresses** option to listen to all IP addresses on the defined port or specify a particular IP address over which the server accepts connections in the **Address** field. Only specify one IP address per port number. To specify

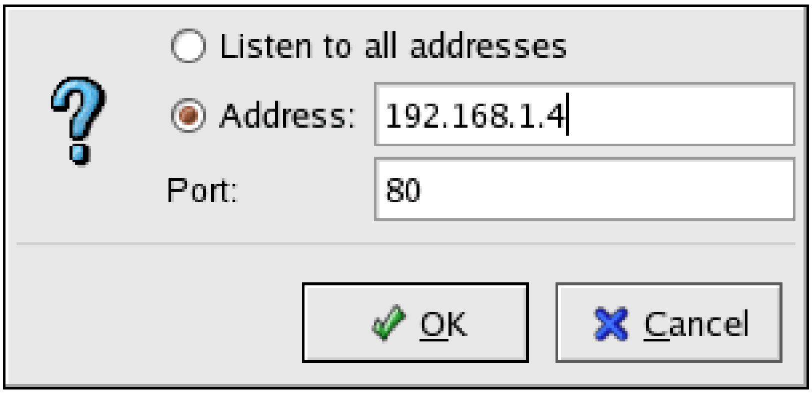
[2] http://httpd.apache.org/docs/2.2/mod/core.html#servername
[3] http://httpd.apache.org/docs/2.2/mod/core.html#serveradmin
[4] http://httpd.apache.org/docs/2.2/mod/mpm_common.html#listen

more than one IP address with the same port number, create an entry for each IP address. If at all possible, use an IP address instead of a domain name to prevent a DNS lookup failure. Refer to *http://httpd.apache.org/docs/2.2/dns-caveats.html* for more information about *Issues Regarding DNS and Apache*.

Entering an asterisk (*) in the **Address** field is the same as choosing **Listen to all addresses**. Clicking the **Edit** button in the **Available Addresses** frame shows the same window as the **Add** button except with the fields populated for the selected entry. To delete an entry, select it and click the **Delete** button.

Tip

If you set the server to listen to a port under 1024, you must be root to start it. For port 1024 and above, **`httpd`** can be started as a regular user.



○ Listen to all addresses

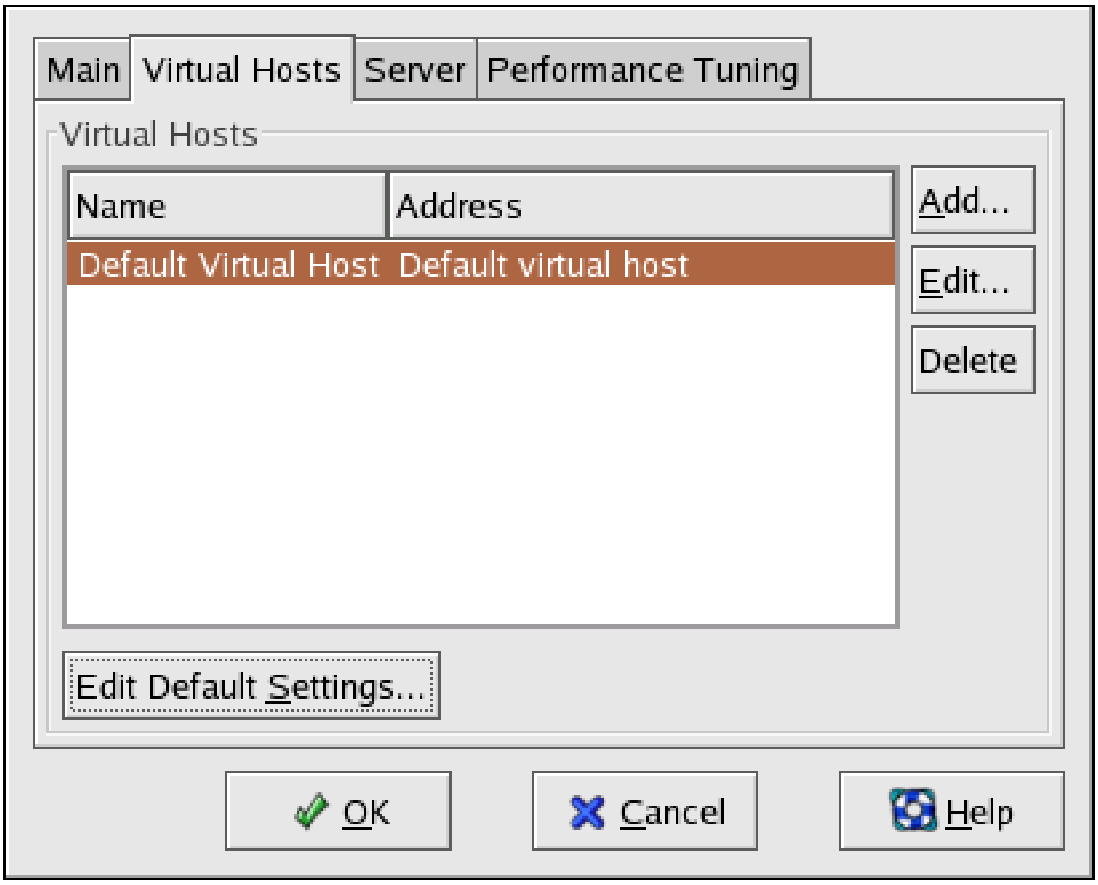
● Address: 192.168.1.4

Port: 80

Figure 11.2. Available Addresses

11.4.2. Default Settings

After defining the **Server Name**, **Webmaster email address**, and **Available Addresses**, click the **Virtual Hosts** tab. The figure below illustrates the **Virtual Hosts** tab.

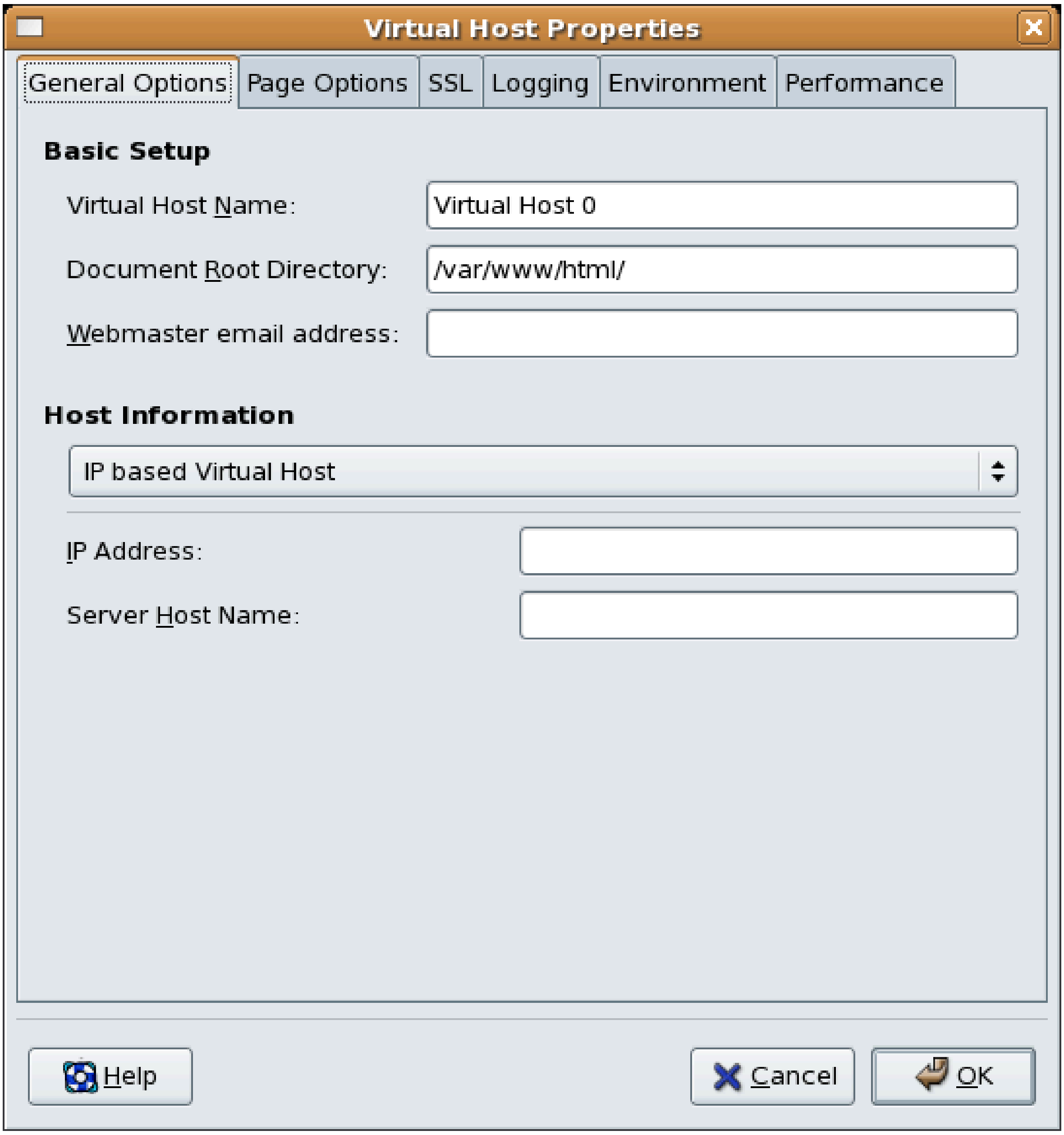


Name	Address
Default Virtual Host	Default virtual host

Figure 11.3. Virtual Hosts Tab

Clicking on **Edit** will display the **Virtual Host Properties** window from which you can set your preferred settings. To add new settings, click on the **Add** button which will also display the **Virtual Host Properties** window. Clicking on the **Edit Default Settings** button, displays the **Virtual Host Properties** window without the **General Options** tab.

In the **General Options** tab, you can change the hostname, the document root directory and also set the webmaster's email address. In the Host information, you can set the Virtual Host's IP Address and Host Name. The figure below illustrates the **General Options** tab.



Basic Setup

Virtual Host Name: Virtual Host 0

Document Root Directory: /var/www/html/

Webmaster email address:

Host Information

IP based Virtual Host

IP Address:

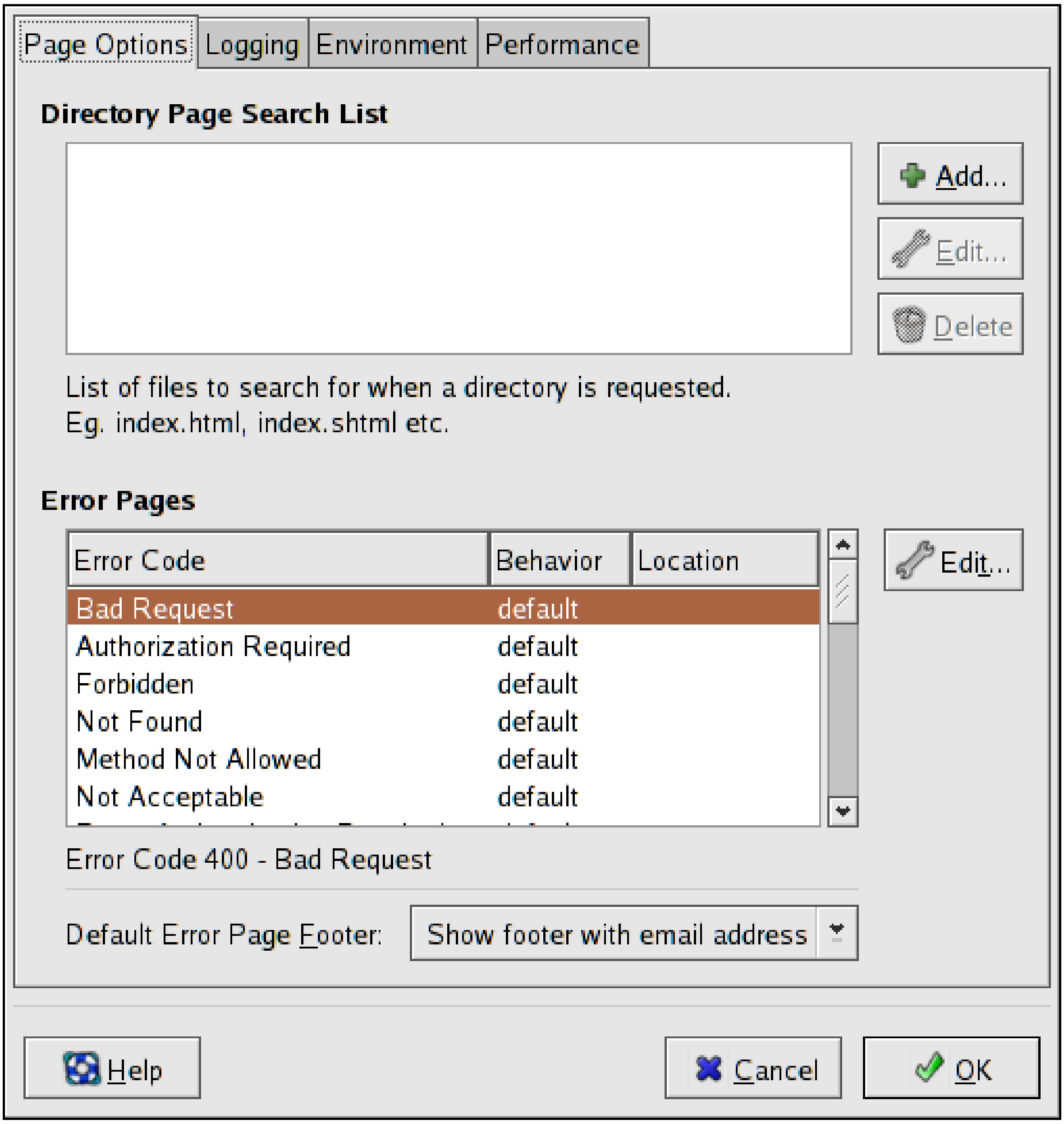
Server Host Name:

Figure 11.4. General Options

If you add a virtual host, the settings you configure for the virtual host take precedence for that virtual host. For a directive not defined within the virtual host settings, the default value is used.

11.4.2.1. Site Configuration

The figure below illustrates the **Page Options**tab from which you can configure the **Directory Page Search List** and **Error Pages**. If you are unsure of these settings, do not modify them.



Directory Page Search List

List of files to search for when a directory is requested.
Eg. index.html, index.shtml etc.

Error Pages

Error Code	Behavior	Location
Bad Request	default	
Authorization Required	default	
Forbidden	default	
Not Found	default	
Method Not Allowed	default	
Not Acceptable	default	

Error Code 400 - Bad Request

Default Error Page Footer: Show footer with email address

Figure 11.5. Site Configuration

The entries listed in the **Directory Page Search List** define the ***`DirectoryIndex`***[5] directive. The **`DirectoryIndex`** is the default page served by the server when a user requests an index of a directory by specifying a forward slash (/) at the end of the directory name.

For example, when a user requests the page **`http://www.example.com/this_directory/`**, they are going to get either the **`DirectoryIndex`** page, if it exists, or a server-generated directory list. The server tries to find one of the files listed in the **`DirectoryIndex`** directive and returns the first one it finds.
If it does not find any of these files and if **`Options Indexes`** is set for that directory, the server generates and returns a list, in HTML format, of the subdirectories and files in the directory.

[5] http://httpd.apache.org/docs/2.2/mod/mod_dir.html#directoryindex

Use the **Error Code** section to configure Apache HTTP Server to redirect the client to a local or external URL in the event of a problem or error. This option corresponds to the ***`ErrorDocument`***[6] directive. If a problem or error occurs when a client tries to connect to the Apache HTTP Server, the default action is to display the short error message shown in the **Error Code** column. To override this default configuration, select the error code and click the **Edit** button. Choose **Default** to display the default short error message. Choose **URL** to redirect the client to an external URL and enter a complete URL, including the **`http://`**, in the **Location** field. Choose **File** to redirect the client to an internal URL and enter a file location under the document root for the Web server. The location must begin the a slash (/) and be relative to the Document Root.

For example, to redirect a 404 Not Found error code to a webpage that you created in a file called **`404.html`**, copy **`404.html`** to ***`DocumentRoot`*****`/../error/404.html`**. In this case, *`DocumentRoot`* is the Document Root directory that you have defined (the default is **`/var/www/html/`**). If the Document Root is left as the default location, the file should be copied to **`/var/www/error/404.html`**. Then, choose **File** as the Behavior for **404 - Not Found** error code and enter **`/error/404.html`** as the **Location**.

From the **Default Error Page Footer** menu, you can choose one of the following options:

- **Show footer with email address** — Display the default footer at the bottom of all error pages along with the email address of the website maintainer specified by the ***`ServerAdmin`***[7] directive.
- **Show footer** — Display just the default footer at the bottom of error pages.
- **No footer** — Do not display a footer at the bottom of error pages.

11.4.2.2. SSL Support

The **`mod_ssl`** enables encryption of the HTTP protocol over SSL. SSL (Secure Sockets Layer) protocol is used for communication and encryption over TCP/IP networks. The SSL tab enables you to configure SSL for your server. To configure SSL you need to provide the path to your:

- Certificate file - equivalent to using the **`SSLCertificateFile`** directive which points the path to the PEM (Privacy Enhanced Mail)-encoded server certificate file.
- Key file - equivalent to using the **`SSLCertificateKeyFile`** directive which points the path to the PEM-encoded server private key file.
- Certificate chain file - equivalent to using the **`SSLCertificateChainFile`** directive which points the path to the certificate file containing all the server's chain of certificates.
- Certificate authority file - is an encrypted file used to confirm the authenticity or identity of parties communicating with the server.

You can find out more about configuration directives for SSL on *http://httpd.apache.org/docs/2.2/mod/directives.html#S*[8]. You also need to determine which SSL options to enable. These are equivalent to using the **`SSLOptions`** with the following options:

- FakeBasicAuth - enables standard authentication methods used by Apache. This means that the Client X509 certificate's Subject Distinguished Name (DN) is translated into a basic HTTP username.

[6] http://httpd.apache.org/docs/2.2/mod/core.html#errordocument
[8] http://httpd.apache.org/docs/2.2/mod/directives.html#S

- ExportCertData - creates CGI environment variables in **`SSL_SERVER_CERT`**, **`SSL_CLIENT_CERT`** and **`SSL_CLIENT_CERT_CHAIN_n`** where n is a number 0,1,2,3,4... These files are used for more certificate checks by CGI scripts.

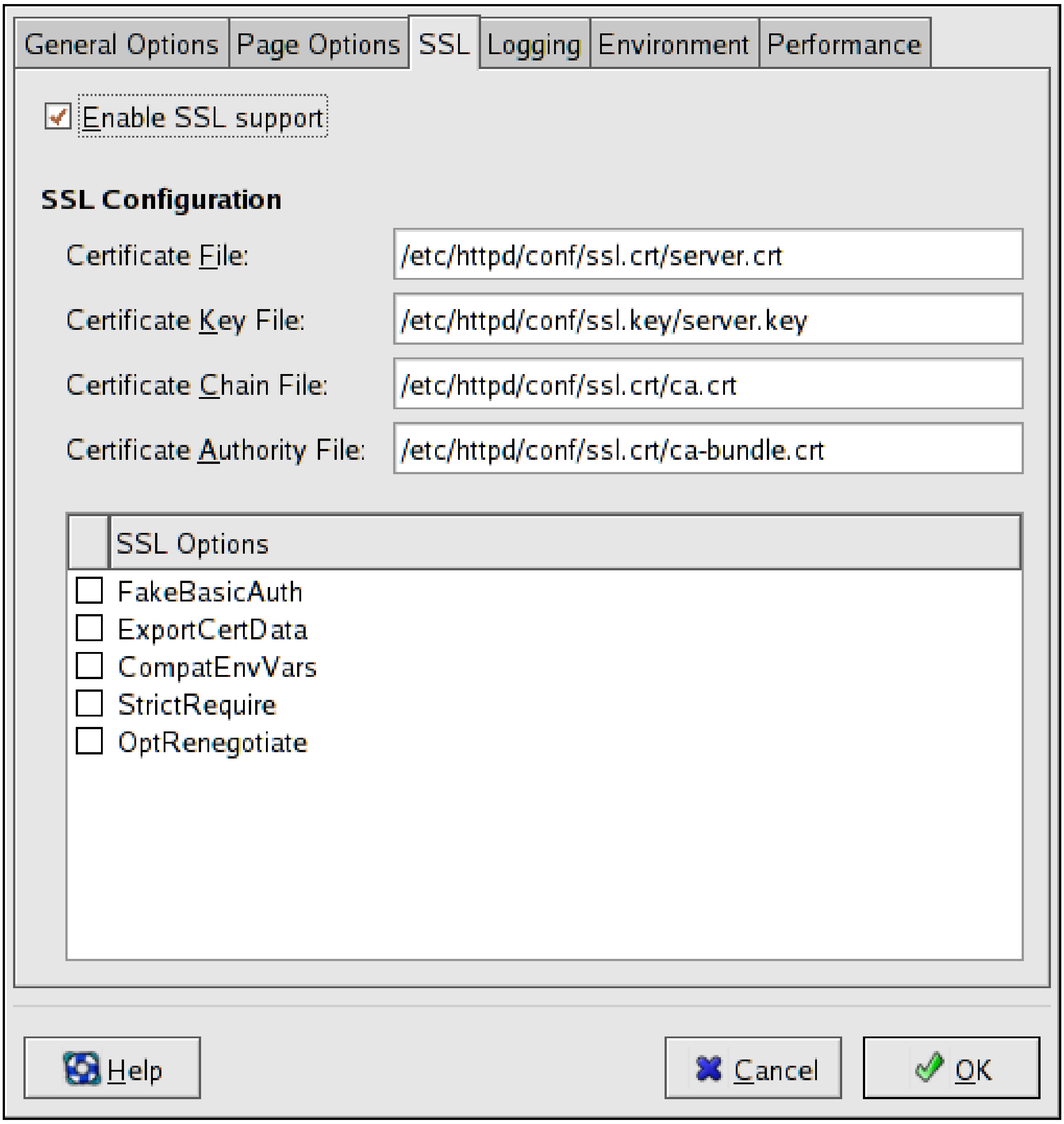
- CompatEnvVars - enables backward compatibility for Apache SSL by adding CGI environment variables.

- StrictRequire - enables strict access which forces denial of access whenever the **`SSLRequireSSL`** and **`SSLRequire`** directives indicate access is forbiden.

- OptRenegotiate - enables avoidance of unnecessary handshakes by **`mod_ssl`** which also performs safe parameter checks. It is recommended to enable OptRenegotiate on a per directory basis.

More information on the above SSL options can be found on *http://httpd.apache.org/docs/2.2/mod/mod_ssl.html#ssloptions*[9]. The figure below illustrates the SSL tab and the options discussed above.

[9] http://httpd.apache.org/docs/2.2/mod/mod_ssl.html#ssloptions



☑ Enable SSL support

SSL Configuration

Certificate File: /etc/httpd/conf/ssl.crt/server.crt

Certificate Key File: /etc/httpd/conf/ssl.key/server.key

Certificate Chain File: /etc/httpd/conf/ssl.crt/ca.crt

Certificate Authority File: /etc/httpd/conf/ssl.crt/ca-bundle.crt

	SSL Options
□	FakeBasicAuth
□	ExportCertData
□	CompatEnvVars
□	StrictRequire
□	OptRenegotiate

Figure 11.6. SSL

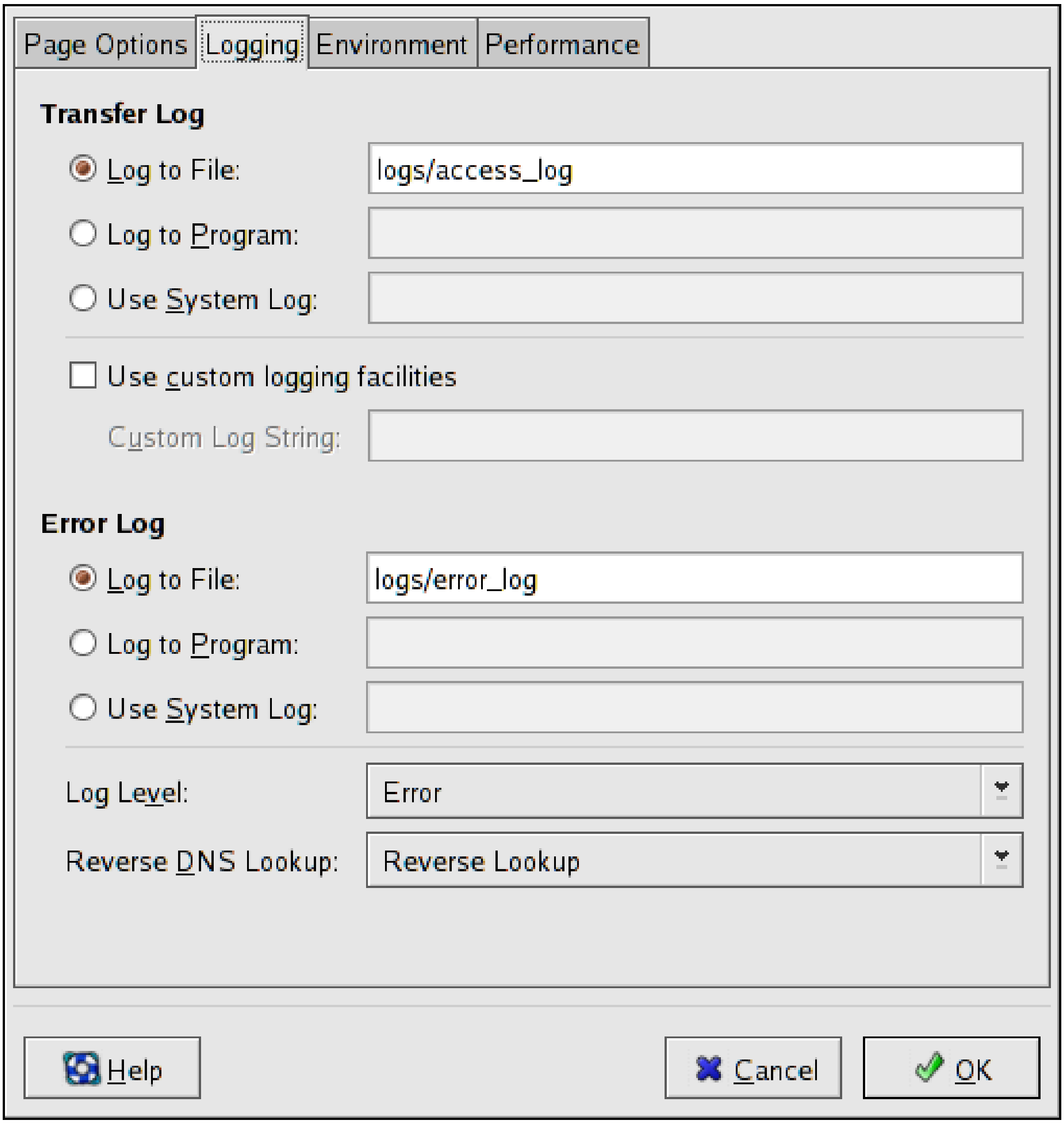
11.4.2.3. Logging

Use the **Logging** tab to configure options for specific transfer and error logs.

By default, the server writes the transfer log to the **`/var/log/httpd/access_log`** file and the error log to the **`/var/log/httpd/error_log`** file.

The transfer log contains a list of all attempts to access the Web server. It records the IP address of the client that is attempting to connect, the date and time of the attempt, and the file on the Web server that it is trying to retrieve. Enter the name of the path and file in which to store this information. If the path and file name do not start with a slash (/), the path is relative to the server root directory as configured. This option corresponds to the ***`TransferLog`***[10] directive.

[10] http://httpd.apache.org/docs/2.2/mod/mod_log_config.html#transferlog



Transfer Log

● Log to File: logs/access_log

○ Log to Program:

○ Use System Log:

□ Use custom logging facilities

Custom Log String:

Error Log

● Log to File: logs/error_log

○ Log to Program:

○ Use System Log:

Log Level: Error

Reverse DNS Lookup: Reverse Lookup

Figure 11.7. Logging

You can configure a custom log format by checking **Use custom logging facilities** and entering a custom log string in the **Custom Log String** field. This configures the `LogFormat`[11] directive. Refer to *http://httpd.apache.org/docs/2.2/mod/mod_log_config.html#logformat*[12] for details on the format of this directive.

The error log contains a list of any server errors that occur. Enter the name of the path and file in which to store this information. If the path and file name do not start with a slash (/), the path is relative to the server root directory as configured. This option corresponds to the `ErrorLog`[13] directive.

[11] http://httpd.apache.org/docs/2.2/mod/mod_log_config.html#logformat
[12] http://httpd.apache.org/docs-2.0/mod/mod_log_config.html#formats
[13] http://httpd.apache.org/docs/2.2/mod/core.html#errorlog

Use the **Log Level** menu to set the verbosity of the error messages in the error logs. It can be set (from least verbose to most verbose) to emerg, alert, crit, error, warn, notice, info or debug. This option corresponds to the `LogLevel`[14] directive.

The value chosen with the **Reverse DNS Lookup** menu defines the `HostnameLookups`[15] directive. Choosing **No Reverse Lookup** sets the value to off. Choosing **Reverse Lookup** sets the value to on. Choosing **Double Reverse Lookup** sets the value to double.

If you choose **Reverse Lookup**, your server automatically resolves the IP address for each connection which requests a document from your Web server. Resolving the IP address means that your server makes one or more connections to the DNS in order to find out the hostname that corresponds to a particular IP address.

If you choose **Double Reverse Lookup**, your server performs a double-reverse DNS. In other words, after a reverse lookup is performed, a forward lookup is performed on the result. At least one of the IP addresses in the forward lookup must match the address from the first reverse lookup.

Generally, you should leave this option set to **No Reverse Lookup**, because the DNS requests add a load to your server and may slow it down. If your server is busy, the effects of trying to perform these reverse lookups or double reverse lookups may be quite noticeable.

Reverse lookups and double reverse lookups are also an issue for the Internet as a whole. Each individual connection made to look up each hostname adds up. Therefore, for your own Web server's benefit, as well as for the Internet's benefit, you should leave this option set to **No Reverse Lookup**.

11.4.2.4. Environment Variables

Use the **Environment** tab to configure options for specific variables to set, pass, or unset for CGI scripts.

Sometimes it is necessary to modify environment variables for CGI scripts or server-side include (SSI) pages. The Apache HTTP Server can use the **`mod_env`** module to configure the environment variables which are passed to CGI scripts and SSI pages. Use the **Environment Variables** page to configure the directives for this module.

Use the **Set for CGI Scripts** section to set an environment variable that is passed to CGI scripts and SSI pages. For example, to set the environment variable **`MAXNUM`** to **`50`**, click the **Add** button inside the **Set for CGI Script** section, as shown in *Figure 11.8, “Environment Variables”*, and type **`MAXNUM`** in the **Environment Variable** text field and **`50`** in the **Value to set** text field. Click **OK** to add it to the list. The **Set for CGI Scripts** section configures the `SetEnv`[16] directive.

Use the **Pass to CGI Scripts** section to pass the value of an environment variable when the server is first started to CGI scripts. To see this environment variable, type the command **env** at a shell prompt. Click the **Add** button inside the **Pass to CGI Scripts** section and enter the name of the environment variable in the resulting dialog box. Click **OK** to add it to the list. The **Pass to CGI Scripts** section configures the `PassEnv`[17] directive.

[14] http://httpd.apache.org/docs/2.2/mod/core.html#loglevel
[15] http://httpd.apache.org/docs/2.2/mod/core.html#hostnamelookups
[16] http://httpd.apache.org/docs/2.2/mod/mod_env.html#setenv
[17] http://httpd.apache.org/docs/2.2/mod/mod_env.html#passenv

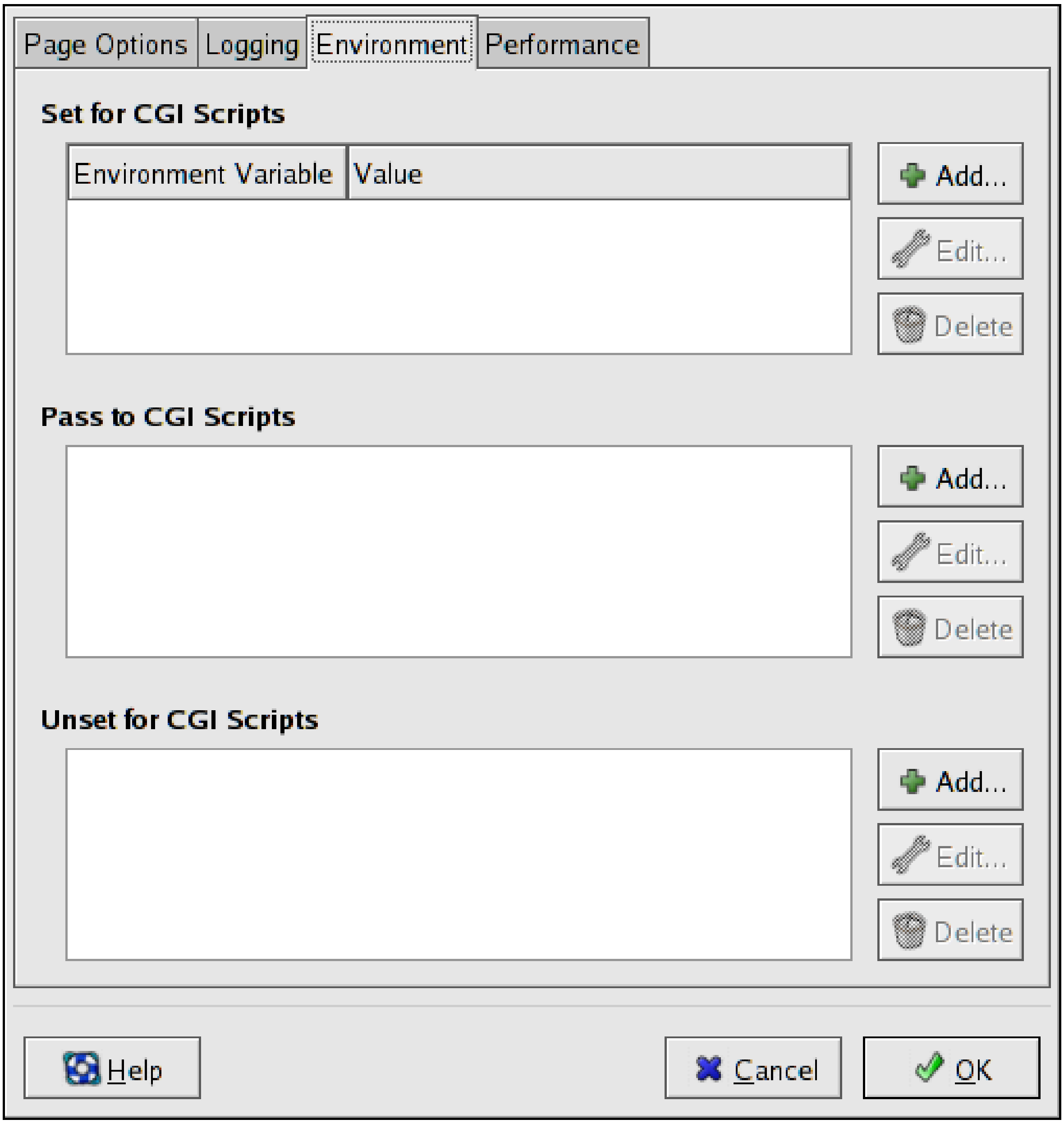


Figure 11.8. Environment Variables

To remove an environment variable so that the value is not passed to CGI scripts and SSI pages, use the **Unset for CGI Scripts** section. Click **Add** in the **Unset for CGI Scripts** section, and enter the name of the environment variable to unset. Click **OK** to add it to the list. This corresponds to the `UnsetEnv`[18] directive.

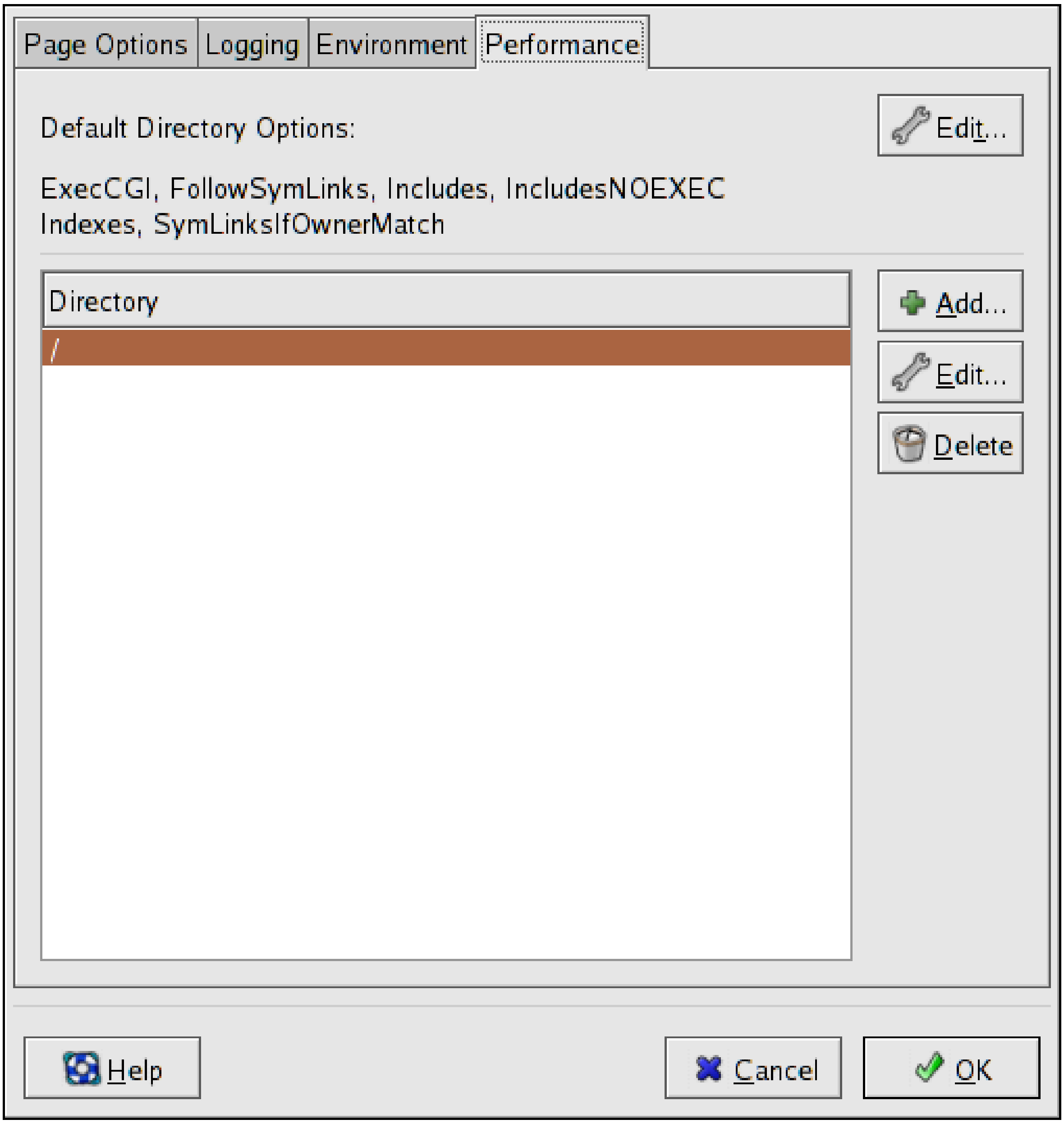
To edit any of these environment values, select it from the list and click the corresponding **Edit** button. To delete any entry from the list, select it and click the corresponding **Delete** button.

To learn more about environment variables in the Apache HTTP Server, refer to the following: *http://httpd.apache.org/docs/2.2/env.html*

[18] http://httpd.apache.org/docs/2.2/mod/mod_env.html#unsetenv

11.4.2.5. Directories

Use the **Directories** page in the **Performance** tab to configure options for specific directories. This corresponds to the `<Directory>`[19] directive.



Default Directory Options:

ExecCGI, FollowSymLinks, Includes, IncludesNOEXEC
Indexes, SymLinksIfOwnerMatch

Directory
/

Figure 11.9. Directories

Click the **Edit** button in the top right-hand corner to configure the **Default Directory Options** for all directories that are not specified in the **Directory** list below it. The options that you choose are listed as the `Options`[20] directive within the `<Directory>`[21] directive. You can configure the following options:

- **ExecCGI** — Allow execution of CGI scripts. CGI scripts are not executed if this option is not chosen.

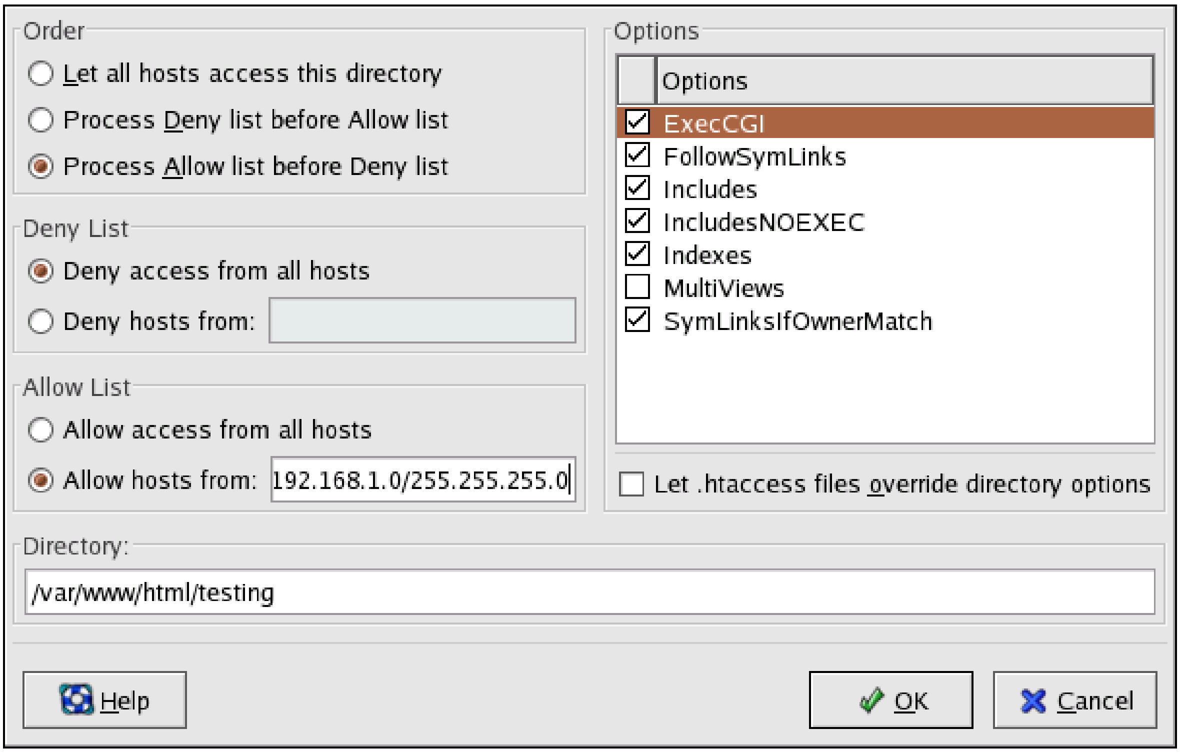
[19] http://httpd.apache.org/docs/2.2/mod/core.html#directory
[20] http://httpd.apache.org/docs/2.2/mod/core.html#options
[21] http://httpd.apache.org/docs/2.2/mod/core.html#directory

- **FollowSymLinks** — Allow symbolic links to be followed.
- **Includes** — Allow server-side includes.
- **IncludesNOEXEC** — Allow server-side includes, but disable the **`#exec`** and **`#include`** commands in CGI scripts.
- **Indexes** — Display a formatted list of the directory's contents, if no **`DirectoryIndex`** (such as **`index.html`**) exists in the requested directory.
- **Multiview** — Support content-negotiated multiviews; this option is disabled by default.
- **SymLinksIfOwnerMatch** — Only follow symbolic links if the target file or directory has the same owner as the link.

To specify options for specific directories, click the **Add** button beside the **Directory** list box. A window as shown in *Figure 11.10, “Directory Settings”* appears. Enter the directory to configure in the **Directory** text field at the bottom of the window. Select the options in the right-hand list and configure the ***`Order`***[22] directive with the left-hand side options. The **`Order`** directive controls the order in which allow and deny directives are evaluated. In the **Allow hosts from** and **Deny hosts from** text field, you can specify one of the following:

- Allow all hosts — Type **`all`** to allow access to all hosts.
- Partial domain name — Allow all hosts whose names match or end with the specified string.
- Full IP address — Allow access to a specific IP address.
- A subnet — Such as **`192.168.1.0/255.255.255.0`**
- A network CIDR specification — such as **`10.3.0.0/16`**

[22] http://httpd.apache.org/docs-2.0/mod/mod_access.html#order



Order

○ Let all hosts access this directory

○ Process Deny list before Allow list

● Process Allow list before Deny list

Deny List

● Deny access from all hosts

○ Deny hosts from:

Allow List

○ Allow access from all hosts

● Allow hosts from: 192.168.1.0/255.255.255.0

Options

	Options
☑	ExecCGI
☑	FollowSymLinks
☑	Includes
☑	IncludesNOEXEC
☑	Indexes
☐	MultiViews
☑	SymLinksIfOwnerMatch

☐ Let .htaccess files override directory options

Directory: /var/www/html/testing

Figure 11.10. Directory Settings

If you check the **Let .htaccess files override directory options**, the configuration directives in the **`.htaccess`** file take precedence.

11.5. Configuration Directives in `httpd.conf`

The Apache HTTP Server configuration file is **`/etc/httpd/conf/httpd.conf`**. The **`httpd.conf`** file is well-commented and mostly self-explanatory. The default configuration works for most situations; however, it is a good idea to become familiar some of the more important configuration options.

Warning

With the release of Apache HTTP Server 2.2, many configuration options have changed. If migrating from version 1.3 to 2.2, please firstly read *Section 11.2.2, “Migrating Apache HTTP Server 1.3 Configuration Files to 2.0”*.

11.5.1. General Configuration Tips

If configuring the Apache HTTP Server, edit **`/etc/httpd/conf/httpd.conf`** and then either reload, restart, or stop and start the **`httpd`** process as outlined in *Section 11.3, “Starting and Stopping `httpd`”*.

Before editing **`httpd.conf`**, make a copy the original file. Creating a backup makes it easier to recover from mistakes made while editing the configuration file.

If a mistake is made and the Web server does not work correctly, first review recently edited passages in **`httpd.conf`** to verify there are no typos.

Next look in the Web server's error log, **`/var/log/httpd/error_log`**. The error log may not be easy to interpret, depending on your level of expertise. However, the last entries in the error log should provide useful information.

The following subsections contain a list of short descriptions for many of the directives included in **`httpd.conf`**. These descriptions are not exhaustive. For more information, refer to the Apache documentation online at *http://httpd.apache.org/docs/2.2/*.

For more information about **`mod_ssl`** directives, refer to the documentation online at *http://httpd.apache.org/docs/2.2/mod/mod_ssl.html*.

AccessFileName

`AccessFileName` names the file which the server should use for access control information in each directory. The default is **`.htaccess`**.

Immediately after the **`AccessFileName`** directive, a set of **`Files`** tags apply access control to any file beginning with a **`.ht`**. These directives deny Web access to any **`.htaccess`** files (or other files which begin with **`.ht`**) for security reasons.

Action

`Action` specifies a MIME content type and CGI script pair, so that when a file of that media type is requested, a particular CGI script is executed.

AddDescription

When using **`FancyIndexing`** as an **`IndexOptions`** parameter, the **`AddDescription`** directive can be used to display user-specified descriptions for certain files or file types in a server generated directory listing. The **`AddDescription`** directive supports listing specific files, wildcard expressions, or file extensions.

AddEncoding

`AddEncoding` names file name extensions which should specify a particular encoding type. **`AddEncoding`** can also be used to instruct some browsers to uncompress certain files as they are downloaded.

AddHandler

`AddHandler` maps file extensions to specific handlers. For example, the **`cgi-script`** handler can be matched with the extension **`.cgi`** to automatically treat a file ending with **`.cgi`** as a CGI script. The following is a sample **`AddHandler`** directive for the **`.cgi`** extension.

```
AddHandler cgi-script .cgi
```

This directive enables CGIs outside of the **`cgi-bin`** to function in any directory on the server which has the **`ExecCGI`** option within the directories container. Refer to *Directory* for more information about setting the **`ExecCGI`** option for a directory.

In addition to CGI scripts, the **AddHandler** directive is used to process server-parsed HTML and image-map files.

AddIcon

AddIcon specifies which icon to show in server generated directory listings for files with certain extensions. For example, the Web server is set to show the icon **binary.gif** for files with **.bin** or **.exe** extensions.

AddIconByEncoding

This directive names icons which are displayed by files with MIME encoding in server generated directory listings. For example, by default, the Web server shows the **compressed.gif** icon next to MIME encoded x-compress and x-gzip files in server generated directory listings.

AddIconByType

This directive names icons which are displayed next to files with MIME types in server generated directory listings. For example, the server shows the icon **text.gif** next to files with a mime-type of `text`, in server generated directory listings.

AddLanguage

AddLanguage associates file name extensions with specific languages. This directive is useful for Apache HTTP Servers which serve content in multiple languages based on the client Web browser's language settings.

AddType

Use the **AddType** directive to define or override a default MIME type and file extension pairs. The following example directive tells the Apache HTTP Server to recognize the **.tgz** file extension:

```
AddType application/x-tar .tgz
```

Alias

The **Alias** setting allows directories outside the **DocumentRoot** directory to be accessible. Any URL ending in the alias automatically resolves to the alias' path. By default, one alias for an **icons/** directory is already set up. An **icons/** directory can be accessed by the Web server, but the directory is not in the **DocumentRoot**.

Allow

Allow specifies which client can access a given directory. The client can be **all**, a domain name, an IP address, a partial IP address, a network/netmask pair, and so on. The **DocumentRoot** directory is configured to **Allow** requests from **all**, meaning everyone has access.

AllowOverride

The **AllowOverride** directive sets whether any **Options** can be overridden by the declarations in an **.htaccess** file. By default, both the root directory and the **DocumentRoot** are set to allow no **.htaccess** overrides.

BrowserMatch

The **BrowserMatch** directive allows the server to define environment variables and take appropriate actions based on the User-Agent HTTP header field — which identifies the client's Web browser type. By default, the Web server uses **BrowserMatch** to deny connections to specific browsers with known problems and also to disable keepalives and HTTP header flushes for browsers that are known to have problems with those actions.

Cache Directives

A number of commented cache directives are supplied by the default Apache HTTP Server configuration file. In most cases, uncommenting these lines by removing the hash mark (#) from the beginning of the line is sufficient. The following, however, is a list of some of the more important cache-related directives.

- **CacheEnable** — Specifies whether the cache is a disk, memory, or file descriptor cache. By default **CacheEnable** configures a disk cache for URLs at or below **/**.
- **CacheRoot** — Specifies the name of the directory containing cached files. The default **CacheRoot** is the **/var/httpd/proxy/** directory.
- **CacheSize** — Specifies how much space the cache can use in kilobytes. The default **CacheSize** is **5** KB.

The following is a list of some of the other common cache-related directives.

- **CacheMaxExpire** — Specifies how long HTML documents are retained (without a reload from the originating Web server) in the cache. The default is **24** hours (**86400** seconds).
- **CacheLastModifiedFactor** — Specifies the creation of an expiry (expiration) date for a document which did not come from its originating server with its own expiry set. The default **CacheLastModifiedFactor** is set to **0.1**, meaning that the expiry date for such documents equals one-tenth of the amount of time since the document was last modified.
- **CacheDefaultExpire** — Specifies the expiry time in hours for a document that was received using a protocol that does not support expiry times. The default is set to **1** hour (**3600** seconds).
- **NoProxy** — Specifies a space-separated list of subnets, IP addresses, domains, or hosts whose content is not cached. This setting is most useful for Intranet sites.

CacheNegotiatedDocs

By default, the Web server asks proxy servers not to cache any documents which were negotiated on the basis of content (that is, they may change over time or because of the input from the requester). If **CacheNegotiatedDocs** is set to on, this function is disabled and proxy servers are allowed to cache such documents.

CustomLog

CustomLog identifies the log file and the log file format. By default, the access log is recorded to the **/var/log/httpd/access_log** file while errors are recorded in the **/var/log/httpd/error_log** file.

The default **CustomLog** format is the `combined` log file format, as illustrated here:

```
remotehost rfc931 user date "request" status bytes referrer user-agent
```

DefaultIcon

DefaultIcon specifies the icon displayed in server generated directory listings for files which have no other icon specified. The **unknown.gif** image file is the default.

DefaultType

DefaultType sets a default content type for the Web server to use for documents whose MIME types cannot be determined. The default is **text/plain**.

Deny

Deny works similar to **Allow**, except it specifies who is denied access. The **DocumentRoot** is not configured to **Deny** requests from anyone by default.

Directory

<Directory /path/to/directory> and **</Directory>** tags create a container used to enclose a group of configuration directives which apply only to a specific directory and its subdirectories. Any directive which is applicable to a directory may be used within **Directory** tags.

By default, very restrictive parameters are applied to the root directory (**/**), using the **Options** (refer to *Options*) and **AllowOverride** (refer to *AllowOverride*) directives. Under this configuration, any directory on the system which needs more permissive settings has to be explicitly given those settings.

In the default configuration, another **Directory** container is configured for the **DocumentRoot** which assigns less rigid parameters to the directory tree so that the Apache HTTP Server can access the files residing there.

The **Directory** container can be also be used to configure additional **cgi-bin** directories for server-side applications outside of the directory specified in the **ScriptAlias** directive (refer to *ScriptAlias* for more information).

To accomplish this, the **Directory** container must set the **ExecCGI** option for that directory.

For example, if CGI scripts are located in **/home/my_cgi_directory**, add the following **Directory** container to the **httpd.conf** file:

```
<Directory /home/my_cgi_directory> Options +ExecCGI </Directory>
```

Next, the **AddHandler** directive must be uncommented to identify files with the **.cgi** extension as CGI scripts. Refer to *AddHandler* for instructions on setting **AddHandler**.

For this to work, permissions for CGI scripts, and the entire path to the scripts, must be set to 0755.

DirectoryIndex

The **DirectoryIndex** is the default page served by the server when a user requests an index of a directory by specifying a forward slash (/) at the end of the directory name.

When a user requests the page http://*example*/*this_directory*/, they get either the **DirectoryIndex** page, if it exists, or a server-generated directory list. The default for **DirectoryIndex** is **index.html** and the **index.html.var** type map. The server tries to find either of these files and returns the first one it finds. If it does not find one of these files and **Options Indexes** is set for that directory, the server generates and returns a listing, in HTML format, of the subdirectories and files within the directory, unless the directory listing feature is turned off.

DocumentRoot

DocumentRoot is the directory which contains most of the HTML files which are served in response to requests. The default **DocumentRoot**, for both the non-secure and secure Web servers, is the **/var/www/html** directory. For example, the server might receive a request for the following document:

```
http://example.com/foo.html
```

The server looks for the following file in the default directory:

```
/var/www/html/foo.html
```

To change the **DocumentRoot** so that it is not shared by the secure and the non-secure Web servers, refer to *Section 11.7, “Virtual Hosts”*.

ErrorDocument

The **ErrorDocument** directive associates an HTTP response code with a message or a URL to be sent back to the client. By default, the Web server outputs a simple and usually cryptic error message when an error occurs. The **ErrorDocument** directive forces the Web server to instead output a customized message or page.

> **Important**
>
> To be valid, the message *must* be enclosed in a pair of double quotes **"**.

ErrorLog

ErrorLog specifies the file where server errors are logged. By default, this directive is set to **/var/log/httpd/error_log**.

ExtendedStatus

The **ExtendedStatus** directive controls whether Apache generates basic (**off**) or detailed server status information (**on**), when the **server-status** handler is called. The **server-status** handler is called using **Location** tags. More information on calling **server-status** is included in *Location*.

Group

Specifies the group name of the Apache HTTP Server processes.

This directive has been deprecated for the configuration of virtual hosts.

By default, **`Group`** is set to **`apache`**.

HeaderName

`HeaderName` names the file which, if it exists in the directory, is prepended to the start of server generated directory listings. Like **`ReadmeName`**, the server tries to include it as an HTML document if possible or in plain text if not.

HostnameLookups

`HostnameLookups` can be set to `on`, `off`, or `double`. If **`HostnameLookups`** is set to `on`, the server automatically resolves the IP address for each connection. Resolving the IP address means that the server makes one or more connections to a DNS server, adding processing overhead. If **`HostnameLookups`** is set to `double`, the server performs a double-reverse DNS look up adding even more processing overhead.

To conserve resources on the server, **`HostnameLookups`** is set to `off` by default.

If hostnames are required in server log files, consider running one of the many log analyzer tools that perform the DNS lookups more efficiently and in bulk when rotating the Web server log files.

IfDefine

The **`IfDefine`** tags surround configuration directives that are applied if the "test" stated in the **`IfDefine`** tag is true. The directives are ignored if the test is false.

The test in the **`IfDefine`** tags is a parameter name (for example, **`HAVE_PERL`**). If the parameter is defined, meaning that it is provided as an argument to the server's start-up command, then the test is true. In this case, when the Web server is started, the test is true and the directives contained in the **`IfDefine`** tags are applied.

IfModule

`<IfModule>` and **`</IfModule>`** tags create a conditional container which are only activated if the specified module is loaded. Directives within the **`IfModule`** container are processed under one of two conditions. The directives are processed if the module contained within the starting **`<IfModule>`** tag is loaded. Or, if an exclamation point **`!`** appears before the module name, the directives are processed only if the module specified in the **`<IfModule>`** tag is *not* loaded.

For more information about Apache HTTP Server modules, refer to *Section 11.6, “Adding Modules”*.

Include

`Include` allows other configuration files to be included at runtime.

The path to these configuration files can be absolute or relative to the **`ServerRoot`**.

Important

For the server to use individually packaged modules, such as **`mod_ssl`**, **`mod_perl`**, and **`php`**, the following directive must be included in **`Section 1: Global Environment`** of **`httpd.conf`**:

```
Include conf.d/*.conf
```

IndexIgnore

IndexIgnore lists file extensions, partial file names, wildcard expressions, or full file names. The Web server does not include any files which match any of those parameters in server generated directory listings.

IndexOptions

IndexOptions controls the appearance of server generated directing listings, by adding icons, file descriptions, and so on. If **Options Indexes** is set (refer to *Options*), the Web server generates a directory listing when the Web server receives an HTTP request for a directory without an index.

First, the Web server looks in the requested directory for a file matching the names listed in the **DirectoryIndex** directive (usually, **index.html**). If an **index.html** file is not found, Apache HTTP Server creates an HTML directory listing of the requested directory. The appearance of this directory listing is controlled, in part, by the **IndexOptions** directive.

The default configuration turns on **FancyIndexing**. This means that a user can re-sort a directory listing by clicking on column headers. Another click on the same header switches from ascending to descending order. **FancyIndexing** also shows different icons for different files, based upon file extensions.

The **AddDescription** option, when used in conjunction with **FancyIndexing**, presents a short description for the file in server generated directory listings.

IndexOptions has a number of other parameters which can be set to control the appearance of server generated directories. The **IconHeight** and **IconWidth** parameters require the server to include HTML **HEIGHT** and **WIDTH** tags for the icons in server generated webpages. The **IconsAreLinks** parameter combines the graphical icon with the HTML link anchor, which contains the URL link target.

KeepAlive

KeepAlive sets whether the server allows more than one request per connection and can be used to prevent any one client from consuming too much of the server's resources.

By default **Keepalive** is set to **off**. If **Keepalive** is set to **on** and the server becomes very busy, the server can quickly spawn the maximum number of child processes. In this situation, the server slows down significantly. If **Keepalive** is enabled, it is a good idea to set the the **KeepAliveTimeout** low (refer to *KeepAliveTimeout* for more information about the **KeepAliveTimeout** directive) and monitor the **/var/log/httpd/error_log** log file on the server. This log reports when the server is running out of child processes.

KeepAliveTimeout

KeepAliveTimeout sets the number of seconds the server waits after a request has been served before it closes the connection. Once the server receives a request, the **Timeout** directive applies instead. The **KeepAliveTimeout** directive is set to 15 seconds by default.

LanguagePriority

`LanguagePriority` sets precedence for different languages in case the client Web browser has no language preference set.

Listen

The **`Listen`** command identifies the ports on which the Web server accepts incoming requests. By default, the Apache HTTP Server is set to listen to port 80 for non-secure Web communications and (in the **`/etc/httpd/conf.d/ssl.conf`** file which defines any secure servers) to port 443 for secure Web communications.

If the Apache HTTP Server is configured to listen to a port under 1024, only the root user can start it. For port 1024 and above, **`httpd`** can be started as a regular user.

The **`Listen`** directive can also be used to specify particular IP addresses over which the server accepts connections.

LoadModule

`LoadModule` is used to load Dynamic Shared Object (DSO) modules. More information on the Apache HTTP Server's DSO support, including instructions for using the **`LoadModule`** directive, can be found in *Section 11.6, “Adding Modules”*. Note, the load order of the modules is *no longer important* with Apache HTTP Server 2.0. Refer to *Section 11.2.2.1.3, “Dynamic Shared Object (DSO) Support”* for more information about Apache HTTP Server 2.0 DSO support.

Location

The **`<Location>`** and **`</Location>`** tags create a container in which access control based on URL can be specified.

For instance, to allow people connecting from within the server's domain to see status reports, use the following directives:

```
<Location /server-status> SetHandler server-status Order deny,allow Deny
 from all Allow from <.example.com> </Location>
```

Replace <*`.example.com`*> with the second-level domain name for the Web server.

To provide server configuration reports (including installed modules and configuration directives) to requests from inside the domain, use the following directives:

```
<Location /server-info> SetHandler server-info Order deny,allow Deny from
 all Allow from <.example.com> </Location>
```

Again, replace <*`.example.com`*> with the second-level domain name for the Web server.

LogFormat

The **`LogFormat`** directive configures the format of the various Web server log files. The actual **`LogFormat`** used depends on the settings given in the **`CustomLog`** directive (refer to *CustomLog*).

The following are the format options if the **`CustomLog`** directive is set to **`combined`**:

`%h` (remote host's IP address or hostname)
: Lists the remote IP address of the requesting client. If **`HostnameLookups`** is set to `on`, the client hostname is recorded unless it is not available from DNS.

`%l` (rfc931)
: Not used. A hyphen **`-`** appears in the log file for this field.

`%u` (authenticated user)
: Lists the username of the user recorded if authentication was required. Usually, this is not used, so a hyphen **`-`** appears in the log file for this field.

`%t` (date)
: Lists the date and time of the request.

`%r` (request string)
: Lists the request string exactly as it came from the browser or client.

`%s` (status)
: Lists the HTTP status code which was returned to the client host.

`%b` (bytes)
: Lists the size of the document.

`%\"%{Referer}i\"` (referrer)
: Lists the URL of the webpage which referred the client host to Web server.

`%\"%{User-Agent}i\"` (user-agent)
: Lists the type of Web browser making the request.

LogLevel

`LogLevel` sets how verbose the error messages in the error logs are. **`LogLevel`** can be set (from least verbose to most verbose) to **`emerg`**, **`alert`**, **`crit`**, **`error`**, **`warn`**, **`notice`**, **`info`**, or **`debug`**. The default **`LogLevel`** is **`warn`**.

MaxKeepAliveRequests

This directive sets the maximum number of requests allowed per persistent connection.
The Apache Project recommends a high setting, which improves the server's performance.
`MaxKeepAliveRequests` is set to **`100`** by default, which should be appropriate for most situations.

NameVirtualHost

The **`NameVirtualHost`** directive associates an IP address and port number, if necessary, for any name-based virtual hosts. Name-based virtual hosting allows one Apache HTTP Server to serve different domains without using multiple IP addresses.

> **Note**
>
> Name-based virtual hosts *only* work with non-secure HTTP connections. If using virtual hosts with a secure server, use IP address-based virtual hosts instead.

To enable name-based virtual hosting, uncomment the **`NameVirtualHost`** configuration directive and add the correct IP address. Then add additional **`VirtualHost`** containers for each virtual host as is necessary for your configuration.

Options

The **`Options`** directive controls which server features are available in a particular directory. For example, under the restrictive parameters specified for the root directory, **`Options`** is only set to the **`FollowSymLinks`** directive. No features are enabled, except that the server is allowed to follow symbolic links in the root directory.

By default, in the **`DocumentRoot`** directory, **`Options`** is set to include **`Indexes`** and **`FollowSymLinks`**. **`Indexes`** permits the server to generate a directory listing for a directory if no **`DirectoryIndex`** (for example, **`index.html`**) is specified. **`FollowSymLinks`** allows the server to follow symbolic links in that directory.

> **Note**
>
> **`Options`** statements from the main server configuration section need to be replicated to each **`VirtualHost`** container individually. Refer to *VirtualHost* for more information.

Order

The **`Order`** directive controls the order in which **`allow`** and **`deny`** directives are evaluated. The server is configured to evaluate the **`Allow`** directives before the **`Deny`** directives for the **`DocumentRoot`** directory.

PidFile

`PidFile` names the file where the server records its process ID (PID). By default the PID is listed in **`/var/run/httpd.pid`**.

Proxy

`<Proxy *>` and **`</Proxy>`** tags create a container which encloses a group of configuration directives meant to apply only to the proxy server. Many directives which are allowed within a **`<Directory>`** container may also be used within **`<Proxy>`** container.

ProxyRequests

To configure the Apache HTTP Server to function as a proxy server, remove the hash mark (**`#`**) from the beginning of the **`<IfModule mod_proxy.c>`** line, the ProxyRequests, and each line in the **`<Proxy>`** stanza. Set the **`ProxyRequests`** directive to **`On`**, and set which domains are allowed access to the server in the **`Allow from`** directive of the **`<Proxy>`** stanza.

ReadmeName

`ReadmeName` names the file which, if it exists in the directory, is appended to the end of server generated directory listings. The Web server first tries to include the file as an HTML document and then tries to include it as plain text. By default, **`ReadmeName`** is set to **`README.html`**.

Redirect

When a webpage is moved, **`Redirect`** can be used to map the file location to a new URL. The format is as follows:

```
Redirect /<old-path>/<file-name> http://<current-domain>/<current-path>/<file-name>
```

In this example, replace <*`old-path`*> with the old path information for <*`file-name`*> and <*`current-domain`*> and <*`current-path`*> with the current domain and path information for <*`file-name`*>.

In this example, any requests for <*`file-name`*> at the old location is automatically redirected to the new location.

For more advanced redirection techniques, use the **`mod_rewrite`** module included with the Apache HTTP Server. For more information about configuring the **`mod_rewrite`** module, refer to the Apache Software Foundation documentation online at *http://httpd.apache.org/docs/2.2/mod/mod_rewrite.html*[23].

ScriptAlias

The **`ScriptAlias`** directive defines where CGI scripts are located. Generally, it is not good practice to leave CGI scripts within the **`DocumentRoot`**, where they can potentially be viewed as text documents. For this reason, a special directory outside of the **`DocumentRoot`** directory containing server-side executables and scripts is designated by the **`ScriptAlias`** directive. This directory is known as a **`cgi-bin`** and is set to **`/var/www/cgi-bin/`** by default.

It is possible to establish directories for storing executables outside of the **`cgi-bin/`** directory. For instructions on doing so, refer to *AddHandler* and *Directory*.

ServerAdmin

Sets the **`ServerAdmin`** directive to the email address of the Web server administrator. This email address shows up in error messages on server-generated Web pages, so users can report a problem by sending email to the server administrator.

By default, **`ServerAdmin`** is set to **`root@localhost`**.

A common way to set up **`ServerAdmin`** is to set it to **`webmaster@example.com`**. Once set, alias **`webmaster`** to the person responsible for the Web server in **`/etc/aliases`** and run **`/usr/bin/newaliases`**.

ServerName

`ServerName` specifies a hostname and port number (matching the **`Listen`** directive) for the server. The **`ServerName`** does not need to match the machine's actual hostname. For example, the Web server may be `www.example.com`, but the server's hostname is actually `foo.example.com`. The value specified in **`ServerName`** must be a valid Domain Name Service (DNS) name that can be resolved by the system — do not make something up.

[23] http://httpd.apache.org/docs/2.2/mod/mod_rewrite.html

The following is a sample **ServerName** directive:

```
ServerName www.example.com:80
```

When specifying a **ServerName**, be sure the IP address and server name pair are included in the **/etc/hosts** file.

ServerRoot

The **ServerRoot** directive specifies the top-level directory containing website content. By default, **ServerRoot** is set to **"/etc/httpd"** for both secure and non-secure servers.

ServerSignature

The **ServerSignature** directive adds a line containing the Apache HTTP Server server version and the **ServerName** to any server-generated documents, such as error messages sent back to clients. **ServerSignature** is set to **on** by default.

ServerSignature can be set to **EMail** which adds a **mailto:ServerAdmin** HTML tag to the signature line of auto-generated responses. **ServerSignature** can also be set to **Off** to stop Apache from sending out its version number and module information. Please also check the **ServerTokens** settings.

ServerTokens

The **ServerTokens** directive determines if the Server response header field sent back to clients should include details of the Operating System type and information about compiled-in modules. By default, **ServerTokens** is set to **Full** which sends information about the Operating System type and compiled-in modules. Setting the **ServerTokens** to **Prod** sends the product name only and is recommended as many hackers check information in the Server header when scanning for vulnerabilities. You can also set the **ServerTokens** to **Min** (minimal) or to **OS** (operating system).

SuexecUserGroup

The **SuexecUserGroup** directive, which originates from the **mod_suexec** module, allows the specification of user and group execution privileges for CGI programs. Non-CGI requests are still processed with the user and group specified in the **User** and **Group** directives.

> **Note**
>
> From version 2.0, the **SuexecUserGroup** directive replaced the Apache HTTP Server 1.3 configuration of using the **User** and **Group** directives inside the configuration of **VirtualHosts** sections.

Timeout

Timeout defines, in seconds, the amount of time that the server waits for receipts and transmissions during communications. **Timeout** is set to **300** seconds by default, which is appropriate for most situations.

TypesConfig

`TypesConfig` names the file which sets the default list of MIME type mappings (file name extensions to content types). The default **`TypesConfig`** file is **`/etc/mime.types`**. Instead of editing **`/etc/mime.types`**, the recommended way to add MIME type mappings is to use the **`AddType`** directive.

For more information about **`AddType`**, refer to *AddType*.

UseCanonicalName

When set to on, this directive configures the Apache HTTP Server to reference itself using the value specified in the **`ServerName`** and **`Port`** directives. When **`UseCanonicalName`** is set to `off`, the server instead uses the value used by the requesting client when referring to itself.

`UseCanonicalName` is set to `off` by default.

User

The **`User`** directive sets the username of the server process and determines what files the server is allowed to access. Any files inaccessible to this user are also inaccessible to clients connecting to the Apache HTTP Server.

By default **`User`** is set to **`apache`**.

This directive has been deprecated for the configuration of virtual hosts.

> **Note**
>
> For security reasons, the Apache HTTP Server does not run as the root user.

UserDir

`UserDir` is the subdirectory within each user's home directory where they should place personal HTML files which are served by the Web server. This directive is set to `disable` by default.

The name for the subdirectory is set to **`public_html`** in the default configuration. For example, the server might receive the following request:

```
http://example.com/~username/foo.html
```

The server would look for the file:

```
/home/username/public_html/foo.html
```

In the above example, **`/home/username/`** is the user's home directory (note that the default path to users' home directories may vary).

Make sure that the permissions on the users' home directories are set correctly. Users' home directories must be set to 0711. The read (r) and execute (x) bits must be set on the users'

public_html directories (0755 also works). Files that are served in a users' **public_html** directories must be set to at least 0644.

VirtualHost

<VirtualHost> and **</VirtualHost>** tags create a container outlining the characteristics of a virtual host. The **VirtualHost** container accepts most configuration directives.

A commented **VirtualHost** container is provided in **httpd.conf**, which illustrates the minimum set of configuration directives necessary for each virtual host. Refer to *Section 11.7, "Virtual Hosts"* for more information about virtual hosts.

> **Note**
>
> The default SSL virtual host container now resides in the file **/etc/httpd/conf.d/ssl.conf**.

11.5.2. Configuration Directives for SSL

The directives in **/etc/httpd/conf.d/ssl.conf** file can be configured to enable secure Web communications using SSL and TLS.

SetEnvIf

SetEnvIf sets environment variables based on the headers of incoming connections. It is *not* solely an SSL directive, though it is present in the supplied **/etc/httpd/conf.d/ssl.conf** file. It's purpose in this context is to disable HTTP keepalive and to allow SSL to close the connection without a closing notification from the client browser. This setting is necessary for certain browsers that do not reliably shut down the SSL connection.

For more information on other directives within the SSL configuration file, refer to the following URLs:

- http://localhost/manual/mod/mod_ssl.html
- *http://httpd.apache.org/docs/2.2/mod/mod_ssl.html*

> **Note**
>
> In most cases, SSL directives are configured appropriately during the installation of Fedora. Be careful when altering Apache HTTP Secure Server directives, misconfiguration can lead to security vulnerabilities.

11.5.3. MPM Specific Server-Pool Directives

As explained in *Section 11.2.2.1.2, "Server-Pool Size Regulation"*, the responsibility for managing characteristics of the server-pool falls to a module group called MPMs under Apache HTTP Server 2.0. The characteristics of the server-pool differ depending upon which MPM is used. For this reason, an **IfModule** container is necessary to define the server-pool for the MPM in use.

By default, Apache HTTP Server 2.0 defines the server-pool for both the **prefork** and **worker** MPMs.

The following section list directives found within the MPM-specific server-pool containers.

MaxClients

`MaxClients` sets a limit on the total number of server processes, or simultaneously connected clients, that can run at one time. The main purpose of this directive is to keep a runaway Apache HTTP Server from crashing the operating system. For busy servers this value should be set to a high value. The server's default is set to 150 regardless of the MPM in use. However, it is not recommended that the value for **`MaxClients`** exceeds **`256`** when using the **`prefork`** MPM.

MaxRequestsPerChild

`MaxRequestsPerChild` sets the total number of requests each child server process serves before the child dies. The main reason for setting **`MaxRequestsPerChild`** is to avoid long-lived process induced memory leaks. The default **`MaxRequestsPerChild`** for the **`prefork`** MPM is **`4000`** and for the **`worker`** MPM is **`0`**.

MinSpareServers and MaxSpareServers

These values are only used with the **`prefork`** MPM. They adjust how the Apache HTTP Server dynamically adapts to the perceived load by maintaining an appropriate number of spare server processes based on the number of incoming requests. The server checks the number of servers waiting for a request and kills some if there are more than **`MaxSpareServers`** or creates some if the number of servers is less than **`MinSpareServers`**.

The default **`MinSpareServers`** value is **`5`**; the default **`MaxSpareServers`** value is **`20`**. These default settings should be appropriate for most situations. Be careful not to increase the **`MinSpareServers`** to a large number as doing so creates a heavy processing load on the server even when traffic is light.

MinSpareThreads and MaxSpareThreads

These values are only used with the **`worker`** MPM. They adjust how the Apache HTTP Server dynamically adapts to the perceived load by maintaining an appropriate number of spare server threads based on the number of incoming requests. The server checks the number of server threads waiting for a request and kills some if there are more than **`MaxSpareThreads`** or creates some if the number of servers is less than **`MinSpareThreads`**.

The default **`MinSpareThreads`** value is **`25`**; the default **`MaxSpareThreads`** value is **`75`**. These default settings should be appropriate for most situations. The value for **`MaxSpareThreads`** must be greater than or equal to the sum of **`MinSpareThreads`** and **`ThreadsPerChild`**, else the Apache HTTP Server automatically corrects it.

StartServers

The **`StartServers`** directive sets how many server processes are created upon startup. Since the Web server dynamically kills and creates server processes based on traffic load, it is not necessary to change this parameter. The Web server is set to start **`8`** server processes at startup for the **`prefork`** MPM and **`2`** for the **`worker`** MPM.

ThreadsPerChild

This value is only used with the **`worker`** MPM. It sets the number of threads within each child process. The default value for this directive is **`25`**.

11.6. Adding Modules

The Apache HTTP Server is distributed with a number of modules. More information about Apache HTTP modules can be found on *http://httpd.apache.org/docs/2.2/mod/*.

The Apache HTTP Server supports *Dynamically Shared Objects* (*DSO*s), or modules, which can easily be loaded at runtime as necessary.

The Apache Project provides complete DSO documentation online at *http://httpd.apache.org/docs/2.2/dso.html*. Or, if the **`http-manual`** package is installed, documentation about DSOs can be found online at *http://localhost/manual/mod/*.

For the Apache HTTP Server to use a DSO, it must be specified in a **`LoadModule`** directive within **`/etc/httpd/conf/httpd.conf`**. If the module is provided by a separate package, the line must appear within the modules configuration file in the **`/etc/httpd/conf.d/`** directory. Refer to *LoadModule* for more information.

If adding or deleting modules from **`http.conf`**, Apache HTTP Server must be reloaded or restarted, as referred to in *Section 11.3, “Starting and Stopping **`httpd`**”*.

If creating a new module, first install the **`httpd-devel`** package which contains the include files, the header files, as well as the *APache eXtenSion* (**`/usr/sbin/apxs`**) application, which uses the include files and the header files to compile DSOs.

After writing a module, use **`/usr/sbin/apxs`** to compile the module sources outside the Apache source tree. For more information about using the **`/usr/sbin/apxs`** command, refer to the the Apache documentation online at *http://httpd.apache.org/docs/2.2/dso.html* as well as the **`apxs`** man page.

Once compiled, put the module in the **`/usr/lib/httpd/modules/`** directory. For RHEL platforms using default-64-bit userspace (x86_64, ia64, ?) this path will be **`/usr/lib64/httpd/modules/`**. Then add a **`LoadModule`** line to the **`httpd.conf`**, using the following structure:

```
LoadModule <module-name> <path/to/module.so>
```

Where <*`module-name`*> is the name of the module and <*`path/to/module.so`*> is the path to the DSO.

11.7. Virtual Hosts

The Apache HTTP Server's built in virtual hosting allows the server to provide different information based on which IP address, hostname, or port is being requested. A complete guide to using virtual hosts is available online at *http://httpd.apache.org/docs/2.2/vhosts/*.

11.7.1. Setting Up Virtual Hosts

To create a name-based virtual host, it is best to use the virtual host container provided in **`httpd.conf`** as an example.

The virtual host example read as follows:

```
#NameVirtualHost *:80 # #<VirtualHost *:80> # ServerAdmin webmaster@dummy-host.example.com # DocumentRoot /www/docs/dummy-host.example.com #
 ServerName dummy-host.example.com # ErrorLog logs/dummy-host.example.com-error_log # CustomLog logs/dummy-host.example.com-access_log common #</VirtualHost>
```

To activate name-based virtual hosting, uncomment the **`NameVirtualHost`** line by removing the hash mark (`#`) and replace the asterisk (`*`) with the IP address assigned to the machine.

Next, configure a virtual host by uncommenting and customizing the **`<VirtualHost>`** container.

On the **`<VirtualHost>`** line, change the asterisk (`*`) to the server's IP address. Change the **`ServerName`** to a *valid* DNS name assigned to the machine, and configure the other directives as necessary.

The **`<VirtualHost>`** container is highly customizable and accepts almost every directive available within the main server configuration.

> **Tip**
>
> If configuring a virtual host to listen on a non-default port, that port must be added to the **`Listen`** directive in the global settings section of **`/etc/httpd/conf/httpd.conf`** file.

To activate a newly created virtual host, the Apache HTTP Server must be reloaded or restarted. Refer to *Section 11.3, "Starting and Stopping **`httpd`**"* for further instructions.

Comprehensive information about creating and configuring both name-based and IP address-based virtual hosts is provided online at *http://httpd.apache.org/docs/2.2/vhosts/*.

11.8. Apache HTTP Secure Server Configuration

This section provides basic information on the Apache HTTP Server with the **`mod_ssl`** security module enabled to use the OpenSSL library and toolkit. The combination of these three components are referred to in this section as the secure Web server or just as the secure server.

The **`mod_ssl`** module is a security module for the Apache HTTP Server. The **`mod_ssl`** module uses the tools provided by the OpenSSL Project to add a very important feature to the Apache HTTP Server — the ability to encrypt communications. In contrast, regular HTTP communications between a browser and a Web server are sent in plain text, which could be intercepted and read by someone along the route between the browser and the server.

This section is not meant to be complete and exclusive documentation for any of these programs. When possible, this guide points to appropriate places where you can find more in-depth documentation on particular subjects.

This section shows you how to install these programs. You can also learn the steps necessary to generate a private key and a certificate request, how to generate your own self-signed certificate, and how to install a certificate to use with your secure server.

The **`mod_ssl`** configuration file is located at **`/etc/httpd/conf.d/ssl.conf`**. For this file to be loaded, and hence for **`mod_ssl`** to work, you must have the statement `Include conf.d/*.conf` in the **`/etc/httpd/conf/httpd.conf`** file. This statement is included by default in the default Apache HTTP Server configuration file.

11.8.1. An Overview of Security-Related Packages

To enable the secure server, you must have the following packages installed at a minimum:

`httpd`

The **`httpd`** package contains the **`httpd`** daemon and related utilities, configuration files, icons, Apache HTTP Server modules, man pages, and other files used by the Apache HTTP Server.

`mod_ssl`

The **`mod_ssl`** package includes the **`mod_ssl`** module, which provides strong cryptography for the Apache HTTP Server via the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols.

`openssl`

The **`openssl`** package contains the OpenSSL toolkit. The OpenSSL toolkit implements the SSL and TLS protocols, and also includes a general purpose cryptography library.

Additionally, other software packages provide certain security functionalities (but are not required by the secure server to function):

11.8.2. An Overview of Certificates and Security

Your secure server provides security using a combination of the Secure Sockets Layer (SSL) protocol and (in most cases) a digital certificate from a Certificate Authority (CA). SSL handles the encrypted communications as well as the mutual authentication between browsers and your secure server. The CA-approved digital certificate provides authentication for your secure server (the CA puts its reputation behind its certification of your organization's identity). When your browser is communicating using SSL encryption, the `https://` prefix is used at the beginning of the Uniform Resource Locator (URL) in the navigation bar.

Encryption depends upon the use of keys (think of them as secret encoder/decoder rings in data format). In conventional or symmetric cryptography, both ends of the transaction have the same key, which they use to decode each other's transmissions. In public or asymmetric cryptography, two keys co-exist: a public key and a private key. A person or an organization keeps their private key a secret and publishes their public key. Data encoded with the public key can only be decoded with the private key; data encoded with the private key can only be decoded with the public key.

To set up your secure server, use public cryptography to create a public and private key pair. In most cases, you send your certificate request (including your public key), proof of your company's identity, and payment to a CA. The CA verifies the certificate request and your identity, and then sends back a certificate for your secure server.

A secure server uses a certificate to identify itself to Web browsers. You can generate your own certificate (called a "self-signed" certificate), or you can get a certificate from a CA. A certificate from a reputable CA guarantees that a website is associated with a particular company or organization.

Alternatively, you can create your own self-signed certificate. Note, however, that self-signed certificates should not be used in most production environments. Self-signed certificates are not automatically accepted by a user's browser — users are prompted by the browser to accept the certificate and create the secure connection. Refer to *Section 11.8.4, “Types of Certificates”* for more information on the differences between self-signed and CA-signed certificates.

Once you have a self-signed certificate or a signed certificate from the CA of your choice, you must install it on your secure server.

11.8.3. Using Pre-Existing Keys and Certificates

If you already have an existing key and certificate (for example, if you are installing the secure server to replace another company's secure server product), you can probably use your existing key and certificate with the secure server. The following two situations provide instances where you are not able to use your existing key and certificate:

- *If you are changing your IP address or domain name* — Certificates are issued for a particular IP address and domain name pair. You must get a new certificate if you are changing your IP address or domain name.
- *If you have a certificate from VeriSign and you are changing your server software* — VeriSign is a widely used CA. If you already have a VeriSign certificate for another purpose, you may have been considering using your existing VeriSign certificate with your new secure server. However, you are not be allowed to because VeriSign issues certificates for one specific server software and IP address/domain name combination.

  If you change either of those parameters (for example, if you previously used a different secure server product), the VeriSign certificate you obtained to use with the previous configuration will not work with the new configuration. You must obtain a new certificate.

If you have an existing key and certificate that you can use, you do not have to generate a new key and obtain a new certificate. However, you may need to move and rename the files which contain your key and certificate.

Move your existing key file to:

```
/etc/pki/tls/private/server.key
```

Move your existing certificate file to:

```
/etc/pki/tls/certs/server.crt
```

If you are upgrading from the Red Hat Secure Web Server, your old key (**`httpsd.key`**) and certificate (**`httpsd.crt`**) are located in **`/etc/httpd/conf/`**. Move and rename your key and certificate so that the secure server can use them. Use the following two commands to move and rename your key and certificate files:

```
mv /etc/httpd/conf/httpsd.key /etc/pki/tls/private/server.key mv /etc/
httpd/conf/httpsd.crt /etc/pki/tls/certs/server.crt
```

Then, start your secure server with the command:

```
/sbin/service httpd start
```

11.8.4. Types of Certificates

If you installed your secure server from the RPM package provided by Red Hat, a randomly generated private key and a test certificate are generated and put into the appropriate directories. Before you begin using your secure server, however, you must generate your own key and obtain a certificate which correctly identifies your server.

You need a key and a certificate to operate your secure server — which means that you can either generate a self-signed certificate or purchase a CA-signed certificate from a CA. What are the differences between the two?

A CA-signed certificate provides two important capabilities for your server:

- Browsers (usually) automatically recognize the certificate and allow a secure connection to be made, without prompting the user.

- When a CA issues a signed certificate, they are guaranteeing the identity of the organization that is providing the webpages to the browser.

If your secure server is being accessed by the public at large, your secure server needs a certificate signed by a CA so that people who visit your website know that the website is owned by the organization who claims to own it. Before signing a certificate, a CA verifies that the organization requesting the certificate was actually who they claimed to be.

Most Web browsers that support SSL have a list of CAs whose certificates they automatically accept. If a browser encounters a certificate whose authorizing CA is not in the list, the browser asks the user to either accept or decline the connection.

You can generate a self-signed certificate for your secure server, but be aware that a self-signed certificate does not provide the same functionality as a CA-signed certificate. A self-signed certificate is not automatically recognized by most Web browsers and does not provide any guarantee concerning the identity of the organization that is providing the website. A CA-signed certificate provides both of these important capabilities for a secure server. If your secure server is to be used in a production environment, a CA-signed certificate is recommended.

The process of getting a certificate from a CA is fairly easy. A quick overview is as follows:

1. Create an encryption private and public key pair.

2. Create a certificate request based on the public key. The certificate request contains information about your server and the company hosting it.

3. Send the certificate request, along with documents proving your identity, to a CA. Red Hat does not make recommendations on which certificate authority to choose. Your decision may be based on your past experiences, on the experiences of your friends or colleagues, or purely on monetary factors.

   Once you have decided upon a CA, you need to follow the instructions they provide on how to obtain a certificate from them.

4. When the CA is satisfied that you are indeed who you claim to be, they provide you with a digital certificate.

5. Install this certificate on your secure server and begin handling secure transactions.

Whether you are getting a certificate from a CA or generating your own self-signed certificate, the first step is to generate a key. Refer to *Section 11.8.5, “Generating a Key”* for instructions.

11.8.5. Generating a Key

You must be root to generate a key.

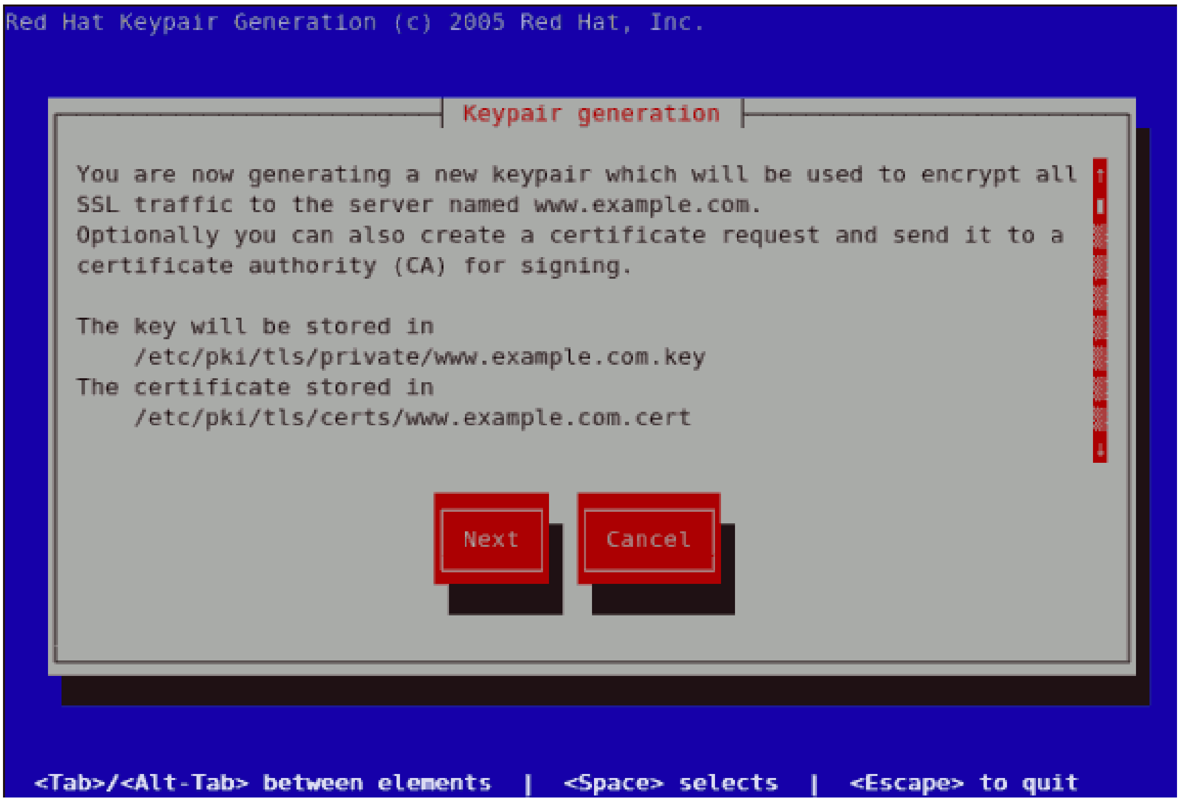
First, use the **cd** command to change to the **/etc/httpd/conf/** directory. Remove the fake key and certificate that were generated during the installation with the following commands:

```
rm ssl.key/server.key
        rm ssl.crt/server.crt
```

The **crypto-utils** package contains the **genkey** utility which you can use to generate keys as the name implies. To create your own private key, please ensure the **crypto-utils** package is installed. You can view more options by typing **man genkey** in your terminal. Assuming you wish to generate keys for www.example.com using the **genkey** utility, type in the following command in your terminal:

```
genkey www.example.com
```

Please note that the **make** based process is no longer shipped with RHEL 5. This will start the **genkey** graphical user interface. The figure below illustrates the first screen. To navigate, use the keyboard arrow and tab keys. This windows indicates where your key will be stored and prompts you to proceed or cancel the operation. To proceed to the next step, select **Next** and press the Return (Enter) key.

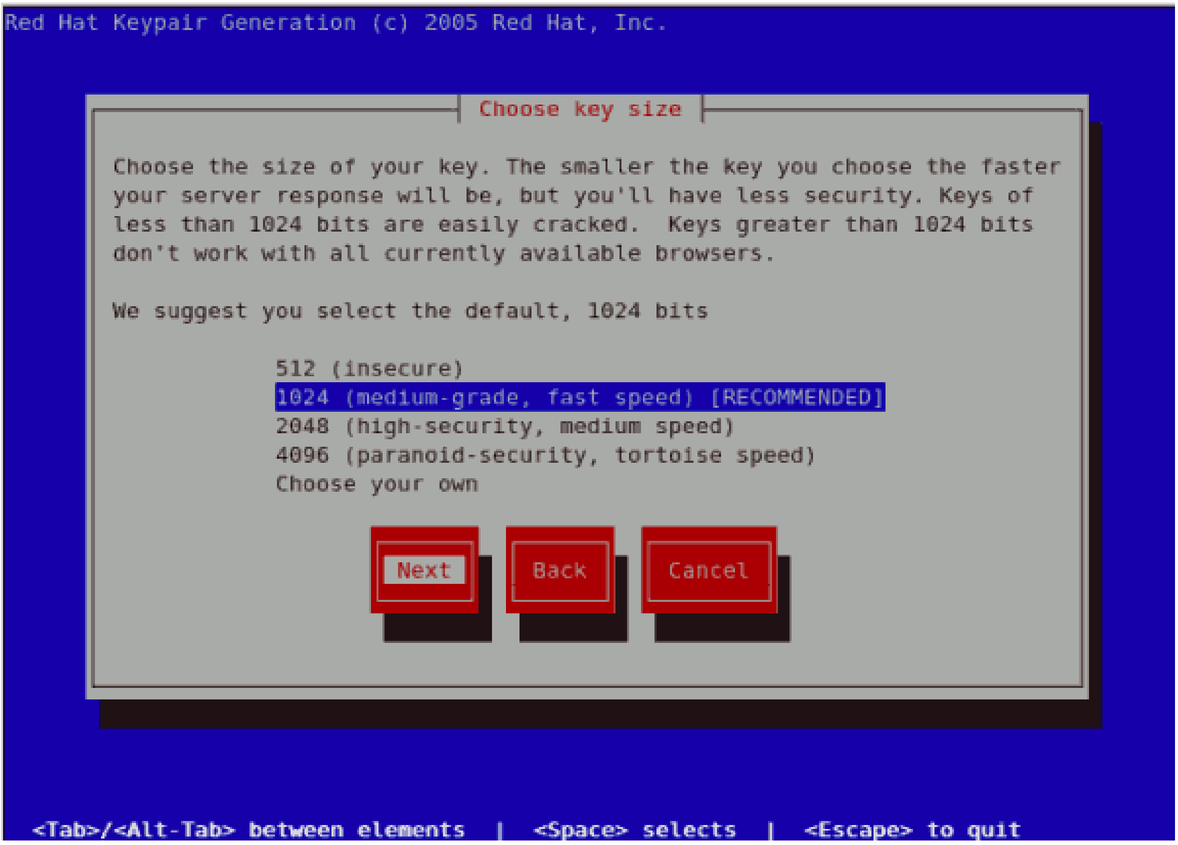


```
Red Hat Keypair Generation (c) 2005 Red Hat, Inc.

Keypair generation

You are now generating a new keypair which will be used to encrypt all
SSL traffic to the server named www.example.com.
Optionally you can also create a certificate request and send it to a
certificate authority (CA) for signing.

The key will be stored in
    /etc/pki/tls/private/www.example.com.key
The certificate stored in
    /etc/pki/tls/certs/www.example.com.cert
```

Next
Cancel
<Tab>/<Alt-Tab> between elements | <Space> selects | <Escape> to quit

Figure 11.11. Keypair generation

The next screen prompts you to choose the size of your key. As indicated, the smaller the size of your key, the faster will the response from your server be and the lesser your level of security. On selecting your preferred, key size using the arrow keys, select **Next** to proceed to the next step. The figure below illustrates the key size selection screen.

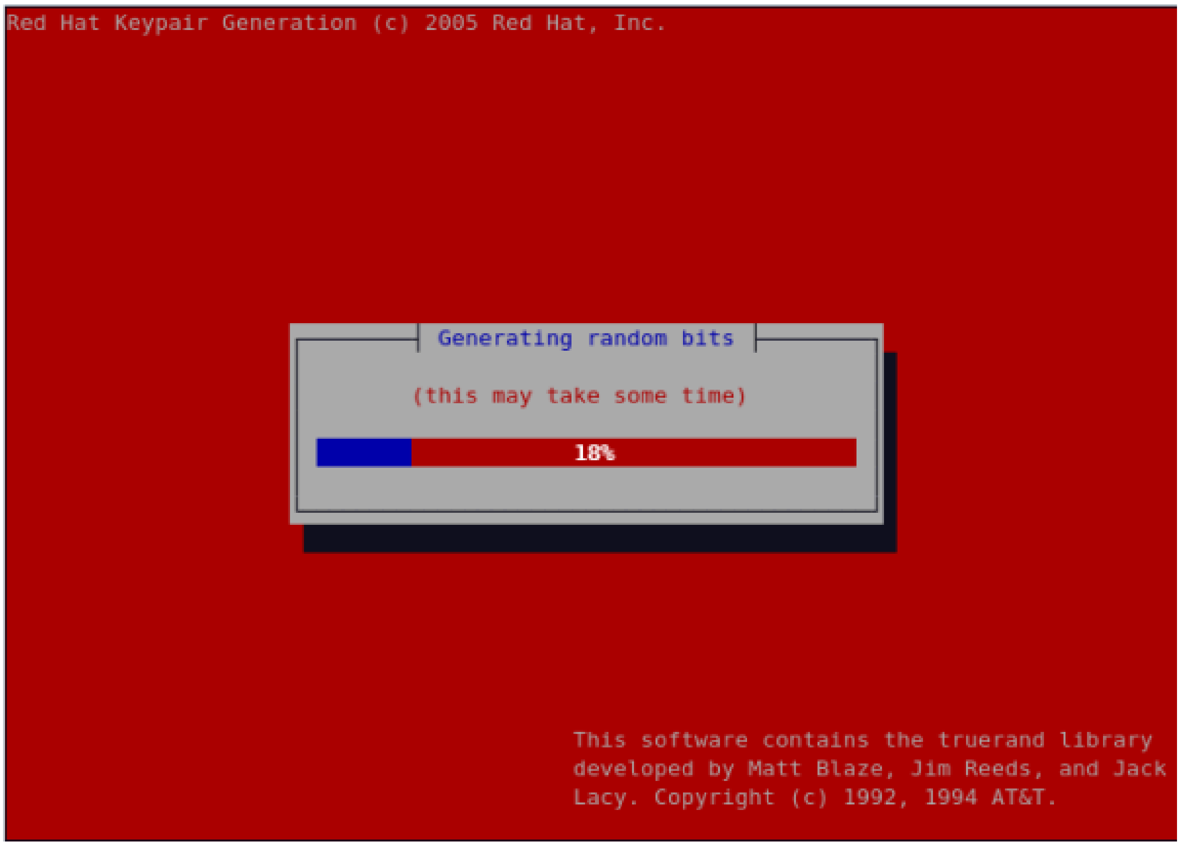


```
Red Hat Keypair Generation (c) 2005 Red Hat, Inc.

                         Choose key size

      Choose the size of your key. The smaller the key you choose the faster
      your server response will be, but you'll have less security. Keys of
      less than 1024 bits are easily cracked.  Keys greater than 1024 bits
      don't work with all currently available browsers.

      We suggest you select the default, 1024 bits

                  512 (insecure)
                  1024 (medium-grade, fast speed) [RECOMMENDED]
                  2048 (high-security, medium speed)
                  4096 (paranoid-security, tortoise speed)
                  Choose your own
```

Next
Back
Cancel
<Tab>/<Alt-Tab> between elements | <Space> selects | <Escape> to quit

Figure 11.12. Choose key size

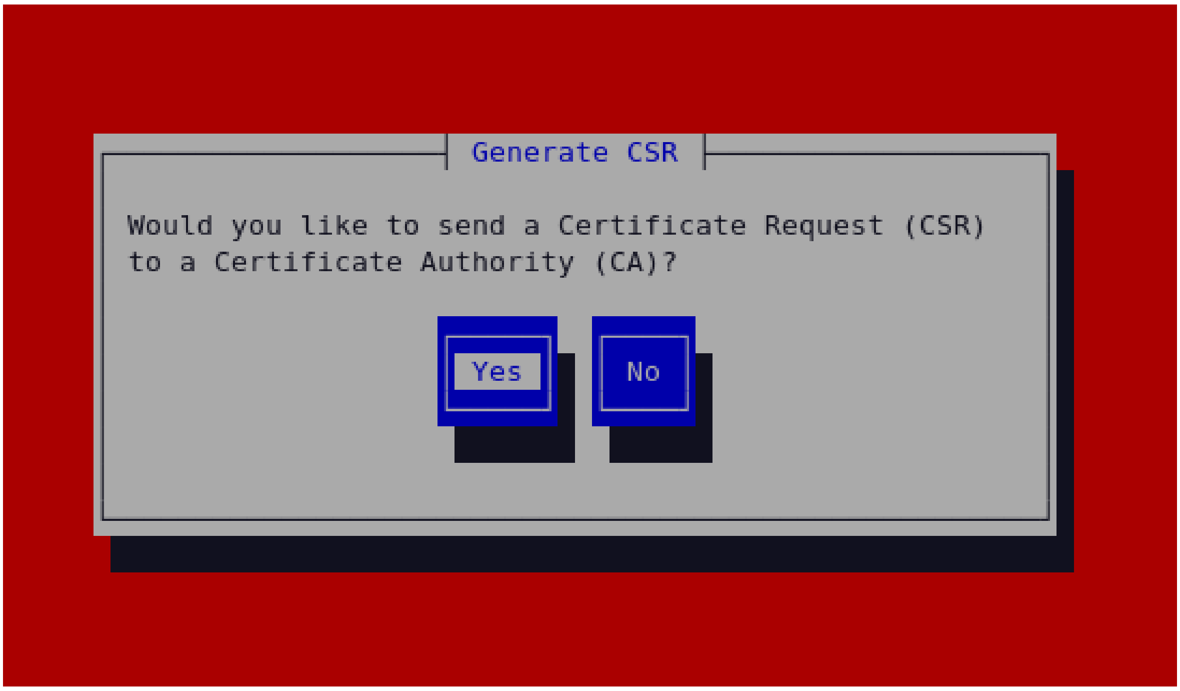
Selecting the next step will initiate the random bits generation process which may take some time depending on the size of your selected key. The larger the size of your key, the longer it will take to generate it.



```
Red Hat Keypair Generation (c) 2005 Red Hat, Inc.

Generating random bits

(this may take some time)

18%

This software contains the truerand library
developed by Matt Blaze, Jim Reeds, and Jack
Lacy. Copyright (c) 1992, 1994 AT&T.
```

Figure 11.13. Generating random bits

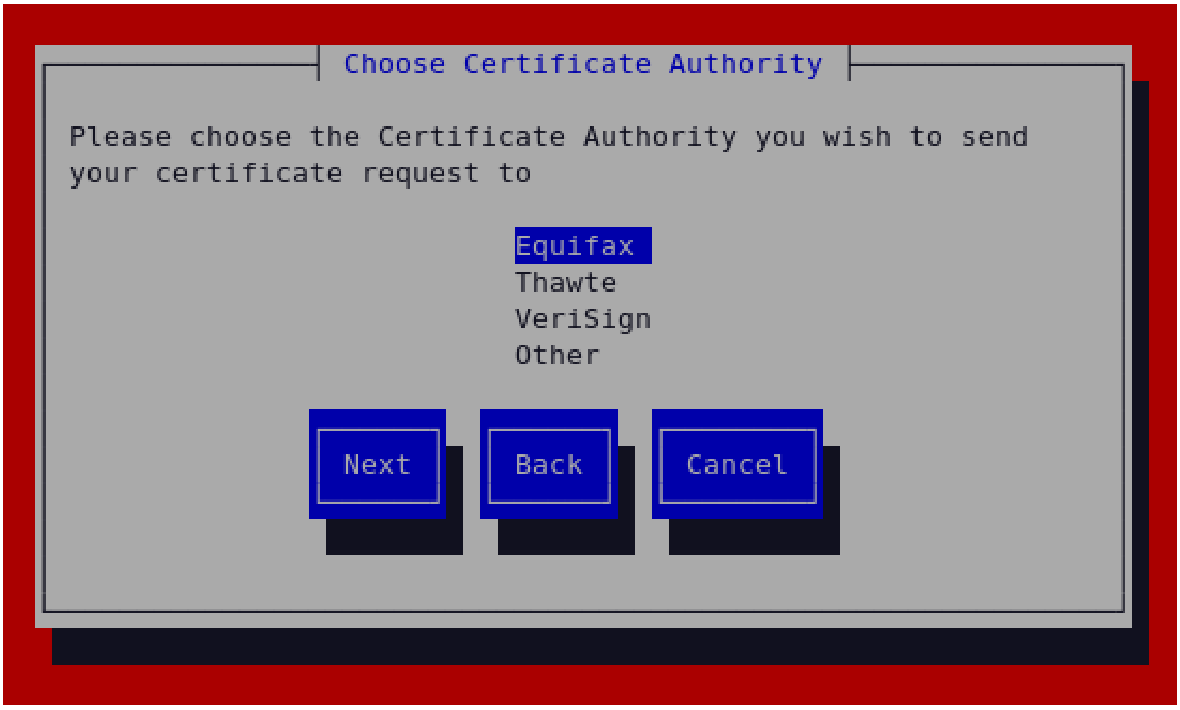
On generating your key, you will be prompted to send a Certificate Request (CSR) to a Certificate Authority (CA).



```
Generate CSR

Would you like to send a Certificate Request (CSR)
to a Certificate Authority (CA)?
```

Figure 11.14. Generate CSR

Selecting **Yes** will prompt you to select the Certificate Authority you wish to send your request to. Selecting **No** will allow you to generate a self-signed certificate. The next step for this is illustrated in *Figure 11.17, “Generating a self signed certificate for your server”*.



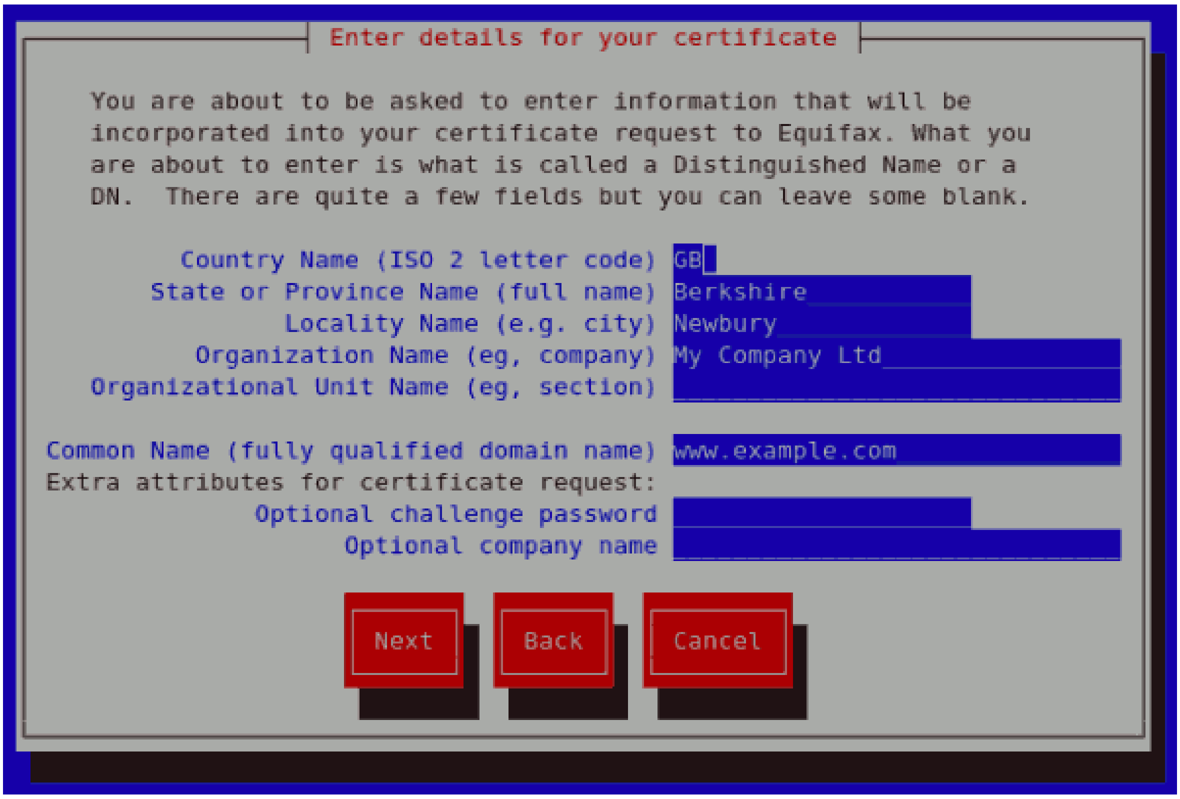
Choose Certificate Authority

Please choose the Certificate Authority you wish to send your certificate request to

Equifax
Thawte
VeriSign
Other

Figure 11.15. Choose Certificate Authority (CA)

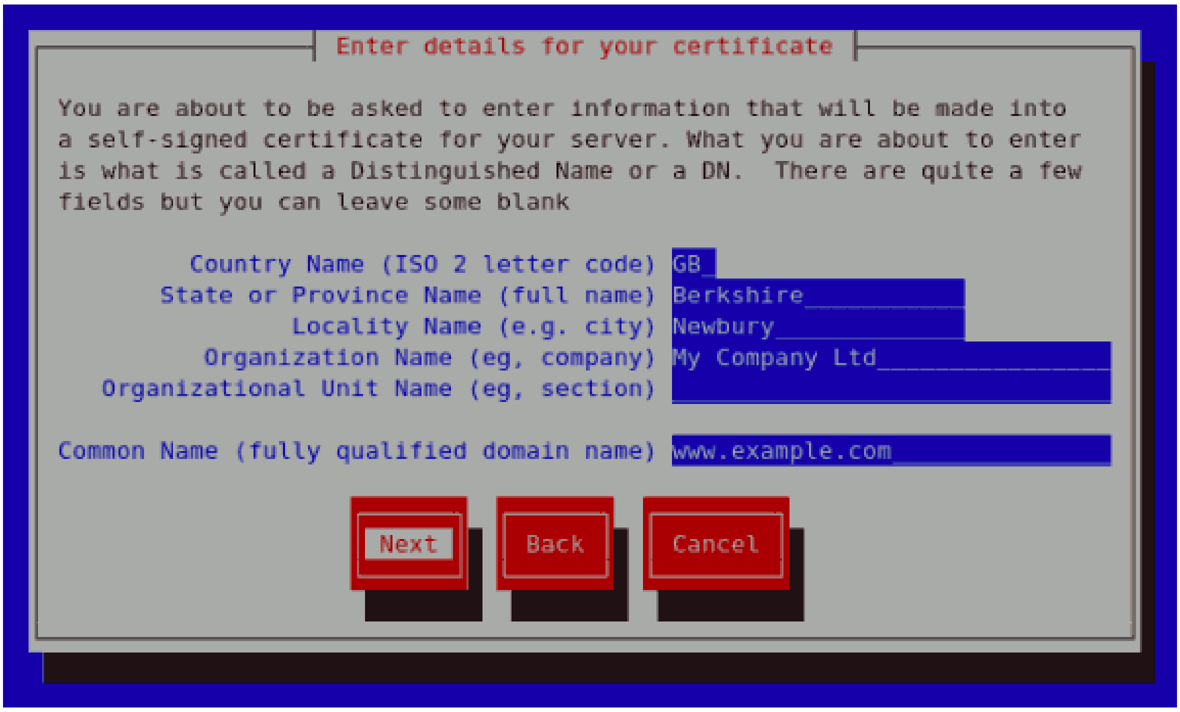
On Selecting your preferred option, select **Next** to proceed to the next step. The next screen allows you to enter the details of your certificate.



```
Enter details for your certificate

You are about to be asked to enter information that will be
incorporated into your certificate request to Equifax. What you
are about to enter is what is called a Distinguished Name or a
DN.  There are quite a few fields but you can leave some blank.

        Country Name (ISO 2 letter code) GB
      State or Province Name (full name) Berkshire
           Locality Name (e.g. city) Newbury
       Organization Name (eg, company) My Company Ltd
  Organizational Unit Name (eg, section)

Common Name (fully qualified domain name) www.example.com
Extra attributes for certificate request:
             Optional challenge password
                   Optional company name
```

Figure 11.16. Enter details for your certificate

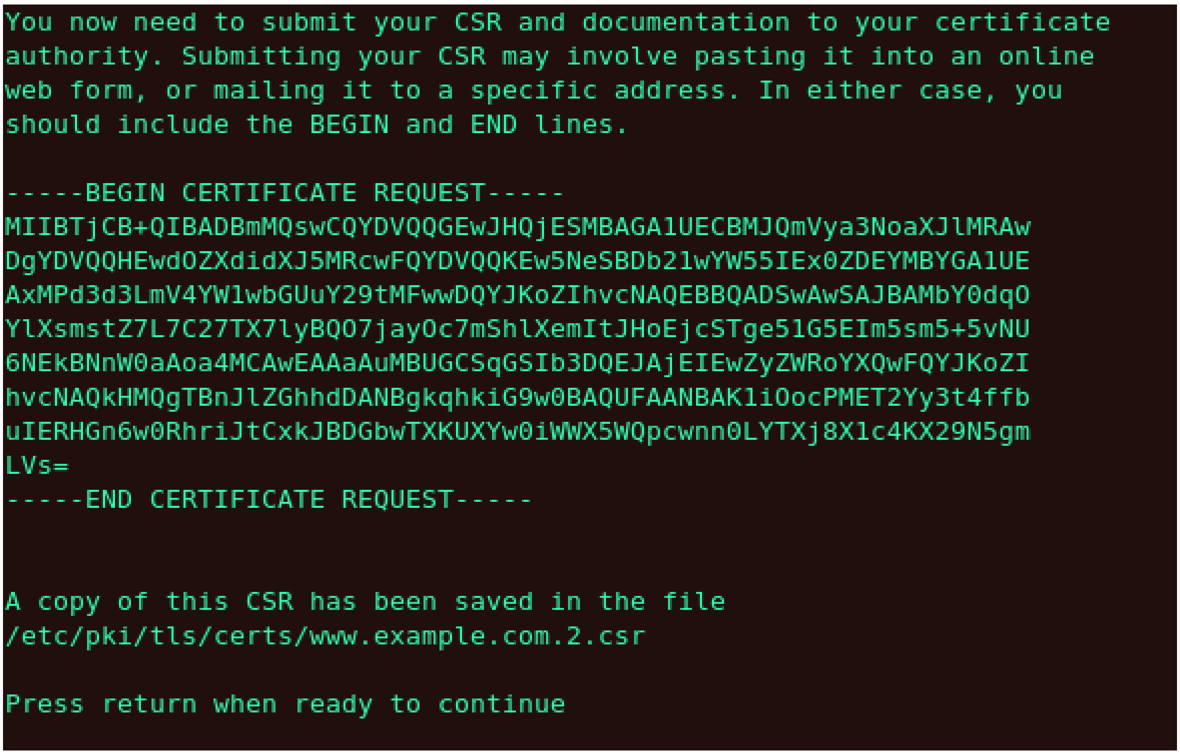
If you prefer to generate a self signed cert key pair, you should not generate a CSR. To do this, select **No** as your preferred option in the Generate CSR screen. This will display the figure below from which you can enter your certificate details. Entering your certificate details and pressing the return key will display the *Figure 11.19, “Protecting your private key”* from which you can choose to encrypt your private key or not.



```
Enter details for your certificate

You are about to be asked to enter information that will be made into
a self-signed certificate for your server. What you are about to enter
is what is called a Distinguished Name or a DN.  There are quite a few
fields but you can leave some blank

         Country Name (ISO 2 letter code) GB_
       State or Province Name (full name) Berkshire____________
              Locality Name (e.g. city) Newbury_____________
          Organization Name (eg, company) My Company Ltd_________________
  Organizational Unit Name (eg, section) ______________________________

Common Name (fully qualified domain name) www.example.com________________
```

Figure 11.17. Generating a self signed certificate for your server

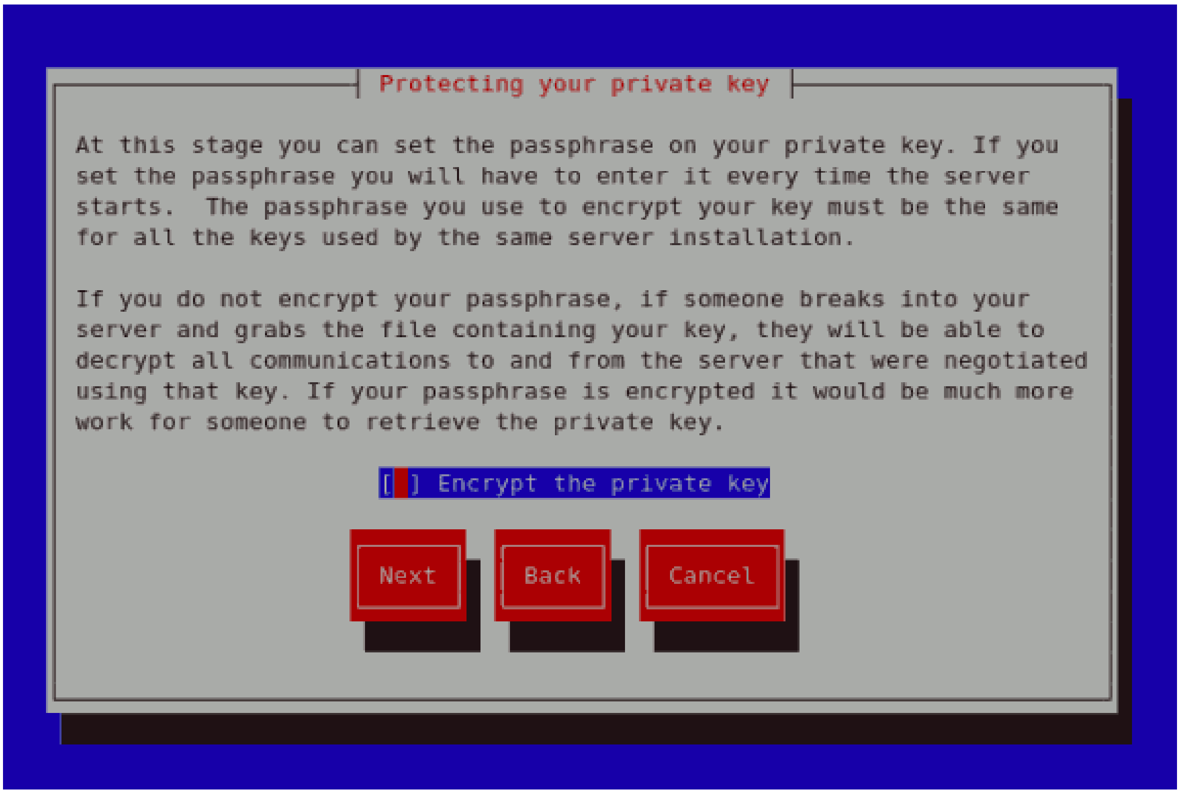
On entering the details of your certificate, select **Next** to proceed. The figure below illustrates an example of a the next screen displayed after completing the details for a certificate to be sent to Equifax. Please note that if you are generating a self signed key, for your server, this screen is not displayed.



```
You now need to submit your CSR and documentation to your certificate
authority. Submitting your CSR may involve pasting it into an online
web form, or mailing it to a specific address. In either case, you
should include the BEGIN and END lines.

-----BEGIN CERTIFICATE REQUEST-----
MIIBTjCB+QIBADBmMQswCQYDVQQGEwJHQjESMBAGA1UECBMJQmVya3NoaXJlMRAw
DgYDVQQHEwdOZXdidXJ5MRcwFQYDVQQKEw5NeSBDb21wYW55IEx0ZDEYMBYGA1UE
AxMPd3d3LmV4YW1wbGUuY29tMFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAMbY0dqO
YlXsmstZ7L7C27TX7lyBQO7jayOc7mShlXemItJHoEjcSTge51G5EIm5sm5+5vNU
6NEkBNnW0aAoa4MCAwEAAaAuMBUGCSqGSIb3DQEJAjEIEwZyZWRoYXQwFQYJKoZI
hvcNAQkHMQgTBnJlZGhhdDANBgkqhkiG9w0BAQUFAANBAK1iOocPMET2Yy3t4ffb
uIERHGn6w0RhriJtCxkJBDGbwTXKUXYw0iWWX5WQpcwnn0LYTXj8X1c4KX29N5gm
LVs=
-----END CERTIFICATE REQUEST-----


A copy of this CSR has been saved in the file
/etc/pki/tls/certs/www.example.com.2.csr

Press return when ready to continue
```

Figure 11.18. Begin certificate request

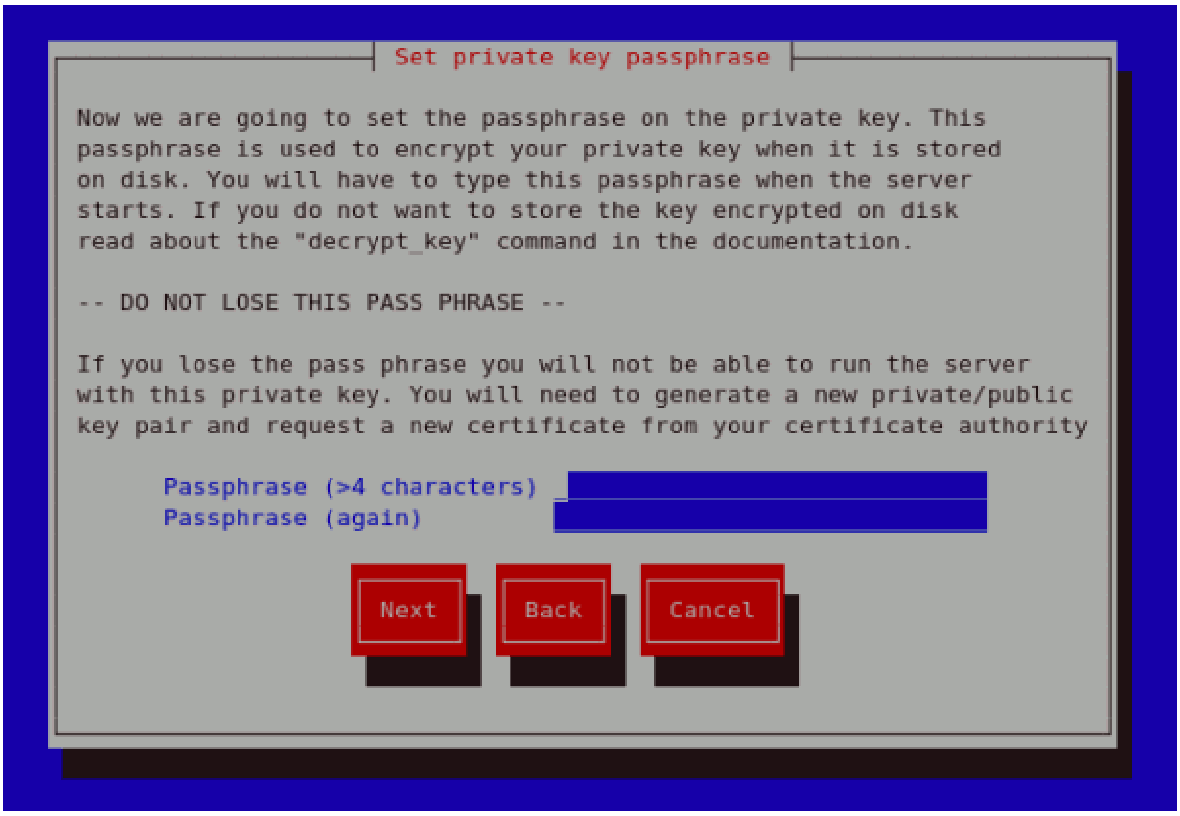
Pressing the return key, will display the next screen from which you can enable or disable the encryption of the private key. Use the spacebar to enable or disable this. When enabled, a [*] character will be displayed. On selecting your preferred option, select **Next** to proceed to the next step.



```
Protecting your private key

At this stage you can set the passphrase on your private key. If you
set the passphrase you will have to enter it every time the server
starts.  The passphrase you use to encrypt your key must be the same
for all the keys used by the same server installation.

If you do not encrypt your passphrase, if someone breaks into your
server and grabs the file containing your key, they will be able to
decrypt all communications to and from the server that were negotiated
using that key. If your passphrase is encrypted it would be much more
work for someone to retrieve the private key.

[ ] Encrypt the private key
```

Figure 11.19. Protecting your private key

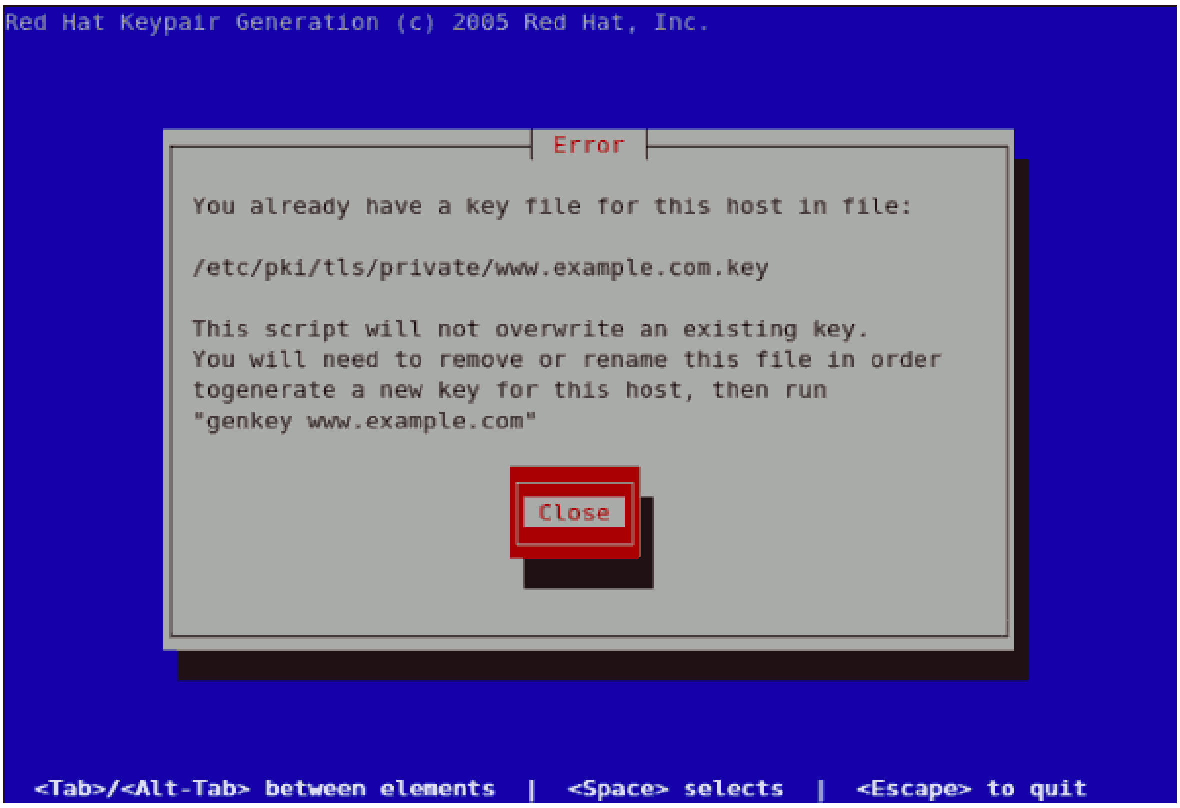
The next screen allows you to set your key passphase. Please do not lose this pass phase as you will not be able to run the server without it. You will need to regenerate a new private or public key pair and request a new certificate from your CA as indicated. For security, the passphase is not displayed as you type. On typing your preferred passphase, select **Next** to go back to your terminal.



```
Set private key passphrase

Now we are going to set the passphrase on the private key. This
passphrase is used to encrypt your private key when it is stored
on disk. You will have to type this passphrase when the server
starts. If you do not want to store the key encrypted on disk
read about the "decrypt_key" command in the documentation.

-- DO NOT LOSE THIS PASS PHRASE --

If you lose the pass phrase you will not be able to run the server
with this private key. You will need to generate a new private/public
key pair and request a new certificate from your certificate authority

      Passphrase (>4 characters) _
      Passphrase (again)
```

Figure 11.20. Set passphase

If you attempt to run **genkey makeca** on a server that has an existing key pair, an error message will be displayed as illustrated below. You need to delete your existing key file as indicated to generate a new key pair.



Red Hat Keypair Generation (c) 2005 Red Hat, Inc.

Error

You already have a key file for this host in file:

/etc/pki/tls/private/www.example.com.key

This script will not overwrite an existing key.
You will need to remove or rename this file in order
togenerate a new key for this host, then run
"genkey www.example.com"

Close
<Tab>/<Alt-Tab> between elements | <Space> selects | <Escape> to quit

Figure 11.21. genkey error

- *http://httpd.apache.org/docs/2.2/ssl/*
- *http://httpd.apache.org/docs/2.2/vhosts/*

11.8.6. How to configure the server to use the new key

The steps to configure the Apache HTTP Server to use the new key are:

- Obtain the signed certificate from the CA after submitting the CSR.
- Copy the certificate to the path, for example **`/etc/pki/tls/certs/www.example.com.crt`**
- Edit **`/etc/httpd/conf.d/ssl.conf`**. Change the SSLCertificateFile and SSLCertificateKey lines to be.

  ```
  SSLCertificateFile /etc/pki/tls/certs/www.example.com.crt
  SSLCertificateKeyFile /etc/pki/tls/private/www.example.com.key
  ```

  where the "www.example.com" part should match the argument passed on the **genkey** command.

11.9. Additional Resources

To learn more about the Apache HTTP Server, refer to the following resources.

11.9.1. Useful Websites

- *http://httpd.apache.org/* — The official website for the Apache HTTP Server with documentation on all the directives and default modules.
- *http://www.modssl.org/* — The official website for **`mod_ssl`**.
- *http://www.apacheweek.com/* — A comprehensive online weekly newsletter about all things Apache.

FTP

File Transfer Protocol (FTP) is one of the oldest and most commonly used protocols found on the Internet today. Its purpose is to reliably transfer files between computer hosts on a network without requiring the user to log directly into the remote host or have knowledge of how to use the remote system. It allows users to access files on remote systems using a standard set of simple commands.

This chapter outlines the basics of the FTP protocol, as well as configuration options for the primary FTP server shipped with Fedora, **`vsftpd`**.

12.1. The File Transfer Protocol

However, because FTP is so prevalent on the Internet, it is often required to share files to the public. System administrators, therefore, should be aware of the FTP protocol's unique characteristics.

12.1.1. Multiple Ports, Multiple Modes

Unlike most protocols used on the Internet, FTP requires multiple network ports to work properly. When an FTP client application initiates a connection to an FTP server, it opens port 21 on the server — known as the *command port*. This port is used to issue all commands to the server. Any data requested from the server is returned to the client via a *data port*. The port number for data connections, and the way in which data connections are initialized, vary depending upon whether the client requests the data in *active* or *passive* mode.

The following defines these modes:

active mode

> Active mode is the original method used by the FTP protocol for transferring data to the client application. When an active mode data transfer is initiated by the FTP client, the server opens a connection from port 20 on the server to the IP address and a random, unprivileged port (greater than 1024) specified by the client. This arrangement means that the client machine must be allowed to accept connections over any port above 1024. With the growth of insecure networks, such as the Internet, the use of firewalls to protect client machines is now prevalent. Because these client-side firewalls often deny incoming connections from active mode FTP servers, passive mode was devised.

passive mode

> Passive mode, like active mode, is initiated by the FTP client application. When requesting data from the server, the FTP client indicates it wants to access the data in passive mode and the server provides the IP address and a random, unprivileged port (greater than 1024) on the server. The client then connects to that port on the server to download the requested information.
>
> While passive mode resolves issues for client-side firewall interference with data connections, it can complicate administration of the server-side firewall. You can reduce the number of open ports on a server by limiting the range of unprivileged ports on the FTP server. This also simplifies the process of configuring firewall rules for the server. Refer to *Section 12.5.8, “Network Options”* for more about limiting passive ports.

12.2. FTP Servers

Fedora ships with two different FTP servers:

- **Red Hat Content Accelerator** — A kernel-based Web server that delivers high performance Web server and FTP services. Since speed as its primary design goal, it has limited functionality and runs only as an anonymous FTP server. For more information about configuring and administering **Red Hat Content Accelerator**, consult the documentation available online at http://www.redhat.com/docs/manuals/tux/.

- **vsftpd** — A fast, secure FTP daemon which is the preferred FTP server for Fedora. The remainder of this chapter focuses on **vsftpd**.

12.2.1. vsftpd

The Very Secure FTP Daemon (**vsftpd**) is designed from the ground up to be fast, stable, and, most importantly, secure. **vsftpd** is the only stand-alone FTP server distributed with Fedora, due to its ability to handle large numbers of connections efficiently and securely.

The security model used by **vsftpd** has three primary aspects:

- *Strong separation of privileged and non-privileged processes* — Separate processes handle different tasks, and each of these processes run with the minimal privileges required for the task.

- *Tasks requiring elevated privileges are handled by processes with the minimal privilege necessary* — By leveraging compatibilities found in the **libcap** library, tasks that usually require full root privileges can be executed more safely from a less privileged process.

- *Most processes run in a **chroot** jail* — Whenever possible, processes are change-rooted to the directory being shared; this directory is then considered a **chroot** jail. For example, if the directory **/var/ftp/** is the primary shared directory, **vsftpd** reassigns **/var/ftp/** to the new root directory, known as **/**. This disallows any potential malicious hacker activities for any directories not contained below the new root directory.

Use of these security practices has the following effect on how **vsftpd** deals with requests:

- *The parent process runs with the least privileges required* — The parent process dynamically calculates the level of privileges it requires to minimize the level of risk. Child processes handle direct interaction with the FTP clients and run with as close to no privileges as possible.

- *All operations requiring elevated privileges are handled by a small parent process* — Much like the Apache HTTP Server, **vsftpd** launches unprivileged child processes to handle incoming connections. This allows the privileged, parent process to be as small as possible and handle relatively few tasks.

- *All requests from unprivileged child processes are distrusted by the parent process* — Communication with child processes are received over a socket, and the validity of any information from child processes is checked before being acted on.

- *Most interaction with FTP clients is handled by unprivileged child processes in a **chroot** jail* — Because these child processes are unprivileged and only have access to the directory being shared, any crashed processes only allows the attacker access to the shared files.

12.3. Files Installed with vsftpd

The **vsftpd** RPM installs the daemon (**/usr/sbin/vsftpd**), its configuration and related files, as well as FTP directories onto the system. The following lists the files and directories related to **vsftpd** configuration:

- **/etc/rc.d/init.d/vsftpd** — The *initialization script* (*initscript*) used by the **/sbin/service** command to start, stop, or reload **vsftpd**. Refer to *Section 12.4, “Starting and Stopping **vsftpd**”* for more information about using this script.

- **/etc/pam.d/vsftpd** — The Pluggable Authentication Modules (PAM) configuration file for **vsftpd**. This file specifies the requirements a user must meet to login to the FTP server. For more information, refer to .

- **/etc/vsftpd/vsftpd.conf** — The configuration file for **vsftpd**. Refer to *Section 12.5, “**vsftpd** Configuration Options”* for a list of important options contained within this file.

- **/etc/vsftpd.ftpusers** — A list of users not allowed to log into **vsftpd**. By default, this list includes the **root**, **bin**, and **daemon** users, among others.

- **/etc/vsftpd.user_list** — This file can be configured to either deny or allow access to the users listed, depending on whether the **userlist_deny** directive is set to **YES** (default) or **NO** in **/etc/vsftpd/vsftpd.conf**. If **/etc/vsftpd.user_list** is used to grant access to users, the usernames listed must *not* appear in **/etc/vsftpd.ftpusers**.

- **/var/ftp/** — The directory containing files served by **vsftpd**. It also contains the **/var/ftp/pub/** directory for anonymous users. Both directories are world-readable, but writable only by the root user.

12.4. Starting and Stopping vsftpd

The **vsftpd** RPM installs the **/etc/rc.d/init.d/vsftpd** script, which can be accessed using the **/sbin/service** command.

To start the server, as root type:

```
/sbin/service vsftpd start
```

To stop the server, as root type:

```
/sbin/service vsftpd stop
```

The `restart` option is a shorthand way of stopping and then starting **vsftpd**. This is the most efficient way to make configuration changes take effect after editing the configuration file for **vsftpd**.

To restart the server, as root type:

```
/sbin/service vsftpd restart
```

The `condrestart` (*conditional restart*) option only starts **vsftpd** if it is currently running. This option is useful for scripts, because it does not start the daemon if it is not running.

To conditionally restart the server, as root type:

```
/sbin/service vsftpd condrestart
```

By default, the **`vsftpd`** service does *not* start automatically at boot time. To configure the **`vsftpd`** service to start at boot time, use an initscript utility, such as **`/sbin/chkconfig`**, **/usr/sbin/ntsysv**, or the **Services Configuration Tool** program. Refer to *Chapter 6, Controlling Access to Services* for more information regarding these tools.

12.4.1. Starting Multiple Copies of `vsftpd`

Sometimes one computer is used to serve multiple FTP domains. This is a technique called *multihoming*. One way to multihome using **`vsftpd`** is by running multiple copies of the daemon, each with its own configuration file.

To do this, first assign all relevant IP addresses to network devices or alias network devices on the system. Refer to *Chapter 5, Network Configuration* for more information about configuring network devices and device aliases. Additional information can be found about network configuration scripts in *Chapter 4, Network Interfaces*.

Next, the DNS server for the FTP domains must be configured to reference the correct machine. For information about BIND and its configuration files, refer to *Chapter 7, Berkeley Internet Name Domain (BIND)*.

For **`vsftpd`** to answer requests on different IP addresses, multiple copies of the daemon must be running. The first copy must be run using the **`vsftpd`** initscripts, as outlined in *Section 12.4, “Starting and Stopping `vsftpd`”*. This copy uses the standard configuration file, **`/etc/vsftpd/vsftpd.conf`**.

Each additional FTP site must have a configuration file with a unique name in the **`/etc/vsftpd/`** directory, such as **`/etc/vsftpd/vsftpd-site-2.conf`**. Each configuration file must be readable and writable only by root. Within each configuration file for each FTP server listening on an IPv4 network, the following directive must be unique:

```

listen_address=N.N.N.N
```

Replace *`N.N.N.N`* with the *unique* IP address for the FTP site being served. If the site is using IPv6, use the **`listen_address6`** directive instead.

Once each additional server has a configuration file, the **`vsftpd`** daemon must be launched from a root shell prompt using the following command:

```

vsftpd /etc/vsftpd/<configuration-file> [amp   ]
```

In the above command, replace *`<configuration-file>`* with the unique name for the server's configuration file, such as **`/etc/vsftpd/vsftpd-site-2.conf`**.

Other directives to consider altering on a per-server basis are:

- **`anon_root`**
- **`local_root`**
- **`vsftpd_log_file`**

- **xferlog_file**

For a detailed list of directives available within **vsftpd**'s configuration file, refer to *Section 12.5, “**vsftpd** Configuration Options”*.

To configure any additional servers to start automatically at boot time, add the above command to the end of the **/etc/rc.local** file.

12.5. vsftpd Configuration Options

Although **vsftpd** may not offer the level of customization other widely available FTP servers have, it offers enough options to fill most administrator's needs. The fact that it is not overly feature-laden limits configuration and programmatic errors.

All configuration of **vsftpd** is handled by its configuration file, **/etc/vsftpd/vsftpd.conf**. Each directive is on its own line within the file and follows the following format:

```
<directive>=<value>
```

For each directive, replace <*directive*> with a valid directive and <*value*> with a valid value.

Important

There must not be any spaces between the <*directive*>, equal symbol, and the <*value*> in a directive.

Comment lines must be preceded by a hash mark (**#**) and are ignored by the daemon.

For a complete list of all directives available, refer to the man page for **vsftpd.conf**.

Important

For an overview of ways to secure **vsftpd**, refer to .

The following is a list of some of the more important directives within **/etc/vsftpd/vsftpd.conf**. All directives not explicitly found within **vsftpd**'s configuration file are set to their default value.

12.5.1. Daemon Options

The following is a list of directives which control the overall behavior of the **vsftpd** daemon.

- **listen** — When enabled, **vsftpd** runs in stand-alone mode. Fedora sets this value to **YES**. This directive cannot be used in conjunction with the **listen_ipv6** directive.

  The default value is **NO**.

- **listen_ipv6** — When enabled, **vsftpd** runs in stand-alone mode, but listens only to IPv6 sockets. This directive cannot be used in conjunction with the **listen** directive.

  The default value is **NO**.

- **`session_support`** — When enabled, **`vsftpd`** attempts to maintain login sessions for each user through Pluggable Authentication Modules (PAM). Refer to for more information. If session logging is not necessary, disabling this option allows **`vsftpd`** to run with less processes and lower privileges.

  The default value is **`YES`**.

12.5.2. Log In Options and Access Controls

The following is a list of directives which control the login behavior and access control mechanisms.

- **`anonymous_enable`** — When enabled, anonymous users are allowed to log in. The usernames `anonymous` and `ftp` are accepted.

  The default value is **`YES`**.

  Refer to *Section 12.5.3, “Anonymous User Options”* for a list of directives affecting anonymous users.

- **`banned_email_file`** — If the **`deny_email_enable`** directive is set to **`YES`**, this directive specifies the file containing a list of anonymous email passwords which are not permitted access to the server.

  The default value is **`/etc/vsftpd.banned_emails`**.

- **`banner_file`** — Specifies the file containing text displayed when a connection is established to the server. This option overrides any text specified in the **`ftpd_banner`** directive.

  There is no default value for this directive.

- **`cmds_allowed`** — Specifies a comma-delimited list of FTP commands allowed by the server. All other commands are rejected.

  There is no default value for this directive.

- **`deny_email_enable`** — When enabled, any anonymous user utilizing email passwords specified in the **`/etc/vsftpd.banned_emails`** are denied access to the server. The name of the file referenced by this directive can be specified using the **`banned_email_file`** directive.

  The default value is **`NO`**.

- **`ftpd_banner`** — When enabled, the string specified within this directive is displayed when a connection is established to the server. This option can be overridden by the **`banner_file`** directive.

  By default **`vsftpd`** displays its standard banner.

- **`local_enable`** — When enabled, local users are allowed to log into the system.

  The default value is **`YES`**.

  Refer to *Section 12.5.4, “Local User Options”* for a list of directives affecting local users.

- **`pam_service_name`** — Specifies the PAM service name for **`vsftpd`**.

  The default value is **`ftp`**. Note, in Fedora, the value is set to **`vsftpd`**.

- The default value is **NO**. Note, in Fedora, the value is set to **YES**.

- **userlist_deny** — When used in conjunction with the **userlist_enable** directive and set to **NO**, all local users are denied access unless the username is listed in the file specified by the **userlist_file** directive. Because access is denied before the client is asked for a password, setting this directive to **NO** prevents local users from submitting unencrypted passwords over the network.

  The default value is **YES**.

- **userlist_enable** — When enabled, the users listed in the file specified by the **userlist_file** directive are denied access. Because access is denied before the client is asked for a password, users are prevented from submitting unencrypted passwords over the network.

  The default value is **NO**, however under Fedora the value is set to **YES**.

- **userlist_file** — Specifies the file referenced by **vsftpd** when the **userlist_enable** directive is enabled.

  The default value is **/etc/vsftpd.user_list** and is created during installation.

12.5.3. Anonymous User Options

The following lists directives which control anonymous user access to the server. To use these options, the **anonymous_enable** directive must be set to **YES**.

- **anon_mkdir_write_enable** — When enabled in conjunction with the **write_enable** directive, anonymous users are allowed to create new directories within a parent directory which has write permissions.

  The default value is **NO**.

- **anon_root** — Specifies the directory **vsftpd** changes to after an anonymous user logs in.

  There is no default value for this directive.

- **anon_upload_enable** — When enabled in conjunction with the **write_enable** directive, anonymous users are allowed to upload files within a parent directory which has write permissions.

  The default value is **NO**.

- **anon_world_readable_only** — When enabled, anonymous users are only allowed to download world-readable files.

  The default value is **YES**.

- **ftp_username** — Specifies the local user account (listed in **/etc/passwd**) used for the anonymous FTP user. The home directory specified in **/etc/passwd** for the user is the root directory of the anonymous FTP user.

  The default value is **ftp**.

- **no_anon_password** — When enabled, the anonymous user is not asked for a password.

  The default value is **NO**.

- **`secure_email_list_enable`** — When enabled, only a specified list of email passwords for anonymous logins are accepted. This is a convenient way to offer limited security to public content without the need for virtual users.

  Anonymous logins are prevented unless the password provided is listed in **`/etc/vsftpd.email_passwords`**. The file format is one password per line, with no trailing white spaces.

  The default value is **`NO`**.

12.5.4. Local User Options

The following lists directives which characterize the way local users access the server. To use these options, the **`local_enable`** directive must be set to **`YES`**.

- **`chmod_enable`** — When enabled, the FTP command **`SITE CHMOD`** is allowed for local users. This command allows the users to change the permissions on files.

  The default value is **`YES`**.

- **`chroot_list_enable`** — When enabled, the local users listed in the file specified in the **`chroot_list_file`** directive are placed in a **`chroot`** jail upon log in.

  If enabled in conjunction with the **`chroot_local_user`** directive, the local users listed in the file specified in the **`chroot_list_file`** directive are *not* placed in a **`chroot`** jail upon log in.

  The default value is **`NO`**.

- **`chroot_list_file`** — Specifies the file containing a list of local users referenced when the **`chroot_list_enable`** directive is set to **`YES`**.

  The default value is **`/etc/vsftpd.chroot_list`**.

- **`chroot_local_user`** — When enabled, local users are change-rooted to their home directories after logging in.

  The default value is **`NO`**.

  > **Warning**
  >
  > Enabling **`chroot_local_user`** opens up a number of security issues, especially for users with upload privileges. For this reason, it is *not* recommended.

- **`guest_enable`** — When enabled, all non-anonymous users are logged in as the user **`guest`**, which is the local user specified in the **`guest_username`** directive.

  The default value is **`NO`**.

- **`guest_username`** — Specifies the username the **`guest`** user is mapped to.

  The default value is **`ftp`**.

- **`local_root`** — Specifies the directory **`vsftpd`** changes to after a local user logs in.

There is no default value for this directive.

- **`local_umask`** — Specifies the umask value for file creation. Note that the default value is in octal form (a numerical system with a base of eight), which includes a "0" prefix. Otherwise the value is treated as a base-10 integer.

  The default value is **`022`**.

- **`passwd_chroot_enable`** — When enabled in conjunction with the **`chroot_local_user`** directive, **`vsftpd`** change-roots local users based on the occurrence of the **`/./`** in the home directory field within **`/etc/passwd`**.

  The default value is **`NO`**.

- **`user_config_dir`** — Specifies the path to a directory containing configuration files bearing the name of local system users that contain specific setting for that user. Any directive in the user's configuration file overrides those found in **`/etc/vsftpd/vsftpd.conf`**.

  There is no default value for this directive.

12.5.5. Directory Options

The following lists directives which affect directories.

- **`dirlist_enable`** — When enabled, users are allowed to view directory lists.

  The default value is **`YES`**.

- **`dirmessage_enable`** — When enabled, a message is displayed whenever a user enters a directory with a message file. This message resides within the current directory. The name of this file is specified in the **`message_file`** directive and is **`.message`** by default.

  The default value is **`NO`**. Note, in Fedora, the value is set to **`YES`**.

- **`force_dot_files`** — When enabled, files beginning with a dot (`.`) are listed in directory listings, with the exception of the `.` and `..` files.

  The default value is **`NO`**.

- **`hide_ids`** — When enabled, all directory listings show `ftp` as the user and group for each file.

  The default value is **`NO`**.

- **`message_file`** — Specifies the name of the message file when using the **`dirmessage_enable`** directive.

  The default value is **`.message`**.

- **`text_userdb_names`** — When enabled, test usernames and group names are used in place of UID and GID entries. Enabling this option may slow performance of the server.

  The default value is **`NO`**.

- **`use_localtime`** — When enabled, directory listings reveal the local time for the computer instead of GMT.

The default value is **NO**.

12.5.6. File Transfer Options

The following lists directives which affect directories.

- **download_enable** — When enabled, file downloads are permitted.

  The default value is **YES**.

- **chown_uploads** — When enabled, all files uploaded by anonymous users are owned by the user specified in the **chown_username** directive.

  The default value is **NO**.

- **chown_username** — Specifies the ownership of anonymously uploaded files if the **chown_uploads** directive is enabled.

  The default value is **root**.

- **write_enable** — When enabled, FTP commands which can change the file system are allowed, such as **DELE**, **RNFR**, and **STOR**.

  The default value is **YES**.

12.5.7. Logging Options

The following lists directives which affect **vsftpd**'s logging behavior.

- **dual_log_enable** — When enabled in conjunction with **xferlog_enable**, **vsftpd** writes two files simultaneously: a **wu-ftpd**-compatible log to the file specified in the **xferlog_file** directive (**/var/log/xferlog** by default) and a standard **vsftpd** log file specified in the **vsftpd_log_file** directive (**/var/log/vsftpd.log** by default).

  The default value is **NO**.

- **log_ftp_protocol** — When enabled in conjunction with **xferlog_enable** and with **xferlog_std_format** set to **NO**, all FTP commands and responses are logged. This directive is useful for debugging.

  The default value is **NO**.

- **syslog_enable** — When enabled in conjunction with **xferlog_enable**, all logging normally written to the standard **vsftpd** log file specified in the **vsftpd_log_file** directive (**/var/log/vsftpd.log** by default) is sent to the system logger instead under the FTPD facility.

  The default value is **NO**.

- **vsftpd_log_file** — Specifies the **vsftpd** log file. For this file to be used, **xferlog_enable** must be enabled and **xferlog_std_format** must either be set to **NO** or, if **xferlog_std_format** is set to **YES**, **dual_log_enable** must be enabled. It is important to note that if **syslog_enable** is set to **YES**, the system log is used instead of the file specified in this directive.

  The default value is **/var/log/vsftpd.log**.

- **`xferlog_enable`** — When enabled, **`vsftpd`** logs connections (**`vsftpd`** format only) and file transfer information to the log file specified in the **`vsftpd_log_file`** directive (**`/var/log/vsftpd.log`** by default). If **`xferlog_std_format`** is set to **`YES`**, file transfer information is logged but connections are not, and the log file specified in **`xferlog_file`** (**`/var/log/xferlog`** by default) is used instead. It is important to note that both log files and log formats are used if **`dual_log_enable`** is set to **`YES`**.

  The default value is **`NO`**. Note, in Fedora, the value is set to **`YES`**.

- **`xferlog_file`** — Specifies the **`wu-ftpd`**-compatible log file. For this file to be used, **`xferlog_enable`** must be enabled and **`xferlog_std_format`** must be set to **`YES`**. It is also used if **`dual_log_enable`** is set to **`YES`**.

  The default value is **`/var/log/xferlog`**.

- **`xferlog_std_format`** — When enabled in conjunction with **`xferlog_enable`**, only a **`wu-ftpd`**-compatible file transfer log is written to the file specified in the **`xferlog_file`** directive (**`/var/log/xferlog`** by default). It is important to note that this file only logs file transfers and does not log connections to the server.

  The default value is **`NO`**. Note, in Fedora, the value is set to **`YES`**.

Important

To maintain compatibility with log files written by the older **`wu-ftpd`** FTP server, the **`xferlog_std_format`** directive is set to **`YES`** under Fedora. However, this setting means that connections to the server are not logged.

To both log connections in **`vsftpd`** format and maintain a **`wu-ftpd`**-compatible file transfer log, set **`dual_log_enable`** to **`YES`**.

If maintaining a **`wu-ftpd`**-compatible file transfer log is not important, either set **`xferlog_std_format`** to **`NO`**, comment the line with a hash mark (#), or delete the line entirely.

12.5.8. Network Options

The following lists directives which affect how **`vsftpd`** interacts with the network.

- **`accept_timeout`** — Specifies the amount of time for a client using passive mode to establish a connection.

  The default value is **`60`**.

- **`anon_max_rate`** — Specifies the maximum data transfer rate for anonymous users in bytes per second.

  The default value is **`0`**, which does not limit the transfer rate.

- **`connect_from_port_20`** When enabled, **`vsftpd`** runs with enough privileges to open port 20 on the server during active mode data transfers. Disabling this option allows **`vsftpd`** to run with less privileges, but may be incompatible with some FTP clients.

The default value is **NO**. Note, in Fedora, the value is set to **YES**.

- **connect_timeout** — Specifies the maximum amount of time a client using active mode has to respond to a data connection, in seconds.

  The default value is **60**.

- **data_connection_timeout** — Specifies maximum amount of time data transfers are allowed to stall, in seconds. Once triggered, the connection to the remote client is closed.

  The default value is **300**.

- **ftp_data_port** — Specifies the port used for active data connections when **connect_from_port_20** is set to **YES**.

  The default value is **20**.

- **idle_session_timeout** — Specifies the maximum amount of time between commands from a remote client. Once triggered, the connection to the remote client is closed.

  The default value is **300**.

- **listen_address** — Specifies the IP address on which **vsftpd** listens for network connections.

  There is no default value for this directive.

  > **Tip**
  >
  > If running multiple copies of **vsftpd** serving different IP addresses, the configuration file for each copy of the **vsftpd** daemon must have a different value for this directive. Refer to *Section 12.4.1, “Starting Multiple Copies of **vsftpd**”* for more information about multihomed FTP servers.

- **listen_address6** — Specifies the IPv6 address on which **vsftpd** listens for network connections when **listen_ipv6** is set to **YES**.

  There is no default value for this directive.

  > **Tip**
  >
  > If running multiple copies of **vsftpd** serving different IP addresses, the configuration file for each copy of the **vsftpd** daemon must have a different value for this directive. Refer to *Section 12.4.1, “Starting Multiple Copies of **vsftpd**”* for more information about multihomed FTP servers.

- **listen_port** — Specifies the port on which **vsftpd** listens for network connections.

  The default value is **21**.

- **local_max_rate** — Specifies the maximum rate data is transferred for local users logged into the server in bytes per second.

The default value is **0**, which does not limit the transfer rate.

- **max_clients** — Specifies the maximum number of simultaneous clients allowed to connect to the server when it is running in standalone mode. Any additional client connections would result in an error message.

  The default value is **0**, which does not limit connections.

- **max_per_ip** — Specifies the maximum of clients allowed to connected from the same source IP address.

  The default value is **0**, which does not limit connections.

- **pasv_address** — Specifies the IP address for the public facing IP address of the server for servers behind Network Address Translation (NAT) firewalls. This enables **vsftpd** to hand out the correct return address for passive mode connections.

  There is no default value for this directive.

- **pasv_enable** — When enabled, passive mode connects are allowed.

  The default value is **YES**.

- **pasv_max_port** — Specifies the highest possible port sent to the FTP clients for passive mode connections. This setting is used to limit the port range so that firewall rules are easier to create.

  The default value is **0**, which does not limit the highest passive port range. The value must not exceed **65535**.

- **pasv_min_port** — Specifies the lowest possible port sent to the FTP clients for passive mode connections. This setting is used to limit the port range so that firewall rules are easier to create.

  The default value is **0**, which does not limit the lowest passive port range. The value must not be lower **1024**.

- **pasv_promiscuous** — When enabled, data connections are not checked to make sure they are originating from the same IP address. This setting is only useful for certain types of tunneling.

  Caution

  Do not enable this option unless absolutely necessary as it disables an important security feature which verifies that passive mode connections originate from the same IP address as the control connection that initiates the data transfer.

  The default value is **NO**.

- **port_enable** — When enabled, active mode connects are allowed.

  The default value is **YES**.

12.6. Additional Resources

For more information about **vsftpd**, refer to the following resources.

12.6.1. Installed Documentation

- The **/usr/share/doc/vsftpd-*<version-number>*/** directory — Replace *<version-number>* with the installed version of the **vsftpd** package. This directory contains a **README** with basic information about the software. The **TUNING** file contains basic performance tuning tips and the **SECURITY/** directory contains information about the security model employed by **vsftpd**.

- **vsftpd** related man pages — There are a number of man pages for the daemon and configuration files. The following lists some of the more important man pages.

  Server Applications
  - **man vsftpd** — Describes available command line options for **vsftpd**.

  Configuration Files
  - **man vsftpd.conf** — Contains a detailed list of options available within the configuration file for **vsftpd**.

  - **man 5 hosts_access** — Describes the format and options available within the TCP wrappers configuration files: **hosts.allow** and **hosts.deny**.

12.6.2. Useful Websites

- *http://vsftpd.beasts.org/* — The **vsftpd** project page is a great place to locate the latest documentation and to contact the author of the software.

- *http://slacksite.com/other/ftp.html* — This website provides a concise explanation of the differences between active and passive mode FTP.

- *http://www.ietf.org/rfc/rfc0959.txt* — The original *Request for Comments* (*RFC*) of the FTP protocol from the IETF.

Email

The birth of electronic mail (*email*) occurred in the early 1960s. The mailbox was a file in a user's home directory that was readable only by that user. Primitive mail applications appended new text messages to the bottom of the file, making the user wade through the constantly growing file to find any particular message. This system was only capable of sending messages to users on the same system.

The first network transfer of an electronic mail message file took place in 1971 when a computer engineer named Ray Tomlinson sent a test message between two machines via ARPANET — the precursor to the Internet. Communication via email soon became very popular, comprising 75 percent of ARPANET's traffic in less than two years.

Today, email systems based on standardized network protocols have evolved into some of the most widely used services on the Internet. Fedora offers many advanced applications to serve and access email.

This chapter reviews modern email protocols in use today and some of the programs designed to send and receive email.

13.1. Email Protocols

Today, email is delivered using a client/server architecture. An email message is created using a mail client program. This program then sends the message to a server. The server then forwards the message to the recipient's email server, where the message is then supplied to the recipient's email client.

To enable this process, a variety of standard network protocols allow different machines, often running different operating systems and using different email programs, to send and receive email.

The following protocols discussed are the most commonly used in the transfer of email.

13.1.1. Mail Transport Protocols

Mail delivery from a client application to the server, and from an originating server to the destination server, is handled by the *Simple Mail Transfer Protocol* (*SMTP*).

13.1.1.1. SMTP

The primary purpose of SMTP is to transfer email between mail servers. However, it is critical for email clients as well. To send email, the client sends the message to an outgoing mail server, which in turn contacts the destination mail server for delivery. For this reason, it is necessary to specify an SMTP server when configuring an email client.

Under Fedora, a user can configure an SMTP server on the local machine to handle mail delivery. However, it is also possible to configure remote SMTP servers for outgoing mail.

One important point to make about the SMTP protocol is that it does not require authentication. This allows anyone on the Internet to send email to anyone else or even to large groups of people. It is this characteristic of SMTP that makes junk email or *spam* possible. Imposing relay restrictions limits random users on the Internet from sending email through your SMTP server, to other servers on the internet. Servers that do not impose such restrictions are called *open relay* servers.

Fedora provides the Postfix, Sendmail and Exim SMTP programs.

13.1.2. Mail Access Protocols

There are two primary protocols used by email client applications to retrieve email from mail servers: the *Post Office Protocol* (*POP*) and the *Internet Message Access Protocol* (*IMAP*).

13.1.2.1. POP

The default POP server under Fedora is **`/usr/lib/cyrus-imapd/pop3d`** and is provided by the **`cyrus-imapd`** package. When using a POP server, email messages are downloaded by email client applications. By default, most POP email clients are automatically configured to delete the message on the email server after it has been successfully transferred, however this setting usually can be changed.

POP is fully compatible with important Internet messaging standards, such as *Multipurpose Internet Mail Extensions* (*MIME*), which allow for email attachments.

POP works best for users who have one system on which to read email. It also works well for users who do not have a persistent connection to the Internet or the network containing the mail server. Unfortunately for those with slow network connections, POP requires client programs upon authentication to download the entire content of each message. This can take a long time if any messages have large attachments.

The most current version of the standard POP protocol is POP3.

There are, however, a variety of lesser-used POP protocol variants:

- *APOP* — POP3 with MDS authentication. An encoded hash of the user's password is sent from the email client to the server rather then sending an unencrypted password.
- *KPOP* — POP3 with Kerberos authentication. Refer to for more information.
- *RPOP* — POP3 with RPOP authentication. This uses a per-user ID, similar to a password, to authenticate POP requests. However, this ID is not encrypted, so RPOP is no more secure than standard POP.

For added security, it is possible to use *Secure Socket Layer* (*SSL*) encryption for client authentication and data transfer sessions. This can be enabled by using the **`ipop3s`** service or by using the **`/usr/sbin/stunnel`** program. Refer to *Section 13.6.1, “Securing Communication”* for more information.

13.1.2.2. IMAP

The default IMAP server under Fedora is **`/usr/lib/cyrus-imapd/imapd`** and is provided by the **`cyrus-imapd`** package. When using an IMAP mail server, email messages remain on the server where users can read or delete them. IMAP also allows client applications to create, rename, or delete mail directories on the server to organize and store email.

IMAP is particularly useful for those who access their email using multiple machines. The protocol is also convenient for users connecting to the mail server via a slow connection, because only the email header information is downloaded for messages until opened, saving bandwidth. The user also has the ability to delete messages without viewing or downloading them.

For convenience, IMAP client applications are capable of caching copies of messages locally, so the user can browse previously read messages when not directly connected to the IMAP server.

IMAP, like POP, is fully compatible with important Internet messaging standards, such as MIME, which allow for email attachments.

For added security, it is possible to use *SSL* encryption for client authentication and data transfer sessions. This can be enabled by using the **`imaps`** service, or by using the **`/usr/sbin/stunnel`** program. Refer to *Section 13.6.1, "Securing Communication"* for more information.

Other free, as well as commercial, IMAP clients and servers are available, many of which extend the IMAP protocol and provide additional functionality. A comprehensive list can be found online at *http://www.imap.org/products/longlist.htm*.

13.1.2.3. Dovecot

The **`imap-login`** and **`pop3-login`** daemons which implement the IMAP and POP3 protocols are included in the **`dovecot`** package. The use of IMAP and POP is configured through **`dovecot`**; by default **`dovecot`** runs only IMAP. To configure **`dovecot`** to use POP:

1. Edit **`/etc/dovecot.conf`** to have the line:

   ```
   protocols = imap imaps pop3 pop3s
   ```

2. Make that change operational for the current session by running the command:

   ```
   /sbin/service dovecot restart
   ```

3. Make that change operational after the next reboot by running the command:

   ```
   chkconfig dovecot on
   ```

   Please note that **`dovecot`** only reports that it started the IMAP server, but also starts the POP3 server.

Unlike SMTP, both of these protocols require connecting clients to authenticate using a username and password. By default, passwords for both protocols are passed over the network unencrypted.

To configure SSL on dovecot:

- Edit the **`dovecot`** configuration file **`/etc/pki/dovecot/dovecot-openssl.conf`** as you prefer. However in a typical installation, this file does not require modification.

- Rename, move or delete the files **`/etc/pki/dovecot/certs/dovecot.pem`** and **`/etc/pki/dovecot/private/dovecot.pem`**.

- Execute the **`/usr/share/doc/dovecot-1.0/examples/mkcert.sh`** script which creates the dovecot self signed certificates. The certificates are copied in the **`/etc/pki/dovecot/certs`** and **`/etc/pki/dovecot/private`** directories. To implement the changes, restart **`dovecot`** (**`/sbin/service dovecot restart`**).

More details on **`dovecot`** can be found online at *http://www.dovecot.org*.

13.2. Email Program Classifications

In general, all email applications fall into at least one of three classifications. Each classification plays a specific role in the process of moving and managing email messages. While most users are only aware of the specific email program they use to receive and send messages, each one is important for ensuring that email arrives at the correct destination.

13.2.1. Mail Transport Agent

A *Mail Transport Agent* (*MTA*) transports email messages between hosts using SMTP. A message may involve several MTAs as it moves to its intended destination.

While the delivery of messages between machines may seem rather straightforward, the entire process of deciding if a particular MTA can or should accept a message for delivery is quite complicated. In addition, due to problems from spam, use of a particular MTA is usually restricted by the MTA's configuration or the access configuration for the network on which the MTA resides.

Many modern email client programs can act as an MTA when sending email. However, this action should not be confused with the role of a true MTA. The sole reason email client programs are capable of sending email like an MTA is because the host running the application does not have its own MTA. This is particularly true for email client programs on non-UNIX-based operating systems. However, these client programs only send outbound messages to an MTA they are authorized to use and do not directly deliver the message to the intended recipient's email server.

Since Fedora installs three MTAs—Sendmail, Postfix and Exim—email client programs are often not required to act as an MTA. Fedora also includes a special purpose MTA called Fetchmail.

For more information on Sendmail, Postfix, and Fetchmail, refer to *Section 13.3, “Mail Transport Agents”*.

13.2.2. Mail Delivery Agent

A *Mail Delivery Agent* (*MDA*) is invoked by the MTA to file incoming email in the proper user's mailbox. In many cases, the MDA is actually a *Local Delivery Agent* (*LDA*), such as **`mail`** or Procmail.

Any program that actually handles a message for delivery to the point where it can be read by an email client application can be considered an MDA. For this reason, some MTAs (such as Sendmail and Postfix) can fill the role of an MDA when they append new email messages to a local user's mail spool file. In general, MDAs do not transport messages between systems nor do they provide a user interface; MDAs distribute and sort messages on the local machine for an email client application to access.

13.2.3. Mail User Agent

A *Mail User Agent* (*MUA*) is synonymous with an email client application. An MUA is a program that, at the very least, allows a user to read and compose email messages. Many MUAs are capable of retrieving messages via the POP or IMAP protocols, setting up mailboxes to store messages, and sending outbound messages to an MTA.

MUAs may be graphical, such as Evolution, or have simple text-based interfaces, such as **`pine`**.

13.3. Mail Transport Agents

Fedora includes three primary MTAs: Sendmail, Postfix and Exim. Sendmail is configured as the default MTA, although it is easy to switch the default MTA to Postfix or Exim.

13.3.1. Sendmail

Sendmail's core purpose, like other MTAs, is to safely transfer email among hosts, usually using the SMTP protocol. However, Sendmail is highly configurable, allowing control over almost every aspect of how email is handled, including the protocol used. Many system administrators elect to use Sendmail as their MTA due to its power and scalability.

13.3.1.1. Purpose and Limitations

It is important to be aware of what Sendmail is and what it can do, as opposed to what it is not. In these days of monolithic applications that fulfill multiple roles, Sendmail may seem like the only application needed to run an email server within an organization. Technically, this is true, as Sendmail can spool mail to each users' directory and deliver outbound mail for users. However, most users actually require much more than simple email delivery. Users usually want to interact with their email using an MUA, that uses POP or IMAP, to download their messages to their local machine. Or, they may prefer a Web interface to gain access to their mailbox. These other applications can work in conjunction with Sendmail, but they actually exist for different reasons and can operate separately from one another.

It is beyond the scope of this section to go into all that Sendmail should or could be configured to do. With literally hundreds of different options and rule sets, entire volumes have been dedicated to helping explain everything that can be done and how to fix things that go wrong. Refer to the *Section 13.7, “Additional Resources”* for a list of Sendmail resources.

This section reviews the files installed with Sendmail by default and reviews basic configuration changes, including how to stop unwanted email (spam) and how to extend Sendmail with the *Lightweight Directory Access Protocol (LDAP)*.

13.3.1.2. The Default Sendmail Installation

The Sendmail executable is **`/usr/sbin/sendmail`**.

Sendmail's lengthy and detailed configuration file is **`/etc/mail/sendmail.cf`**. Avoid editing the **`sendmail.cf`** file directly. To make configuration changes to Sendmail, edit the **`/etc/mail/sendmail.mc`** file, back up the original **`/etc/mail/sendmail.cf`**, and use the following alternatives to generate a new configuration file:

- Use the included makefile in **`/etc/mail`** (**`make all -C /etc/mail`**) to create a new **`/etc/mail/sendmail.cf`** configuration file. All other generated files in **`/etc/mail`** (db files) will be regenerated if needed. The old makemap commands are still usable. The make command will automatically be used by **`service sendmail start | restart | reload`** if the **`make`** package is installed.

- Alternatively you may use the included **`m4`** macro processor to create a new **`/etc/mail/sendmail.cf`**.

More information on configuring Sendmail can be found in *Section 13.3.1.3, “Common Sendmail Configuration Changes”*.

Various Sendmail configuration files are installed in the **`/etc/mail/`** directory including:

- **`access`** — Specifies which systems can use Sendmail for outbound email.

- **`domaintable`** — Specifies domain name mapping.

- **`local-host-names`** — Specifies aliases for the host.

- **`mailertable`** — Specifies instructions that override routing for particular domains.

- **`virtusertable`** — Specifies a domain-specific form of aliasing, allowing multiple virtual domains to be hosted on one machine.

Several of the configuration files in **`/etc/mail/`**, such as **`access`**, **`domaintable`**, **`mailertable`** and **`virtusertable`**, must actually store their information in database files before Sendmail can use any configuration changes. To include any changes made to these configurations in their database files, run the following command:

```
makemap hash /etc/mail/<name> < /etc/mail/<name>
```

where <*name*> is replaced with the name of the configuration file to convert.

For example, to have all emails addressed to the **`example.com`** domain delivered to *`bob@other-example.com`* , add the following line to the **`virtusertable`** file:

```
@example.com bob@other-example.com
```

To finalize the change, the **`virtusertable.db`** file must be updated using the following command as root:

```
makemap hash /etc/mail/virtusertable < /etc/mail/virtusertable
```

This creates an updated **`virtusertable.db`** file containing the new configuration.

13.3.1.3. Common Sendmail Configuration Changes

When altering the Sendmail configuration file, it is best not to edit an existing file, but to generate an entirely new **`/etc/mail/sendmail.cf`** file.

Caution

Before changing the **`sendmail.cf`** file, it is a good idea to create a backup copy.

To add the desired functionality to Sendmail, edit the **`/etc/mail/sendmail.mc`** file as the root user. When finished, use the **`m4`** macro processor to generate a new **`sendmail.cf`** by executing the following command:

```
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```

By default, the **`m4`** macro processor is installed with Sendmail but is part of the **`m4`** package.

After creating a new **`/etc/mail/sendmail.cf`** file, restart Sendmail for the changes to take effect. The easiest way to do this is to type the following command:

```
/sbin/service sendmail restart
```

Important

The default **sendmail.cf** file does not allow Sendmail to accept network connections from any host other than the local computer. To configure Sendmail as a server for other clients, edit the **/etc/mail/sendmail.mc** file, and either change the address specified in the **Addr=** option of the **DAEMON_OPTIONS** directive from **127.0.0.1** to the IP address of an active network device or comment out the **DAEMON_OPTIONS** directive all together by placing **dnl** at the beginning of the line. When finished, regenerate **/etc/mail/sendmail.cf** by executing the following command:

```
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```

The default configuration which ships with Fedora works for most SMTP-only sites. However, it does not work for UUCP (UNIX to UNIX Copy) sites. If using UUCP mail transfers, the **/etc/mail/sendmail.mc** file must be reconfigured and a new **/etc/mail/sendmail.cf** must be generated.

Consult the **/usr/share/sendmail-cf/README** file before editing any files in the directories under the **/usr/share/sendmail-cf** directory, as they can affect the future configuration of **/etc/mail/sendmail.cf** files.

13.3.1.4. Masquerading

One common Sendmail configuration is to have a single machine act as a mail gateway for all machines on the network. For instance, a company may want to have a machine called `mail.example.com` that handles all of their email and assigns a consistent return address to all outgoing mail.

In this situation, the Sendmail server must masquerade the machine names on the company network so that their return address is `user@example.com` instead of `user@host.example.com`.

To do this, add the following lines to **/etc/mail/sendmail.mc**:

```
FEATURE(always_add_domain)dnl
FEATURE(`masquerade_entire_domain')dnl
FEATURE(`masquerade_envelope')dnl
FEATURE(`allmasquerade')dnl
MASQUERADE_AS(`bigcorp.com.')dnl
MASQUERADE_DOMAIN(`bigcorp.com.')dnl
MASQUERADE_AS(bigcorp.com)dnl
```

After generating a new **sendmail.cf** using **m4**, this configuration makes all mail from inside the network appear as if it were sent from `bigcorp.com`.

13.3.1.5. Stopping Spam

Email spam can be defined as unnecessary and unwanted email received by a user who never requested the communication. It is a disruptive, costly, and widespread abuse of Internet communication standards.

Sendmail makes it relatively easy to block new spamming techniques being employed to send junk email. It even blocks many of the more usual spamming methods by default. Main anti-spam features available in sendmail are *header checks, relaying denial (default from version 8.9), access database and sender information checks.*

For example, forwarding of SMTP messages, also called relaying, has been disabled by default since Sendmail version 8.9. Before this change occurred, Sendmail directed the mail host (**x.edu**) to accept messages from one party (**y.com**) and sent them to a different party (**z.net**). Now, however, Sendmail must be configured to permit any domain to relay mail through the server. To configure relay domains, edit the **/etc/mail/relay-domains** file and restart Sendmail.

However, many times users are bombarded with spam from other servers throughout the Internet. In these instances, Sendmail's access control features available through the **/etc/mail/access** file can be used to prevent connections from unwanted hosts. The following example illustrates how this file can be used to both block and specifically allow access to the Sendmail server:

```
badspammer.com ERROR:550 "Go away and do not spam us anymore"
 tux.badspammer.com OK 10.0 RELAY
```

This example shows that any email sent from **badspammer.com** is blocked with a 550 RFC-821 compliant error code, with a message sent back to the spammer. Email sent from the **tux.badspammer.com** sub-domain, is accepted. The last line shows that any email sent from the 10.0.*.* network can be relayed through the mail server.

Because **/etc/mail/access.db** is a database, use **makemap** to activate any changes. Do this using the following command as root:

```
makemap hash /etc/mail/access < /etc/mail/access
```

Message header analysis allows you to reject mail based on header contents. SMTP servers store information about an emails journey in the message header. As the message travels from one MTA to another, each puts in a "Received" header above all the other Received headers. It is however important to note that this information may be altered by spammers.

The above examples only represent a small part of what Sendmail can do in terms of allowing or blocking access. Refer to the **/usr/share/sendmail-cf/README** for more information and examples.

Since Sendmail calls the Procmail MDA when delivering mail, it is also possible to use a spam filtering program, such as SpamAssassin, to identify and file spam for users. Refer to *Section 13.5.2.6, “Spam Filters”* for more about using SpamAssassin.

13.3.1.6. Using Sendmail with LDAP

Using the *Lightweight Directory Access Protocol (LDAP)* is a very quick and powerful way to find specific information about a particular user from a much larger group. For example, an LDAP server can be used to look up a particular email address from a common corporate directory by the user's last name. In this kind of implementation, LDAP is largely separate from Sendmail, with LDAP storing the hierarchical user information and Sendmail only being given the result of LDAP queries in pre-addressed email messages.

However, Sendmail supports a much greater integration with LDAP, where it uses LDAP to replace separately maintained files, such as **`aliases`** and **`virtusertables`**, on different mail servers that work together to support a medium- to enterprise-level organization. In short, LDAP abstracts the mail routing level from Sendmail and its separate configuration files to a powerful LDAP cluster that can be leveraged by many different applications.

The current version of Sendmail contains support for LDAP. To extend the Sendmail server using LDAP, first get an LDAP server, such as **OpenLDAP**, running and properly configured. Then edit the **`/etc/mail/sendmail.mc`** to include the following:

```
LDAPROUTE_DOMAIN('yourdomain.com')dnl
FEATURE('ldap_routing')dnl
```

Note

This is only for a very basic configuration of Sendmail with LDAP. The configuration can differ greatly from this depending on the implementation of LDAP, especially when configuring several Sendmail machines to use a common LDAP server.

Consult **`/usr/share/sendmail-cf/README`** for detailed LDAP routing configuration instructions and examples.

Next, recreate the **`/etc/mail/sendmail.cf`** file by running **`m4`** and restarting Sendmail. Refer to *Section 13.3.1.3, “Common Sendmail Configuration Changes”* for instructions.

For more information on LDAP, refer to *Chapter 14, Lightweight Directory Access Protocol (LDAP)*.

13.3.2. Postfix

Originally developed at IBM by security expert and programmer Wietse Venema, Postfix is a Sendmail-compatible MTA that is designed to be secure, fast, and easy to configure.

To improve security, Postfix uses a modular design, where small processes with limited privileges are launched by a *master* daemon. The smaller, less privileged processes perform very specific tasks related to the various stages of mail delivery and run in a change rooted environment to limit the effects of attacks.

Configuring Postfix to accept network connections from hosts other than the local computer takes only a few minor changes in its configuration file. Yet for those with more complex needs, Postfix provides a variety of configuration options, as well as third party add ons that make it a very versatile and full-featured MTA.

The configuration files for Postfix are human readable and support upward of 250 directives. Unlike Sendmail, no macro processing is required for changes to take effect and the majority of the most commonly used options are described in the heavily commented files.

Important

Before using Postfix or Exim, the default MTA must be switched from Sendmail to the desired MTA.

13.3.2.1. The Default Postfix Installation

The Postfix executable is **/usr/sbin/postfix**. This daemon launches all related processes needed to handle mail delivery.

Postfix stores its configuration files in the **/etc/postfix/** directory. The following is a list of the more commonly used files:

- **access** — Used for access control, this file specifies which hosts are allowed to connect to Postfix.
- **aliases** — A configurable list required by the mail protocol.
- **main.cf** — The global Postfix configuration file. The majority of configuration options are specified in this file.
- **master.cf** — Specifies how Postfix interacts with various processes to accomplish mail delivery.
- **transport** — Maps email addresses to relay hosts.

> **Important**
>
> The default **/etc/postfix/main.cf** file does not allow Postfix to accept network connections from a host other than the local computer. For instructions on configuring Postfix as a server for other clients, refer to *Section 13.3.2.2, “Basic Postfix Configuration”*.

When changing some options within files in the **/etc/postfix/** directory, it may be necessary to restart the **postfix** service for the changes to take effect. The easiest way to do this is to type the following command:

```
/sbin/service postfix restart
```

13.3.2.2. Basic Postfix Configuration

By default, Postfix does not accept network connections from any host other than the local host. Perform the following steps as root to enable mail delivery for other hosts on the network:

- Edit the **/etc/postfix/main.cf** file with a text editor, such as **vi**.
- Uncomment the **mydomain** line by removing the hash mark (#), and replace *domain.tld* with the domain the mail server is servicing, such as **example.com**.
- Uncomment the **myorigin = $mydomain** line.
- Uncomment the **myhostname** line, and replace *host.domain.tld* with the hostname for the machine.
- Uncomment the **mydestination = $myhostname, localhost.$mydomain** line.
- Uncomment the **mynetworks** line, and replace *168.100.189.0/28* with a valid network setting for hosts that can connect to the server.
- Uncomment the **inet_interfaces = all** line.

- Comment the **`inet_interfaces = localhost`** line.
- Restart the **`postfix`** service.

Once these steps are complete, the host accepts outside emails for delivery.

Postfix has a large assortment of configuration options. One of the best ways to learn how to configure Postfix is to read the comments within **`/etc/postfix/main.cf`**. Additional resources including information about LDAP and SpamAssassin integration are available online at *http://www.postfix.org/*.

13.3.3. Fetchmail

Fetchmail is an MTA which retrieves email from remote servers and delivers it to the local MTA. Many users appreciate the ability to separate the process of downloading their messages located on a remote server from the process of reading and organizing their email in an MUA. Designed with the needs of dial-up users in mind, Fetchmail connects and quickly downloads all of the email messages to the mail spool file using any number of protocols, including POP3 and IMAP. It can even forward email messages to an SMTP server, if necessary.

Fetchmail is configured for each user through the use of a **`.fetchmailrc`** file in the user's home directory.

Using preferences in the **`.fetchmailrc`** file, Fetchmail checks for email on a remote server and downloads it. It then delivers it to port 25 on the local machine, using the local MTA to place the email in the correct user's spool file. If Procmail is available, it is launched to filter the email and place it in a mailbox so that it can be read by an MUA.

13.3.3.1. Fetchmail Configuration Options

Although it is possible to pass all necessary options on the command line to check for email on a remote server when executing Fetchmail, using a **`.fetchmailrc`** file is much easier. Place any desired configuration options in the **`.fetchmailrc`** file for those options to be used each time the **`fetchmail`** command is issued. It is possible to override these at the time Fetchmail is run by specifying that option on the command line.

A user's **`.fetchmailrc`** file contains three classes of configuration options:

- *global options* — Gives Fetchmail instructions that control the operation of the program or provide settings for every connection that checks for email.
- *server options* — Specifies necessary information about the server being polled, such as the hostname, as well as preferences for specific email servers, such as the port to check or number of seconds to wait before timing out. These options affect every user using that server.
- *user options* — Contains information, such as username and password, necessary to authenticate and check for email using a specified email server.

Global options appear at the top of the **`.fetchmailrc`** file, followed by one or more server options, each of which designate a different email server that Fetchmail should check. User options follow server options for each user account checking that email server. Like server options, multiple user options may be specified for use with a particular server as well as to check multiple email accounts on the same server.

Server options are called into service in the **`.fetchmailrc`** file by the use of a special option verb, **`poll`** or **`skip`**, that precedes any of the server information. The **`poll`** action tells Fetchmail to use this server option when it is run, which checks for email using the specified user options. Any server options after a **`skip`** action, however, are not checked unless this server's hostname is specified when Fetchmail is invoked. The **`skip`** option is useful when testing configurations in **`.fetchmailrc`** because it only checks skipped servers when specifically invoked, and does not affect any currently working configurations.

A sample **`.fetchmailrc`** file looks similar to the following example:

```
set postmaster "user1" set bouncemail poll pop.domain.com proto pop3 user
 'user1' there with password 'secret' is user1 here poll mail.domain2.com
 user 'user5' there with password 'secret2' is user1 here user 'user7'
 there with password 'secret3' is user1 here
```

In this example, the global options specify that the user is sent email as a last resort (**`postmaster`** option) and all email errors are sent to the postmaster instead of the sender (**`bouncemail`** option). The **`set`** action tells Fetchmail that this line contains a global option. Then, two email servers are specified, one set to check using POP3, the other for trying various protocols to find one that works. Two users are checked using the second server option, but all email found for any user is sent to **`user1`**'s mail spool. This allows multiple mailboxes to be checked on multiple servers, while appearing in a single MUA inbox. Each user's specific information begins with the **`user`** action.

Note

Users are not required to place their password in the **`.fetchmailrc`** file. Omitting the **`with password '<password>'`** section causes Fetchmail to ask for a password when it is launched.

Fetchmail has numerous global, server, and local options. Many of these options are rarely used or only apply to very specific situations. The **`fetchmail`** man page explains each option in detail, but the most common ones are listed here.

13.3.3.2. Global Options

Each global option should be placed on a single line after a **`set`** action.

- **`daemon <`*`seconds`*`>`** — Specifies daemon-mode, where Fetchmail stays in the background. Replace <*`seconds`*> with the number of seconds Fetchmail is to wait before polling the server.
- **`postmaster`** — Specifies a local user to send mail to in case of delivery problems.
- **`syslog`** — Specifies the log file for errors and status messages. By default, this is **`/var/log/maillog`**.

13.3.3.3. Server Options

Server options must be placed on their own line in **`.fetchmailrc`** after a **`poll`** or **`skip`** action.

- **`auth <`*`auth-type`*`>`** — Replace <*`auth-type`*> with the type of authentication to be used. By default, **`password`** authentication is used, but some protocols support other types of authentication, including **`kerberos_v5`**, **`kerberos_v4`**, and **`ssh`**. If the **`any`** authentication type is used, Fetchmail first tries methods that do not require a password, then methods that mask the password, and finally attempts to send the password unencrypted to authenticate to the server.

- **`interval <number>`** — Polls the specified server every **`<number>`** of times that it checks for email on all configured servers. This option is generally used for email servers where the user rarely receives messages.

- **`port <port-number>`** — Replace *`<port-number>`* with the port number. This value overrides the default port number for the specified protocol.

- **`proto <protocol>`** — Replace *`<protocol>`* with the protocol, such as **`pop3`** or **`imap`**, to use when checking for messages on the server.

- **`timeout <seconds>`** — Replace *`<seconds>`* with the number of seconds of server inactivity after which Fetchmail gives up on a connection attempt. If this value is not set, a default of **`300`** seconds is assumed.

13.3.3.4. User Options

User options may be placed on their own lines beneath a server option or on the same line as the server option. In either case, the defined options must follow the **`user`** option (defined below).

- **`fetchall`** — Orders Fetchmail to download all messages in the queue, including messages that have already been viewed. By default, Fetchmail only pulls down new messages.

- **`fetchlimit <number>`** — Replace *`<number>`* with the number of messages to be retrieved before stopping.

- **`flush`** — Deletes all previously viewed messages in the queue before retrieving new messages.

- **`limit <max-number-bytes>`** — Replace *`<max-number-bytes>`* with the maximum size in bytes that messages are allowed to be when retrieved by Fetchmail. This option is useful with slow network links, when a large message takes too long to download.

- **`password '<password>'`** — Replace *`<password>`* with the user's password.

- **`preconnect "<command>"`** — Replace *`<command>`* with a command to be executed before retrieving messages for the user.

- **`postconnect "<command>"`** — Replace *`<command>`* with a command to be executed after retrieving messages for the user.

- **`ssl`** — Activates SSL encryption.

- **`user "<username>"`** — Replace *`<username>`* with the username used by Fetchmail to retrieve messages. *This option must precede all other user options.*

13.3.3.5. Fetchmail Command Options

Most Fetchmail options used on the command line when executing the **`fetchmail`** command mirror the **`.fetchmailrc`** configuration options. In this way, Fetchmail may be used with or without a configuration file. These options are not used on the command line by most users because it is easier to leave them in the **`.fetchmailrc`** file.

There may be times when it is desirable to run the **`fetchmail`** command with other options for a particular purpose. It is possible to issue command options to temporarily override a **`.fetchmailrc`** setting that is causing an error, as any options specified at the command line override configuration file options.

13.3.3.6. Informational or Debugging Options

Certain options used after the **`fetchmail`** command can supply important information.

- **`--configdump`** — Displays every possible option based on information from **`.fetchmailrc`** and Fetchmail defaults. No email is retrieved for any users when using this option.
- **`-s`** — Executes Fetchmail in silent mode, preventing any messages, other than errors, from appearing after the **`fetchmail`** command.
- **`-v`** — Executes Fetchmail in verbose mode, displaying every communication between Fetchmail and remote email servers.
- **`-V`** — Displays detailed version information, lists its global options, and shows settings to be used with each user, including the email protocol and authentication method. No email is retrieved for any users when using this option.

13.3.3.7. Special Options

These options are occasionally useful for overriding defaults often found in the **`.fetchmailrc`** file.

- **`-a`** — Fetchmail downloads all messages from the remote email server, whether new or previously viewed. By default, Fetchmail only downloads new messages.
- **`-k`** — Fetchmail leaves the messages on the remote email server after downloading them. This option overrides the default behavior of deleting messages after downloading them.
- **`-l <max-number-bytes>`** — Fetchmail does not download any messages over a particular size and leaves them on the remote email server.
- **`--quit`** — Quits the Fetchmail daemon process.

More commands and **`.fetchmailrc`** options can be found in the **`fetchmail`** man page.

13.4. Mail Transport Agent (MTA) Configuration

A *Mail Transport Agent* (MTA) is essential for sending email. A *Mail User Agent* (MUA) such as **Evolution**, **Thunderbird**, and **Mutt**, is used to read and compose email. When a user sends an email from an MUA, the message is handed off to the MTA, which sends the message through a series of MTAs until it reaches its destination.

Even if a user does not plan to send email from the system, some automated tasks or system programs might use the **`/bin/mail`** command to send email containing log messages to the root user of the local system.

Fedora 12 provides three MTAs: Sendmail, Postfix, and Exim. If all three are installed, **`sendmail`** is the default MTA. The **Mail Transport Agent Switcher** allows for the selection of either **`sendmail`**, **`postfix`**, or **`exim`** as the default MTA for the system.

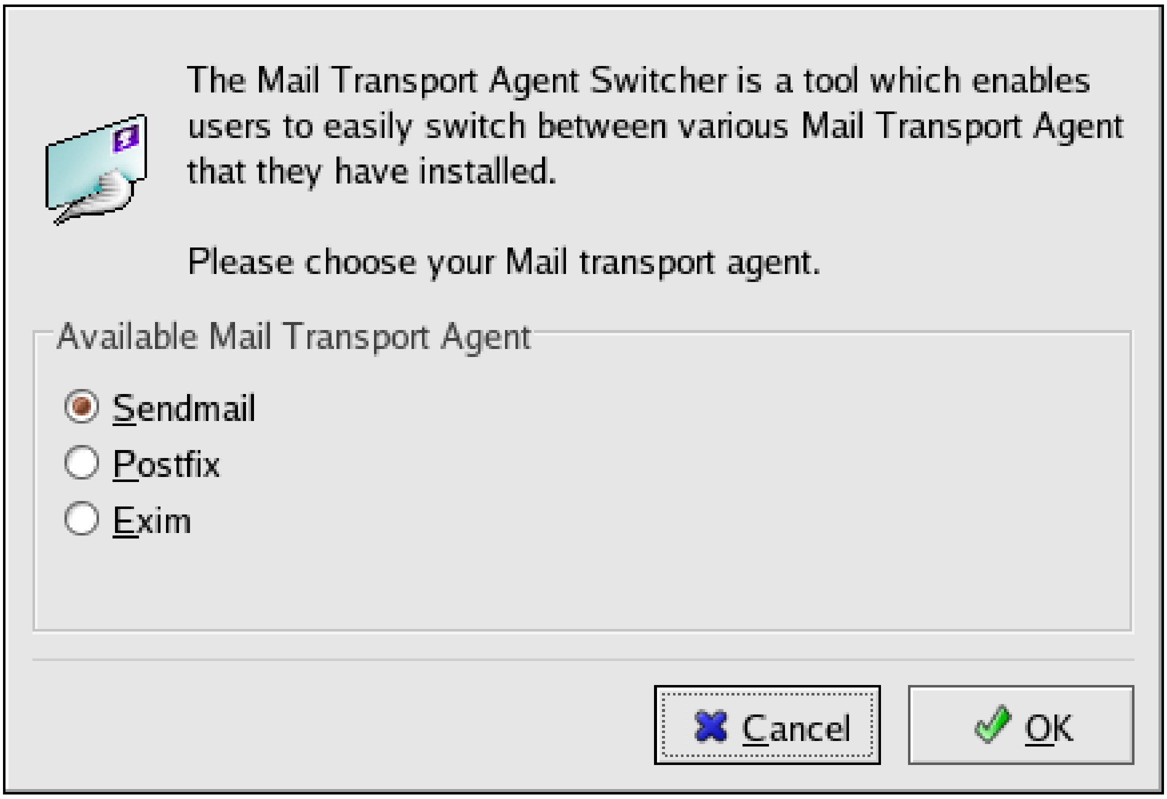
The **system-switch-mail** RPM package must be installed to use the text-based version of the **Mail Transport Agent Switcher** program. If you want to use the graphical version, the **system-switch-mail-gnome** package must also be installed.

Note

For more information on installing RPM packages, refer to *Part I, "Package Management"*.

To start the **Mail Transport Agent Switcher**, select **System** (the main menu on the panel) > **Administration** > **Mail Transport Agent Switcher**, or type the command **system-switch-mail** at a shell prompt (for example, in an XTerm or GNOME terminal).

The program automatically detects if the X Window System is running. If it is running, the program starts in graphical mode as shown in *Figure 13.1, "**Mail Transport Agent Switcher**"*. If X is not detected, it starts in text-mode. To force **Mail Transport Agent Switcher** to run in text-mode, use the command **system-switch-mail-nox**.



The Mail Transport Agent Switcher is a tool which enables users to easily switch between various Mail Transport Agent that they have installed.

Please choose your Mail transport agent.

Available Mail Transport Agent

● Sendmail

○ Postfix

○ Exim

Figure 13.1. **Mail Transport Agent Switcher**

If you select **OK** to change the MTA, the selected mail daemon is enabled to start at boot time, and the unselected mail daemons are disabled so that they do not start at boot time. The selected mail daemon is started, and any other mail daemon is stopped; thus making the changes take place immediately.

13.5. Mail Delivery Agents

Fedora includes two primary MDAs, Procmail and **`mail`**. Both of the applications are considered LDAs and both move email from the MTA's spool file into the user's mailbox. However, Procmail provides a robust filtering system.

This section details only Procmail. For information on the **`mail`** command, consult its man page.

Procmail delivers and filters email as it is placed in the mail spool file of the localhost. It is powerful, gentle on system resources, and widely used. Procmail can play a critical role in delivering email to be read by email client applications.

Procmail can be invoked in several different ways. Whenever an MTA places an email into the mail spool file, Procmail is launched. Procmail then filters and files the email for the MUA and quits. Alternatively, the MUA can be configured to execute Procmail any time a message is received so that messages are moved into their correct mailboxes. By default, the presence of **`/etc/procmailrc`** or of a **`.procmailrc`** file (also called an *rc* file) in the user's home directory invokes Procmail whenever an MTA receives a new message.

Whether Procmail acts upon an email message depends upon whether the message matches a specified set of conditions or *recipes* in the **`rc`** file. If a message matches a recipe, then the email is placed in a specified file, is deleted, or is otherwise processed.

When Procmail starts, it reads the email message and separates the body from the header information. Next, Procmail looks for **`/etc/procmailrc`** and **`rc`** files in the **`/etc/procmailrcs`** directory for default, system-wide, Procmail environmental variables and recipes. Procmail then searches for a **`.procmailrc`** file in the user's home directory. Many users also create additional **`rc`** files for Procmail that are referred to within the **`.procmailrc`** file in their home directory.

By default, no system-wide **`rc`** files exist in the **`/etc/`** directory and no **`.procmailrc`** files exist in any user's home directory. Therefore, to use Procmail, each user must construct a **`.procmailrc`** file with specific environment variables and rules.

13.5.1. Procmail Configuration

The Procmail configuration file contains important environmental variables. These variables specify things such as which messages to sort and what to do with the messages that do not match any recipes.

These environmental variables usually appear at the beginning of **`.procmailrc`** in the following format:

```
<env-variable>="<value>"
```

In this example, ***`<env-variable>`*** is the name of the variable and ***`<value>`*** defines the variable.

There are many environment variables not used by most Procmail users and many of the more important environment variables are already defined by a default value. Most of the time, the following variables are used:

- **`DEFAULT`** — Sets the default mailbox where messages that do not match any recipes are placed.

The default **DEFAULT** value is the same as **$ORGMAIL**.

- **INCLUDERC** — Specifies additional **rc** files containing more recipes for messages to be checked against. This breaks up the Procmail recipe lists into individual files that fulfill different roles, such as blocking spam and managing email lists, that can then be turned off or on by using comment characters in the user's **.procmailrc** file.

  For example, lines in a user's **.procmailrc** file may look like this:

  ```
  MAILDIR=$HOME/Msgs INCLUDERC=$MAILDIR/lists.rc INCLUDERC=$MAILDIR/spam.rc
  ```

  If the user wants to turn off Procmail filtering of their email lists but leave spam control in place, they would comment out the first **INCLUDERC** line with a hash mark character (**#**).

- **LOCKSLEEP** — Sets the amount of time, in seconds, between attempts by Procmail to use a particular lockfile. The default is eight seconds.

- **LOCKTIMEOUT** — Sets the amount of time, in seconds, that must pass after a lockfile was last modified before Procmail assumes that the lockfile is old and can be deleted. The default is 1024 seconds.

- **LOGFILE** — The file to which any Procmail information or error messages are written.

- **MAILDIR** — Sets the current working directory for Procmail. If set, all other Procmail paths are relative to this directory.

- **ORGMAIL** — Specifies the original mailbox, or another place to put the messages if they cannot be placed in the default or recipe-required location.

  By default, a value of **/var/spool/mail/$LOGNAME** is used.

- **SUSPEND** — Sets the amount of time, in seconds, that Procmail pauses if a necessary resource, such as swap space, is not available.

- **SWITCHRC** — Allows a user to specify an external file containing additional Procmail recipes, much like the **INCLUDERC** option, except that recipe checking is actually stopped on the referring configuration file and only the recipes on the **SWITCHRC**-specified file are used.

- **VERBOSE** — Causes Procmail to log more information. This option is useful for debugging.

Other important environmental variables are pulled from the shell, such as **LOGNAME**, which is the login name; **HOME**, which is the location of the home directory; and **SHELL**, which is the default shell.

A comprehensive explanation of all environments variables, as well as their default values, is available in the **procmailrc** man page.

13.5.2. Procmail Recipes

New users often find the construction of recipes the most difficult part of learning to use Procmail. To some extent, this is understandable, as recipes do their message matching using *regular expressions*, which is a particular format used to specify qualifications for a matching string. However, regular expressions are not very difficult to construct and even less difficult to understand when read. Additionally, the consistency of the way Procmail recipes are written, regardless of regular expressions, makes it easy to learn by example. To see example Procmail recipes, refer to *Section 13.5.2.5, “Recipe Examples”*.

Procmail recipes take the following form:

```
:0<flags>: <lockfile-name> * <special-condition-character>
        <condition-1> * <special-condition-character>
        <condition-2> * <special-condition-character>
        <condition-N>
        <special-action-character>
        <action-to-perform>
```

The first two characters in a Procmail recipe are a colon and a zero. Various flags can be placed after the zero to control how Procmail processes the recipe. A colon after the ***<flags>*** section specifies that a lockfile is created for this message. If a lockfile is created, the name can be specified by replacing ***<lockfile-name>***.

A recipe can contain several conditions to match against the message. If it has no conditions, every message matches the recipe. Regular expressions are placed in some conditions to facilitate message matching. If multiple conditions are used, they must all match for the action to be performed. Conditions are checked based on the flags set in the recipe's first line. Optional special characters placed after the ***** character can further control the condition.

The ***<action-to-perform>*** specifies the action taken when the message matches one of the conditions. There can only be one action per recipe. In many cases, the name of a mailbox is used here to direct matching messages into that file, effectively sorting the email. Special action characters may also be used before the action is specified. Refer to *Section 13.5.2.4, “Special Conditions and Actions”* for more information.

13.5.2.1. Delivering vs. Non-Delivering Recipes

The action used if the recipe matches a particular message determines whether it is considered a *delivering* or *non-delivering* recipe. A delivering recipe contains an action that writes the message to a file, sends the message to another program, or forwards the message to another email address. A non-delivering recipe covers any other actions, such as a *nesting block*. A nesting block is a set of actions, contained in braces **{}**, that are performed on messages which match the recipe's conditions. Nesting blocks can be nested inside one another, providing greater control for identifying and performing actions on messages.

When messages match a delivering recipe, Procmail performs the specified action and stops comparing the message against any other recipes. Messages that match non-delivering recipes continue to be compared against other recipes.

13.5.2.2. Flags

Flags are essential to determine how or if a recipe's conditions are compared to a message. The following flags are commonly used:

- **A** — Specifies that this recipe is only used if the previous recipe without an **A** or **a** flag also matched this message.
- **a** — Specifies that this recipe is only used if the previous recipe with an **A** or **a** flag also matched this message *and* was successfully completed.

- **`B`** — Parses the body of the message and looks for matching conditions.
- **`b`** — Uses the body in any resulting action, such as writing the message to a file or forwarding it. This is the default behavior.
- **`c`** — Generates a carbon copy of the email. This is useful with delivering recipes, since the required action can be performed on the message and a copy of the message can continue being processed in the **`rc`** files.
- **`D`** — Makes the **`egrep`** comparison case-sensitive. By default, the comparison process is not case-sensitive.
- **`E`** — While similar to the **`A`** flag, the conditions in the recipe are only compared to the message if the immediately preceding the recipe without an **`E`** flag did not match. This is comparable to an *else* action.
- **`e`** — The recipe is compared to the message only if the action specified in the immediately preceding recipe fails.
- **`f`** — Uses the pipe as a filter.
- **`H`** — Parses the header of the message and looks for matching conditions. This occurs by default.
- **`h`** — Uses the header in a resulting action. This is the default behavior.
- **`w`** — Tells Procmail to wait for the specified filter or program to finish, and reports whether or not it was successful before considering the message filtered.
- **`W`** — Is identical to **`w`** except that "Program failure" messages are suppressed.

For a detailed list of additional flags, refer to the **`procmailrc`** man page.

13.5.2.3. Specifying a Local Lockfile

Lockfiles are very useful with Procmail to ensure that more than one process does not try to alter a message simultaneously. Specify a local lockfile by placing a colon (**`:`**) after any flags on a recipe's first line. This creates a local lockfile based on the destination file name plus whatever has been set in the **`LOCKEXT`** global environment variable.

Alternatively, specify the name of the local lockfile to be used with this recipe after the colon.

13.5.2.4. Special Conditions and Actions

Special characters used before Procmail recipe conditions and actions change the way they are interpreted.

The following characters may be used after the **`*`** character at the beginning of a recipe's condition line:

- **`!`** — In the condition line, this character inverts the condition, causing a match to occur only if the condition does not match the message.
- **`<`** — Checks if the message is under a specified number of bytes.
- **`>`** — Checks if the message is over a specified number of bytes.

The following characters are used to perform special actions:

- **!** — In the action line, this character tells Procmail to forward the message to the specified email addresses.
- **$** — Refers to a variable set earlier in the **rc** file. This is often used to set a common mailbox that is referred to by various recipes.
- **|** — Starts a specified program to process the message.
- **{** and **}** — Constructs a nesting block, used to contain additional recipes to apply to matching messages.

If no special character is used at the beginning of the action line, Procmail assumes that the action line is specifying the mailbox in which to write the message.

13.5.2.5. Recipe Examples

Procmail is an extremely flexible program, but as a result of this flexibility, composing Procmail recipes from scratch can be difficult for new users.

The best way to develop the skills to build Procmail recipe conditions stems from a strong understanding of regular expressions combined with looking at many examples built by others. A thorough explanation of regular expressions is beyond the scope of this section. The structure of Procmail recipes and useful sample Procmail recipes can be found at various places on the Internet (such as *http://www.iki.fi/era/procmail/links.html*). The proper use and adaptation of regular expressions can be derived by viewing these recipe examples. In addition, introductory information about basic regular expression rules can be found in the **grep** man page.

The following simple examples demonstrate the basic structure of Procmail recipes and can provide the foundation for more intricate constructions.

A basic recipe may not even contain conditions, as is illustrated in the following example:

```
:0: new-mail.spool
```

The first line specifies that a local lockfile is to be created but does not specify a name, so Procmail uses the destination file name and appends the value specified in the **LOCKEXT** environment variable. No condition is specified, so every message matches this recipe and is placed in the single spool file called **new-mail.spool**, located within the directory specified by the **MAILDIR** environment variable. An MUA can then view messages in this file.

A basic recipe, such as this, can be placed at the end of all **rc** files to direct messages to a default location.

The following example matched messages from a specific email address and throws them away.

```
:0 * ^From: spammer@domain.com /dev/null
```

With this example, any messages sent by `spammer@domain.com` are sent to the **/dev/null** device, deleting them.

Caution

Be certain that rules are working as intended before sending messages to **`/dev/null`** for permanent deletion. If a recipe inadvertently catches unintended messages, and those messages disappear, it becomes difficult to troubleshoot the rule.

A better solution is to point the recipe's action to a special mailbox, which can be checked from time to time to look for false positives. Once satisfied that no messages are accidentally being matched, delete the mailbox and direct the action to send the messages to **`/dev/null`**.

The following recipe grabs email sent from a particular mailing list and places it in a specified folder.

```
:0: * ^(From|CC|To).*tux-lug tuxlug
```

Any messages sent from the `tux-lug@domain.com` mailing list are placed in the **`tuxlug`** mailbox automatically for the MUA. Note that the condition in this example matches the message if it has the mailing list's email address on the `From`, `CC`, or `To` lines.

Consult the many Procmail online resources available in *Section 13.7, “Additional Resources”* for more detailed and powerful recipes.

13.5.2.6. Spam Filters

Because it is called by Sendmail, Postfix, and Fetchmail upon receiving new emails, Procmail can be used as a powerful tool for combating spam.

This is particularly true when Procmail is used in conjunction with SpamAssassin. When used together, these two applications can quickly identify spam emails, and sort or destroy them.

SpamAssassin uses header analysis, text analysis, blacklists, a spam-tracking database, and self-learning Bayesian spam analysis to quickly and accurately identify and tag spam.

The easiest way for a local user to use SpamAssassin is to place the following line near the top of the **`~/.procmailrc`** file:

```

INCLUDERC=/etc/mail/spamassassin/spamassassin-default.rc
```

The **`/etc/mail/spamassassin/spamassassin-default.rc`** contains a simple Procmail rule that activates SpamAssassin for all incoming email. If an email is determined to be spam, it is tagged in the header as such and the title is prepended with the following pattern:

```
*****SPAM*****
```

The message body of the email is also prepended with a running tally of what elements caused it to be diagnosed as spam.

To file email tagged as spam, a rule similar to the following can be used:

```
:0 Hw * ^X-Spam-Status: Yes spam
```

This rule files all email tagged in the header as spam into a mailbox called **spam**.

Since SpamAssassin is a Perl script, it may be necessary on busy servers to use the binary SpamAssassin daemon (**spamd**) and client application (**spamc**). Configuring SpamAssassin this way, however, requires root access to the host.

To start the **spamd** daemon, type the following command as root:

```
/sbin/service spamassassin start
```

To start the SpamAssassin daemon when the system is booted, use an initscript utility, such as the **Services Configuration Tool** (**system-config-services**), to turn on the `spamassassin` service. Refer to for more information about initscript utilities.

To configure Procmail to use the SpamAssassin client application instead of the Perl script, place the following line near the top of the **~/.procmailrc** file. For a system-wide configuration, place it in **/etc/procmailrc**:

```
INCLUDERC=/etc/mail/spamassassin/spamassassin-spamc.rc
```

13.6. Mail User Agents

There are scores of mail programs available under Fedora. There are full-featured, graphical email client programs, such as **Ximian Evolution**, as well as text-based email programs such as **mutt**.

The remainder of this section focuses on securing communication between the client and server.

13.6.1. Securing Communication

Popular MUAs included with Fedora, such as **Ximian Evolution** and **mutt** offer SSL-encrypted email sessions.

Like any other service that flows over a network unencrypted, important email information, such as usernames, passwords, and entire messages, may be intercepted and viewed by users on the network. Additionally, since the standard POP and IMAP protocols pass authentication information unencrypted, it is possible for an attacker to gain access to user accounts by collecting usernames and passwords as they are passed over the network.

13.6.1.1. Secure Email Clients

Most Linux MUAs designed to check email on remote servers support SSL encryption. To use SSL when retrieving email, it must be enabled on both the email client and server.

SSL is easy to enable on the client-side, often done with the click of a button in the MUA's configuration window or via an option in the MUA's configuration file. Secure IMAP and POP have known port numbers (993 and 995, respectively) that the MUA uses to authenticate and download messages.

13.6.1.2. Securing Email Client Communications

Offering SSL encryption to IMAP and POP users on the email server is a simple matter.

First, create an SSL certificate. This can be done two ways: by applying to a *Certificate Authority* (*CA*) for an SSL certificate or by creating a self-signed certificate.

Caution

Self-signed certificates should be used for testing purposes only. Any server used in a production environment should use an SSL certificate granted by a CA.

To create a self-signed SSL certificate for IMAP, change to the **/etc/pki/tls/certs/** directory and type the following commands as root:

```
rm -f cyrus-imapd.pem make cyrus-imapd.pem
```

Answer all of the questions to complete the process.

To create a self-signed SSL certificate for POP, change to the **/etc/pki/tls/certs/** directory, and type the following commands as root:

```
rm -f ipop3d.pem make ipop3d.pem
```

Again, answer all of the questions to complete the process.

Important

Please be sure to remove the default **imapd.pem** and **ipop3d.pem** files before issuing each **make** command.

Once finished, execute the **/sbin/service xinetd restart** command to restart the **xinetd** daemon which controls **imapd** and **ipop3d**.

Alternatively, the **stunnel** command can be used as an SSL encryption wrapper around the standard, non-secure daemons, **imapd** or **pop3d**.

The **stunnel** program uses external OpenSSL libraries included with Fedora to provide strong cryptography and protect the connections. It is best to apply to a CA to obtain an SSL certificate, but it is also possible to create a self-signed certificate.

To create a self-signed SSL certificate, change to the **/etc/pki/tls/certs/** directory, and type the following command:

```
make stunnel.pem
```

Again, answer all of the questions to complete the process.

Once the certificate is generated, it is possible to use the **stunnel** command to start the **imapd** mail daemon using the following command:

```
/usr/sbin/stunnel -d 993 -l /usr/sbin/imapd imapd
```

Once this command is issued, it is possible to open an IMAP email client and connect to the email server using SSL encryption.

To start the **pop3d** using the **stunnel** command, type the following command:

```
/usr/sbin/stunnel -d 995 -l /usr/sbin/pop3d pop3d
```

For more information about how to use **stunnel**, read the **stunnel** man page or refer to the documents in the **/usr/share/doc/stunnel-<*version-number*>**/ directory, where <*version-number*> is the version number for **stunnel**.

13.7. Additional Resources

The following is a list of additional documentation about email applications.

13.7.1. Installed Documentation

- Information on configuring Sendmail is included with the **sendmail** and **sendmail-cf** packages.

  - **/usr/share/sendmail-cf/README** — Contains information on **m4**, file locations for Sendmail, supported mailers, how to access enhanced features, and more.

  In addition, the **sendmail** and **aliases** man pages contain helpful information covering various Sendmail options and the proper configuration of the Sendmail **/etc/mail/aliases** file.

- **/usr/share/doc/postfix-<*version-number*>** — Contains a large amount of information about ways to configure Postfix. Replace <*version-number*> with the version number of Postfix.

- **/usr/share/doc/fetchmail-<*version-number*>** — Contains a full list of Fetchmail features in the **FEATURES** file and an introductory **FAQ** document. Replace <*version-number*> with the version number of Fetchmail.

- **/usr/share/doc/procmail-<*version-number*>** — Contains a **README** file that provides an overview of Procmail, a **FEATURES** file that explores every program feature, and an **FAQ** file with answers to many common configuration questions. Replace <*version-number*> with the version number of Procmail.

  When learning how Procmail works and creating new recipes, the following Procmail man pages are invaluable:

  - **procmail** — Provides an overview of how Procmail works and the steps involved with filtering email.

  - **procmailrc** — Explains the **rc** file format used to construct recipes.

  - **procmailex** — Gives a number of useful, real-world examples of Procmail recipes.

  - **`procmailsc`** — Explains the weighted scoring technique used by Procmail to match a particular recipe to a message.

  - ***`/usr/share/doc/spamassassin-<version-number>/`*** — Contains a large amount of information pertaining to SpamAssassin. Replace *`<version-number>`* with the version number of the **`spamassassin`** package.

13.7.2. Useful Websites

- *http://www.sendmail.org/* — Offers a thorough technical breakdown of Sendmail features, documentation and configuration examples.

- *http://www.sendmail.com/* — Contains news, interviews and articles concerning Sendmail, including an expanded view of the many options available.

- *http://www.postfix.org/* — The Postfix project home page contains a wealth of information about Postfix. The mailing list is a particularly good place to look for information.

- *http://fetchmail.berlios.de/* — The home page for Fetchmail, featuring an online manual, and a thorough FAQ.

- *http://www.procmail.org/* — The home page for Procmail with links to assorted mailing lists dedicated to Procmail as well as various FAQ documents.

- *http://partmaps.org/era/procmail/mini-faq.html* — An excellent Procmail FAQ, offers troubleshooting tips, details about file locking, and the use of wildcard characters.

- *http://www.uwasa.fi/~ts/info/proctips.html* — Contains dozens of tips that make using Procmail much easier. Includes instructions on how to test **`.procmailrc`** files and use Procmail scoring to decide if a particular action should be taken.

- *http://www.spamassassin.org/* — The official site of the SpamAssassin project.

13.7.3. Related Books

- *Sendmail Milters: A Guide for Fighting Spam* by Bryan Costales and Marcia Flynt; Addison-Wesley — A good Sendmail guide that can help you customise your mail filters.

- *Sendmail* by Bryan Costales with Eric Allman et al; O'Reilly & Associates — A good Sendmail reference written with the assistance of the original creator of Delivermail and Sendmail.

- *Removing the Spam: Email Processing and Filtering* by Geoff Mulligan; Addison-Wesley Publishing Company — A volume that looks at various methods used by email administrators using established tools, such as Sendmail and Procmail, to manage spam problems.

- *Internet Email Protocols: A Developer's Guide* by Kevin Johnson; Addison-Wesley Publishing Company — Provides a very thorough review of major email protocols and the security they provide.

- *Managing IMAP* by Dianna Mullet and Kevin Mullet; O'Reilly & Associates — Details the steps required to configure an IMAP server.

Lightweight Directory Access Protocol (LDAP)

The *Lightweight Directory Access Protocol* (*LDAP*) is a set of open protocols used to access centrally stored information over a network. It is based on the *X.500* standard for directory sharing, but is less complex and resource-intensive. For this reason, LDAP is sometimes referred to as "*X.500 Lite*." The X.500 standard is a directory that contains hierarchical and categorized information, which could include information such as names, addresses, and phone numbers.

Like X.500, LDAP organizes information in a hierarchal manner using directories. These directories can store a variety of information and can even be used in a manner similar to the Network Information Service (NIS), enabling anyone to access their account from any machine on the LDAP enabled network.

In many cases, LDAP is used as a virtual phone directory, allowing users to easily access contact information for other users. But LDAP is more flexible than a traditional phone directory, as it is capable of referring a querent to other LDAP servers throughout the world, providing an ad-hoc global repository of information. Currently, however, LDAP is more commonly used within individual organizations, like universities, government departments, and private companies.

LDAP is a client/server system. The server can use a variety of databases to store a directory, each optimized for quick and copious read operations. When an LDAP client application connects to an LDAP server, it can either query a directory or attempt to modify it. In the event of a query, the server either answers the query locally, or it can refer the querent to an LDAP server which does have the answer. If the client application is attempting to modify information within an LDAP directory, the server verifies that the user has permission to make the change and then adds or updates the information.

This chapter refers to the configuration and use of OpenLDAP 2.0, an open source implementation of the LDAPv2 and LDAPv3 protocols.

14.1. Why Use LDAP?

The main benefit of using LDAP is that information for an entire organization can be consolidated into a central repository. For example, rather than managing user lists for each group within an organization, LDAP can be used as a central directory accessible from anywhere on the network. And because LDAP supports Secure Sockets Layer (SSL) and Transport Layer Security (TLS), sensitive data can be protected from prying eyes.

LDAP also supports a number of back-end databases in which to store directories. This allows administrators the flexibility to deploy the database best suited for the type of information the server is to disseminate. Because LDAP also has a well-defined client Application Programming Interface (API), the number of LDAP-enabled applications are numerous and increasing in quantity and quality.

14.1.1. OpenLDAP Features

OpenLDAP includes a number of important features.

- *LDAPv3 Support* — OpenLDAP supports Simple Authentication and Security Layer (SASL), Transport Layer Security (TLS), and Secure Sockets Layer (SSL), among other improvements. Many of the changes in the protocol since LDAPv2 are designed to make LDAP more secure.

- *IPv6 Support* — OpenLDAP supports the next generation Internet Protocol version 6.
- *LDAP Over IPC* — OpenLDAP can communicate within a system using interprocess communication (IPC). This enhances security by eliminating the need to communicate over a network.
- *Updated C API* — Improves the way programmers can connect to and use LDAP directory servers.
- *LDIFv1 Support* — Provides full compliance with the LDAP Data Interchange Format (LDIF) version 1.
- *Enhanced Stand-Alone LDAP Server* — Includes an updated access control system, thread pooling, better tools, and much more.

14.2. LDAP Terminology

Any discussion of LDAP requires a basic understanding of a set of LDAP-specific terms:

- *entry* — A single unit within an LDAP directory. Each entry is identified by its unique *Distinguished Name (DN)*.
- *attributes* — Information directly associated with an entry. For example, an organization could be represented as an LDAP entry. Attributes associated with the organization might include a fax number, an address, and so on. People can also be represented as entries in an LDAP directory, with common attributes such as the person's telephone number and email address.

  Some attributes are required, while other attributes are optional. An *objectclass* definition sets which attributes are required for each entry. Objectclass definitions are found in various schema files, located in the **`/etc/openldap/schema/`** directory. For more information, refer to *Section 14.5, “The **`/etc/openldap/schema/`** Directory”*.

  The assertion of an attribute and its corresponding value is also referred to as a *Relative Distinguished Name* (RDN). An RDN is only unique per entry, whereas a DN is globally unique.
- *LDIF* — The *LDAP Data Interchange Format* (LDIF) is an ASCII text representation of LDAP entries. Files used for importing data to LDAP servers must be in LDIF format. An LDIF entry looks similar to the following example:

  ```
  [<id>] dn: <distinguished name>
  <attrtype>: <attrvalue>
  <attrtype>: <attrvalue>
  <attrtype>: <attrvalue>
  ```

  Each entry can contain as many **`<attrtype>: <attrvalue>`** pairs as needed. A blank line indicates the end of an entry.

  Caution

  All **`<attrtype>`** and **`<attrvalue>`** pairs *must* be defined in a corresponding schema file to use this information.

Any value enclosed within a **<** and a **>** is a variable and can be set whenever a new LDAP entry is created. This rule does not apply, however, to **<*id*>**. The **<*id*>** is a number determined by the application used to edit the entry.

14.3. OpenLDAP Daemons and Utilities

The suite of OpenLDAP libraries and tools are included within the following packages:

- **openldap** — Contains the libraries necessary to run the OpenLDAP server and client applications.
- **openldap-clients** — Contains command line tools for viewing and modifying directories on an LDAP server.
- **openldap-servers** — Contains the servers and other utilities necessary to configure and run an LDAP server.

There are two servers contained in the **openldap-servers** package: the *Standalone LDAP Daemon* (**/usr/sbin/slapd**) and the *Standalone LDAP Update Replication Daemon* (**/usr/sbin/slurpd**).

The **slapd** daemon is the standalone LDAP server while the **slurpd** daemon is used to synchronize changes from one LDAP server to other LDAP servers on the network. The **slurpd** daemon is only used when dealing with multiple LDAP servers.

To perform administrative tasks, the **openldap-servers** package installs the following utilities into the **/usr/sbin/** directory:

- **slapadd** — Adds entries from an LDIF file to an LDAP directory. For example, the command **/usr/sbin/slapadd -l *ldif-input*** reads in the LDIF file, ***ldif-input***, containing the new entries.

> ### Important
>
> Only the root user may use **/usr/sbin/slapadd**. However, the directory server runs as the **ldap** user. Therefore the directory server is unable to modify any files created by **slapadd**. To correct this issue, after using **slapadd**, type the following command:
>
> ```
> chown -R ldap /var/lib/ldap
> ```

- **slapcat** — Pulls entries from an LDAP directory in the default format, *Sleepycat Software's Berkeley DB* system, and saves them in an LDIF file. For example, the command **/usr/sbin/slapcat -l *ldif-output*** outputs an LDIF file called ***ldif-output*** containing the entries from the LDAP directory.
- **slapindex** — Re-indexes the **slapd** directory based on the current content. This tool should be run whenever indexing options within **/etc/openldap/slapd.conf** are changed.
- **slappasswd** — Generates an encrypted user password value for use with **ldapmodify** or the **rootpw** value in the **slapd** configuration file, **/etc/openldap/slapd.conf**. Execute the **/usr/sbin/slappasswd** command to create the password.

Warning

You must stop **`slapd`** by issuing the **`/sbin/service ldap stop`** command before using **`slapadd`**, **`slapcat`** or **`slapindex`**. Otherwise, the integrity of the LDAP directory is at risk.

For more information on using these utilities, refer to their respective man pages.

The **`openldap-clients`** package installs tools into **`/usr/bin/`** which are used to add, modify, and delete entries in an LDAP directory. These tools include the following:

- **`ldapadd`** — Adds entries to an LDAP directory by accepting input via a file or standard input; **`ldapadd`** is actually a hard link to **`ldapmodify -a`**.
- **`ldapdelete`** — Deletes entries from an LDAP directory by accepting user input at a shell prompt or via a file.
- **`ldapmodify`** — Modifies entries in an LDAP directory, accepting input via a file or standard input.
- **`ldappasswd`** — Sets the password for an LDAP user.
- **`ldapsearch`** — Searches for entries in an LDAP directory using a shell prompt.
- **`ldapcompare`** — Opens a connection to an LDAP server, binds, and performs a comparison using specified parameters.
- **`ldapwhoami`** — Opens a connection to an LDAP server, binds, and performs a **`whoami`** operation.
- **`ldapmodrdn`** — Opens a connection to an LDAP server, binds, and modifies the RDNs of entries.

With the exception of **`ldapsearch`**, each of these utilities is more easily used by referencing a file containing the changes to be made rather than typing a command for each entry to be changed within an LDAP directory. The format of such a file is outlined in the man page for each utility.

14.3.1. NSS, PAM, and LDAP

In addition to the OpenLDAP packages, Fedora includes a package called **`nss_ldap`**, which enhances LDAP's ability to integrate into both Linux and other UNIX environments.

The **`nss_ldap`** package provides the following modules (where `<`*`version`*`>` refers to the version of **`libnss_ldap`** in use):

- **`/lib/libnss_ldap-<`*`version`*`>.so`**
- **`/lib/security/pam_ldap.so`**

The **`nss_ldap`** package provides the following modules for Itanium or AMD64 architectures:

- **`/lib64/libnss_ldap-<`*`version`*`>.so`**
- **`/lib64/security/pam_ldap.so`**

The **`libnss_ldap-<`*`version`*`>.so`** module allows applications to look up users, groups, hosts, and other information using an LDAP directory via the *Nameservice Switch* (NSS) interface of **`glibc`**.

NSS allows applications to authenticate using LDAP in conjunction with the NIS name service and flat authentication files.

The **pam_ldap** module allows PAM-aware applications to authenticate users using information stored in an LDAP directory. PAM-aware applications include console login, POP and IMAP mail servers, and Samba. By deploying an LDAP server on a network, all of these applications can authenticate using the same user ID and password combination, greatly simplifying administration.

For more about configuring PAM, refer to and the PAM man pages.

14.3.2. PHP4, LDAP, and the Apache HTTP Server

Fedora includes a package containing an LDAP module for the PHP server-side scripting language.

The **php-ldap** package adds LDAP support to the PHP4 HTML-embedded scripting language via the **/usr/lib/php4/ldap.so** module. This module allows PHP4 scripts to access information stored in an LDAP directory.

Fedora ships with the **mod_authz_ldap** module for the Apache HTTP Server. This module uses the short form of the distinguished name for a subject and the issuer of the client SSL certificate to determine the distinguished name of the user within an LDAP directory. It is also capable of authorizing users based on attributes of that user's LDAP directory entry, determining access to assets based on the user and group privileges of the asset, and denying access for users with expired passwords. The **mod_ssl** module is required when using the **mod_authz_ldap** module.

Important

The **mod_authz_ldap** module does not authenticate a user to an LDAP directory using an encrypted password hash. This functionality is provided by the experimental **mod_auth_ldap** module, which is not included with Fedora. Refer to the Apache Software Foundation website online at *http://www.apache.org/* for details on the status of this module.

14.3.3. LDAP Client Applications

There are graphical LDAP clients available which support creating and modifying directories, but they are *not* included with Fedora. One such application is **LDAP Browser/Editor** — A Java-based tool available online at *http://www.iit.edu/~gawojar/ldap/*.

Other LDAP clients access directories as read-only, using them to reference, but not alter, organization-wide information. Some examples of such applications are Sendmail, **Mozilla**, **Gnome Meeting**, and **Evolution**.

14.4. OpenLDAP Configuration Files

OpenLDAP configuration files are installed into the **/etc/openldap/** directory. The following is a brief list highlighting the most important directories and files:

- **/etc/openldap/ldap.conf** — This is the configuration file for all *client* applications which use the OpenLDAP libraries such as **ldapsearch**, **ldapadd**, Sendmail, **Evolution**, and **Gnome Meeting**.

- **`/etc/openldap/slapd.conf`** — This is the configuration file for the **`slapd`** daemon. Refer to *Section 14.6.1, “Editing **`/etc/openldap/slapd.conf`**”* for more information.
- **`/etc/openldap/schema/`** directory — This subdirectory contains the schema used by the **`slapd`** daemon. Refer to *Section 14.5, “The **`/etc/openldap/schema/`** Directory”* for more information.

> **Note**
>
> If the **`nss_ldap`** package is installed, it creates a file named **`/etc/ldap.conf`**. This file is used by the PAM and NSS modules supplied by the **`nss_ldap`** package. Refer to *Section 14.7, “Configuring a System to Authenticate Using OpenLDAP”* for more information.

14.5. The `/etc/openldap/schema/` Directory

The **`/etc/openldap/schema/`** directory holds LDAP definitions, previously located in the **`slapd.at.conf`** and **`slapd.oc.conf`** files. The **`/etc/openldap/schema/redhat/`** directory holds customized schemas distributed by Red Hat for Fedora.

All *attribute syntax definitions* and *objectclass definitions* are now located in the different schema files. The various schema files are referenced in **`/etc/openldap/slapd.conf`** using **`include`** lines, as shown in this example:

```
include   /etc/openldap/schema/core.schema
include   /etc/openldap/schema/cosine.schema
include   /etc/openldap/schema/inetorgperson.schema
include   /etc/openldap/schema/nis.schema
include   /etc/openldap/schema/rfc822-MailMember.schema
include   /etc/openldap/schema/redhat/autofs.schema
```

> **Caution**
>
> Do not modify schema items defined in the schema files installed by OpenLDAP.

It is possible to extend the schema used by OpenLDAP to support additional attribute types and object classes using the default schema files as a guide. To do this, create a **`local.schema`** file in the **`/etc/openldap/schema/`** directory. Reference this new schema within **`slapd.conf`** by adding the following line below the default **`include`** schema lines:

```
include           /etc/openldap/schema/local.schema
```

Next, define new attribute types and object classes within the **`local.schema`** file. Many organizations use existing attribute types from the schema files installed by default and add new object classes to the **`local.schema`** file.

Extending the schema to match certain specialized requirements is quite involved and beyond the scope of this chapter. Refer to *http://www.openldap.org/doc/admin/schema.html* for information.

14.6. OpenLDAP Setup Overview

This section provides a quick overview for installing and configuring an OpenLDAP directory. For more details, refer to the following URLs:

- *http://www.openldap.org/doc/admin/quickstart.html* — The *Quick-Start Guide* on the OpenLDAP website.

- *http://www.tldp.org/HOWTO/LDAP-HOWTO/index.html* — The *LDAP Linux HOWTO* from the Linux Documentation Project.

The basic steps for creating an LDAP server are as follows:

1. Install the **openldap**, **openldap-servers**, and **openldap-clients** RPMs.

2. Edit the **/etc/openldap/slapd.conf** file to specify the LDAP domain and server. Refer to *Section 14.6.1, "Editing **/etc/openldap/slapd.conf**"* for more information.

3. Start **slapd** with the command:

   ```
   /sbin/service ldap start
   ```

   After configuring LDAP, use **chkconfig**, **/usr/sbin/ntsysv**, or the **Services Configuration Tool** to configure LDAP to start at boot time. For more information about configuring services, refer to *Chapter 6, Controlling Access to Services*.

4. Add entries to an LDAP directory with **ldapadd**.

5. Use **ldapsearch** to determine if **slapd** is accessing the information correctly.

6. At this point, the LDAP directory should be functioning properly and can be configured with LDAP-enabled applications.

14.6.1. Editing /etc/openldap/slapd.conf

To use the **slapd** LDAP server, modify its configuration file, **/etc/openldap/slapd.conf**, to specify the correct domain and server.

The **suffix** line names the domain for which the LDAP server provides information and should be changed from:

```
suffix          "dc=your-domain,dc=com"
```

Edit it accordingly so that it reflects a fully qualified domain name. For example:

```
suffix          "dc=example,dc=com"
```

The **rootdn** entry is the Distinguished Name (DN) for a user who is unrestricted by access controls or administrative limit parameters set for operations on the LDAP directory. The **rootdn** user can be thought of as the root user for the LDAP directory. In the configuration file, change the **`rootdn`** line from its default value as in the following example:

```
rootdn          "cn=root,dc=example,dc=com"
```

When populating an LDAP directory over a network, change the **`rootpw`** line — replacing the default value with an encrypted password string. To create an encrypted password string, type the following command:

```
slappasswd
```

When prompted, type and then re-type a password. The program prints the resulting encrypted password to the shell prompt.

Next, copy the newly created encrypted password into the **`/etc/openldap/slapd.conf`** on one of the **`rootpw`** lines and remove the hash mark (**`#`**).

When finished, the line should look similar to the following example:

```
rootpw {SSHA}vv2y+i6V6esazrIv70xSSnNAJE18bb2u
```

Warning

LDAP passwords, including the **`rootpw`** directive specified in **`/etc/openldap/slapd.conf`**, are sent over the network *unencrypted*, unless TLS encryption is enabled.

To enable TLS encryption, review the comments in **`/etc/openldap/slapd.conf`** and refer to the man page for **`slapd.conf`**.

For added security, the **`rootpw`** directive should be commented out after populating the LDAP directory by preceding it with a hash mark (**`#`**).

When using the **`/usr/sbin/slapadd`** command line tool locally to populate the LDAP directory, use of the **`rootpw`** directive is not necessary.

Important

Only the root user can use **`/usr/sbin/slapadd`**. However, the directory server runs as the **`ldap`** user. Therefore, the directory server is unable to modify any files created by **`slapadd`**. To correct this issue, after using **`slapadd`**, type the following command:

```
chown -R ldap /var/lib/ldap
```

14.7. Configuring a System to Authenticate Using OpenLDAP

This section provides a brief overview of how to configure OpenLDAP user authentication. Unless you are an OpenLDAP expert, more documentation than is provided here is necessary. Refer to the references provided in *Section 14.9, “Additional Resources”* for more information.

Install the Necessary LDAP Packages.

First, make sure that the appropriate packages are installed on both the LDAP server and the LDAP client machines. The LDAP server needs the **openldap-servers** package.

The **openldap**, **openldap-clients**, and **nss_ldap** packages need to be installed on all LDAP client machines.

Edit the Configuration Files.

- On the server, edit the **/etc/openldap/slapd.conf** file on the LDAP server to make sure it matches the specifics of the organization. Refer to *Section 14.6.1, “Editing **/etc/openldap/slapd.conf**”* for instructions about editing **slapd.conf**.

- On the client machines, both **/etc/ldap.conf** and **/etc/openldap/ldap.conf** need to contain the proper server and search base information for the organization.

  To do this, run the graphical **Authentication Configuration Tool** (**system-config-authentication**) and select **Enable LDAP Support** under the **User Information** tab.

  It is also possible to edit these files by hand.

- On the client machines, the **/etc/nsswitch.conf** must be edited to use LDAP.

  To do this, run the **Authentication Configuration Tool** (**system-config-authentication**) and select **Enable LDAP Support** under the **User Information** tab.

  If editing **/etc/nsswitch.conf** by hand, add **ldap** to the appropriate lines.

  For example:

```
passwd: files ldap
shadow: files ldap
group: files ldap
```

14.7.1. PAM and LDAP

To have standard PAM-enabled applications use LDAP for authentication, run the **Authentication Configuration Tool** (**system-config-authentication**) and select **Enable LDAP Support** under the the **Authentication** tab. For more about configuring PAM, refer to and the PAM man pages.

14.7.2. Migrating Old Authentication Information to LDAP Format

The **`/usr/share/openldap/migration/`** directory contains a set of shell and Perl scripts for migrating authentication information into an LDAP format.

> **Note**
>
> Perl must be installed on the system to use these scripts.

First, modify the **`migrate_common.ph`** file so that it reflects the correct domain. The default DNS domain should be changed from its default value to something like:

```
$DEFAULT_MAIL_DOMAIN = "example";
```

The default base should also be changed to something like:

```
$DEFAULT_BASE = "dc=example,dc=com";
```

The job of migrating a user database into a format that is LDAP readable falls to a group of migration scripts installed in the same directory. Using *Table 14.1, “LDAP Migration Scripts”*, decide which script to run to migrate the user database.

Run the appropriate script based on the existing name service.

The **`README`** and the **`migration-tools.txt`** files in the **`/usr/share/openldap/migration/`** directory provide more details on how to migrate the information.

Existing name service	Is LDAP running?	Script to Use
`/etc` flat files	yes	**`migrate_all_online.sh`**
`/etc` flat files	no	**`migrate_all_offline.sh`**
NetInfo	yes	**`migrate_all_netinfo_online.sh`**
NetInfo	no	**`migrate_all_netinfo_offline.sh`**
NIS (YP)	yes	**`migrate_all_nis_online.sh`**
NIS (YP)	no	**`migrate_all_nis_offline.sh`**

Table 14.1. LDAP Migration Scripts

14.8. Migrating Directories from Earlier Releases

With Fedora, OpenLDAP uses Sleepycat Software's Berkeley DB system as its on-disk storage format for directories. Earlier versions of OpenLDAP used *GNU Database Manager* (*gdbm*). For this reason, before upgrading an LDAP implementation to Fedora 5.2, original LDAP data should first be exported before the upgrade, and then reimported afterwards. This can be achieved by performing the following steps:

1. Before upgrading the operating system, run the command **`/usr/sbin/slapcat -l`** ***`ldif-output`***. This outputs an LDIF file called ***`ldif-output`*** containing the entries from the LDAP directory.
2. Upgrade the operating system, being careful not to reformat the partition containing the LDIF file.
3. Re-import the LDAP directory to the upgraded Berkeley DB format by executing the command **`/usr/sbin/slapadd -l`** ***`ldif-output`***.

14.9. Additional Resources

The following resources offer additional information on LDAP. It is highly recommended that you review these, especially the OpenLDAP website and the LDAP HOWTO, before configuring LDAP on your system(s).

14.9.1. Installed Documentation

- `/usr/share/docs/openldap-`*`<versionnumber>`*`/` directory — Contains a general **`README`** document and miscellaneous information.
- LDAP related man pages — There are a number of man pages for the various applications and configuration files involved with LDAP. The following is a list of some of the more important man pages.

  Client Applications

  - **`man ldapadd`** — Describes how to add entries to an LDAP directory.
  - **`man ldapdelete`** — Describes how to delete entries within an LDAP directory.
  - **`man ldapmodify`** — Describes how to modify entries within an LDAP directory.
  - **`man ldapsearch`** — Describes how to search for entries within an LDAP directory.
  - **`man ldappasswd`** — Describes how to set or change the password of an LDAP user.
  - **`man ldapcompare`** — Describes how to use the **`ldapcompare`** tool.
  - **`man ldapwhoami`** — Describes how to use the **`ldapwhoami`** tool.
  - **`man ldapmodrdn`** — Describes how to modify the RDNs of entries.

  Server Applications

  - **`man slapd`** — Describes command line options for the LDAP server.
  - **`man slurpd`** — Describes command line options for the LDAP replication server.

  Administrative Applications

  - **`man slapadd`** — Describes command line options used to add entries to a **`slapd`** database.
  - **`man slapcat`** — Describes command line options used to generate an LDIF file from a **`slapd`** database.
  - **`man slapindex`** — Describes command line options used to regenerate an index based upon the contents of a **`slapd`** database.

- **`man slappasswd`** — Describes command line options used to generate user passwords for LDAP directories.

Configuration Files

- **`man ldap.conf`** — Describes the format and options available within the configuration file for LDAP clients.
- **`man slapd.conf`** — Describes the format and options available within the configuration file referenced by both the LDAP server applications (**`slapd`** and **`slurpd`**) and the LDAP administrative tools (**`slapadd`**, **`slapcat`**, and **`slapindex`**).

14.9.2. Useful Websites

- *http://www.openldap.org/*[1] — Home of the OpenLDAP Project. This website contains a wealth of information about configuring OpenLDAP as well as a future roadmap and version changes.
- *http://www.padl.com/*[2] — Developers of **`nss_ldap`** and **`pam_ldap`**, among other useful LDAP tools.
- *http://www.kingsmountain.com/ldapRoadmap.shtml* — Jeff Hodges' LDAP Road Map contains links to several useful FAQs and emerging news concerning the LDAP protocol.
- *http://www.ldapman.org/articles/* — Articles that offer a good introduction to LDAP, including methods to design a directory tree and customizing directory structures.

14.9.3. Related Books

- *OpenLDAP by Example* by John Terpstra and Benjamin Coles; Prentice Hall.
- *Implementing LDAP* by Mark Wilcox; Wrox Press, Inc.
- *Understanding and Deploying LDAP Directory Services* by Tim Howes et al.; Macmillan Technical Publishing.

Authentication Configuration

When a user logs in to a Fedora system, the username and password combination must be verified, or *authenticated*, as a valid and active user. Sometimes the information to verify the user is located on the local system, and other times the system defers the authentication to a user database on a remote system.

The **Authentication Configuration Tool** provides a graphical interface for configuring user information retrieval from NIS, LDAP, and Hesiod servers. This tool also allows you to configure LDAP, Kerberos, and SMB as authentication protocols.

Note

If you configured a medium or high security level during installation (or with the **Security Level Configuration Tool**), then the firewall will prevent NIS (Network Information Service) authentication.

This chapter does not explain each of the different authentication types in detail. Instead, it explains how to use the **Authentication Configuration Tool** to configure them.

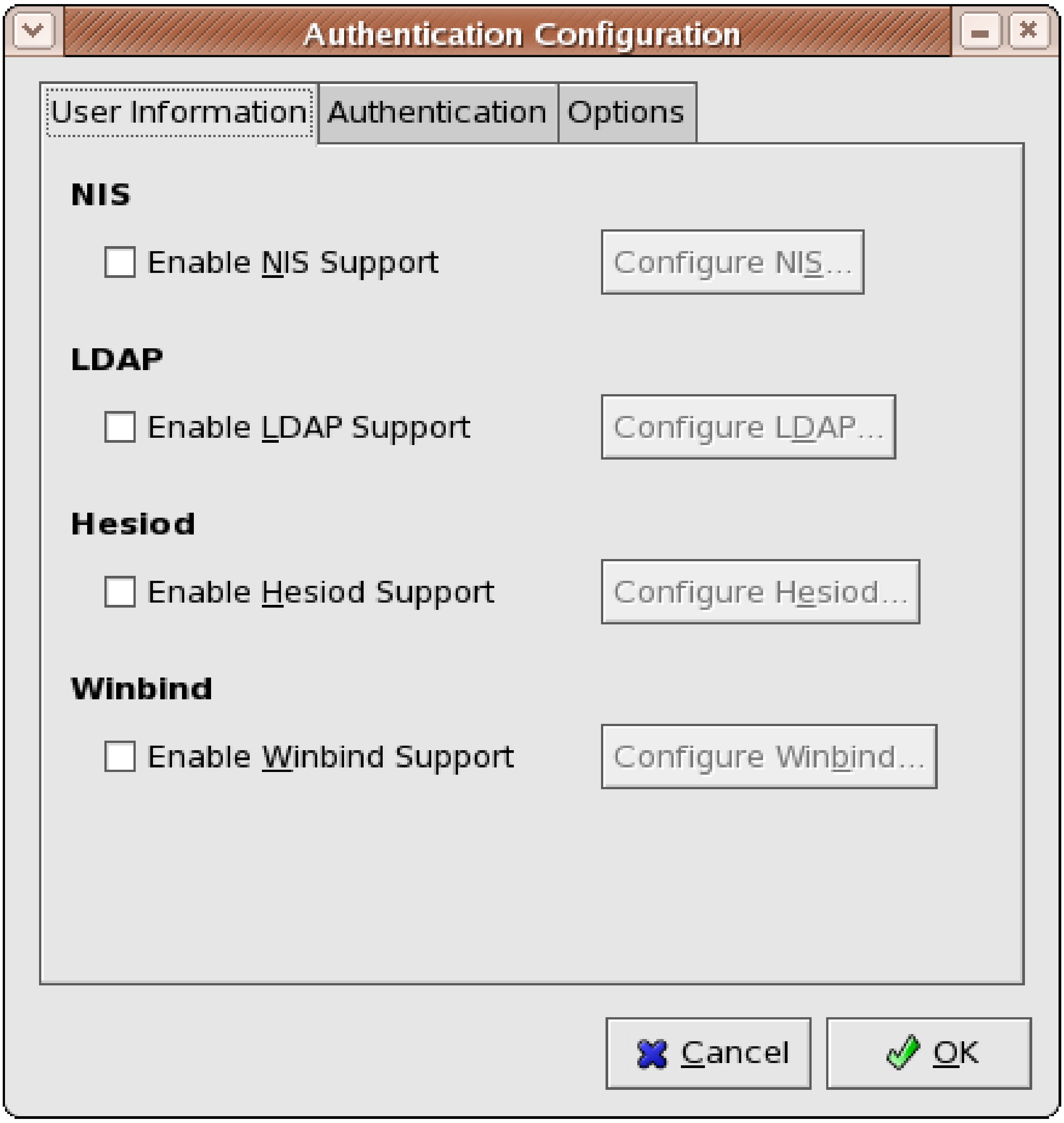
To start the graphical version of the **Authentication Configuration Tool** from the desktop, select the System (on the panel) > **Administration** > **Authentication** or type the command **`system-config-authentication`** at a shell prompt (for example, in an **XTerm** or a **GNOME** terminal).

Important

After exiting the authentication program, the changes made take effect immediately.

15.1. User Information

The **User Information** tab allows you to configure how users should be authenticated, and has several options. To enable an option, click the empty checkbox beside it. To disable an option, click the checkbox beside it to clear the checkbox. Click **OK** to exit the program and apply the changes.



NIS

□ Enable NIS Support

LDAP

□ Enable LDAP Support

Hesiod

□ Enable Hesiod Support

Winbind

□ Enable Winbind Support

Figure 15.1. **User Information**

The following list explains what each option configures:

NIS

The **Enable NIS Support** option configures the system to connect to an NIS server (as an NIS client) for user and password authentication. Click the **Configure NIS...** button to specify the NIS domain and NIS server. If the NIS server is not specified, the daemon attempts to find it via broadcast.

The **`ypbind`** package must be installed for this option to work. If NIS support is enabled, the **`portmap`** and **`ypbind`** services are started and are also enabled to start at boot time.

For more information about NIS, refer to .

LDAP

The **Enable LDAP Support** option instructs the system to retrieve user information via LDAP. Click the **Configure LDAP...** button to specify the following:

- **LDAP Search Base DN** — Specifies that user information should be retrieved using the listed Distinguished Name (DN).
- **LDAP Server** — Specifies the IP address of the LDAP server.
- **Use TLS to encrypt connections** — When enabled, Transport Layer Security will be used to encrypt passwords sent to the LDAP server. The **Download CA Certificate** option allows you to specify a URL from which to download a valid *CA (Certificate Authority) Certificate*. A valid CA Certificate must be in PEM (Privacy Enhanced Mail) format.

  For more information about CA Certificates, refer to *Section 11.8.2, "An Overview of Certificates and Security"*.

The **`openldap-clients`** package must be installed for this option to work.

For more information about LDAP, refer to *Chapter 14, Lightweight Directory Access Protocol (LDAP)*.

Hesiod

The **Enable Hesiod Support** option configures the system to retrieve information (including user information) from a remote Hesiod database. Click the **Configure Hesiod...** button to specify the following:

- **Hesiod LHS** — Specifies the domain prefix used for Hesiod queries.
- **Hesiod RHS** — Specifies the default Hesiod domain.

The **`hesiod`** package must be installed for this option to work.

For more information about Hesiod, refer to its man page using the command **`man hesiod`**. You can also refer to the **`hesiod.conf`** man page (**`man hesiod.conf`**) for more information on LHS and RHS.

Winbind

The **Enable Winbind Support** option configures the system to connect to a Windows Active Directory or a Windows domain controller. User information from the specified directory or domain controller can then be accessed, and server authentication options can be configured. Click the **Configure Winbind...** button to specify the following:

- **Winbind Domain** — Specifies the Windows Active Directory or domain controller to connect to.
- **Security Model** — Allows you to select a security model, which configures how clients should respond to Samba. The drop-down list allows you select any of the following:
  - **user** — This is the default mode. With this level of security, a client must first log in with a valid username and password. Encrypted passwords can also be used in this security mode.
  - **server** — In this mode, Samba will attempt to validate the username/password by authenticating it through another SMB server (for example, a Windows NT Server). If the attempt fails, the **user** mode will take effect instead.

  - **domain** — In this mode, Samba will attempt to validate the username/password by authenticating it through a Windows NT Primary or Backup Domain Controller, similar to how a Windows NT Server would.

  - **ads** — This mode instructs Samba to act as a domain member in an Active Directory Server (ADS) realm. To operate in this mode, the **`krb5-server`** package must be installed, and Kerberos must be configured properly.

- **Winbind ADS Realm** — When the **ads** Security Model is selected, this allows you to specify the ADS Realm the Samba server should act as a domain member of.

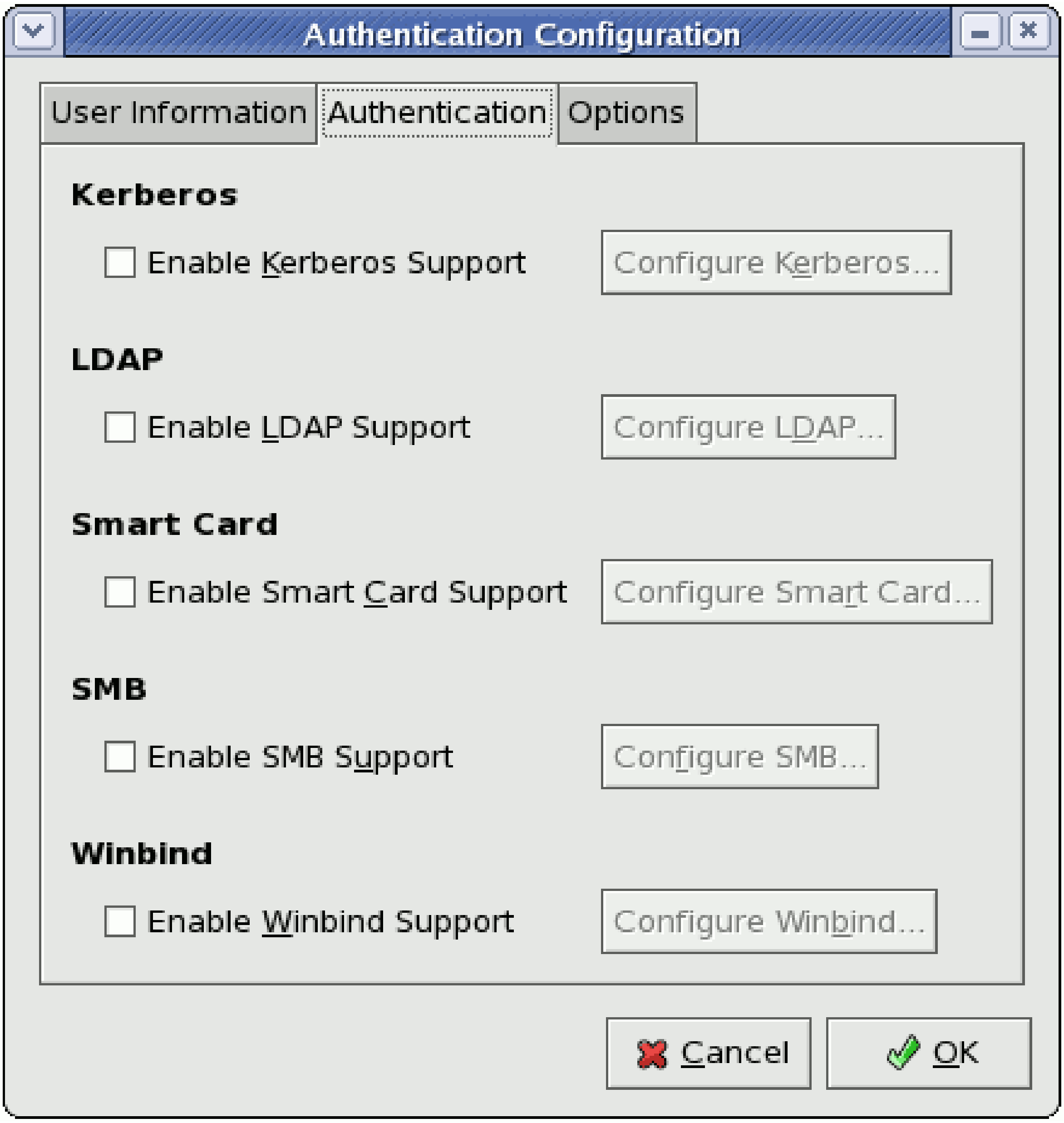
- **Winbind Domain Controllers** — Use this option to specify which domain controller **`winbind`** should use. For more information about domain controllers, please refer to *Section 9.6.3, “Domain Controller”*.

- **Template Shell** — When filling out the user information for a Windows NT user, the **`winbindd`** daemon uses the value chosen here to to specify the login shell for that user.

For more information about the **`winbind`** service, refer to ***`winbindd`*** under *Section 9.2, “Samba Daemons and Related Services”*.

15.2. Authentication

The **Authentication** tab allows for the configuration of network authentication methods. To enable an option, click the empty checkbox beside it. To disable an option, click the checkbox beside it to clear the checkbox.



Kerberos

□ Enable Kerberos Support

LDAP

□ Enable LDAP Support

Smart Card

□ Enable Smart Card Support

SMB

□ Enable SMB Support

Winbind

□ Enable Winbind Support

Figure 15.2. **Authentication**

The following explains what each option configures:

Kerberos

The **Enable Kerberos Support** option enables Kerberos authentication. Click the **Configure Kerberos...** button to open the **Kerberos Settings** dialogue and configure the following:

- **Realm** — Configures the realm for the Kerberos server. The realm is the network that uses Kerberos, composed of one or more KDCs and a potentially large number of clients.
- **KDC** — Defines the Key Distribution Center (KDC), which is the server that issues Kerberos tickets.
- **Admin Servers** — Specifies the administration server(s) running **`kadmind`**.

The **Kerberos Settings** dialogue also allows you to use DNS to resolve hosts to realms and locate KDCs for realms.

The **`krb5-libs`** and **`krb5-workstation`** packages must be installed for this option to work. For more information about Kerberos, refer to .

LDAP

The **Enable LDAP Support** option instructs standard PAM-enabled applications to use LDAP for authentication. The **Configure LDAP...** button allows you to configure LDAP support with options identical to those present in **Configure LDAP...** under the **User Information** tab. For more information about these options, refer to *Section 15.1, "User Information"*.

The **`openldap-clients`** package must be installed for this option to work.

Smart Card

The **Enable Smart Card Support** option enables Smart Card authentication. This allows users to log in using a certificate and key associated stored on a smart card. Click the **Configure Smart Card...** button for more options.

The **`pam_pkcs11`** and **`coolkey`** packages must be installed for this option to work. For more information about smart cards, refer to under .

SMB

The **Enable SMB Support** option configures PAM to use a Server Message Block (SMB) server to authenticate users. SMB refers to a client-server protocol used for cross-system communication; it is also the protocol used by Samba to appear as a Windows server to Windows clients. Click the **Configure SMB...** button to specify the following:

- **Workgroup** — Specifies the SMB workgroup to use.
- **Domain Controllers** — Specifies the SMB domain controllers to use.

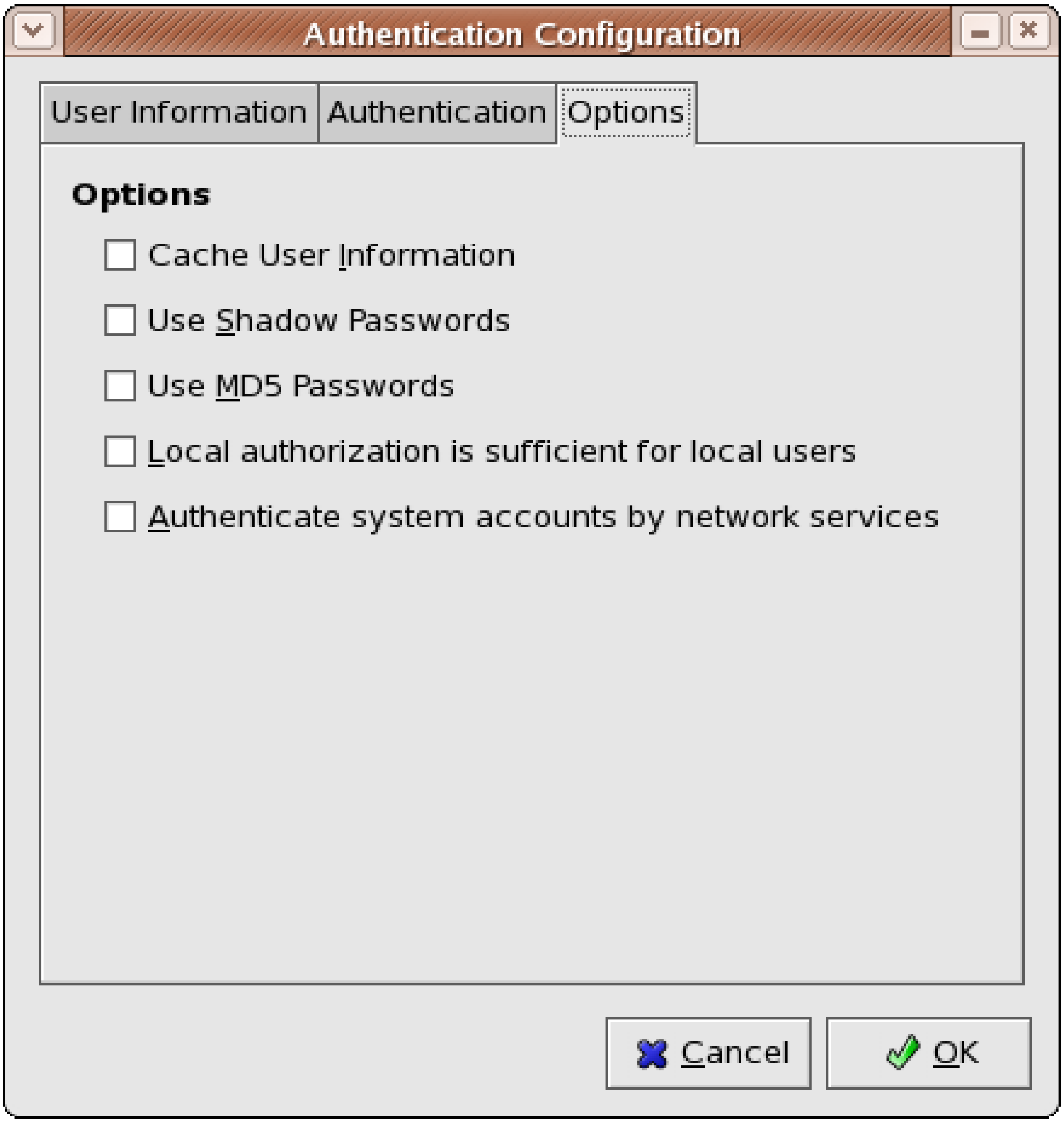
Winbind

The **Enable Winbind Support** option configures the system to connect to a Windows Active Directory or a Windows domain controller. User information from the specified directory or domain controller can then be accessed, and server authentication options can be configured.

The **Configure Winbind...** options are identical to those in the **Configure Winbind...** button on the **User Information** tab. Please refer to *Winbind* (under *Section 15.1, "User Information"*) for more information.

15.3. Options

This tab allows other configuration options, as listed below.



Options

□ Cache User Information

□ Use Shadow Passwords

□ Use MD5 Passwords

□ Local authorization is sufficient for local users

□ Authenticate system accounts by network services

Figure 15.3. **Options**

Cache User Information

Select this option to enable the name service cache daemon (**nscd**) and configure it to start at boot time.

The **nscd** package must be installed for this option to work. For more information about **nscd**, refer to its man page using the command `man nscd`.

Use Shadow Passwords

Select this option to store passwords in shadow password format in the **/etc/shadow** file instead of **/etc/passwd**. Shadow passwords are enabled by default during installation and are highly recommended to increase the security of the system.

The **shadow-utils** package must be installed for this option to work. For more information about shadow passwords, refer to *Section 22.6, “Shadow Passwords”*.

Use MD5 Passwords

Select this option to enable MD5 passwords, which allows passwords to be up to 256 characters instead of eight characters or less. It is selected by default during installation and is highly recommended for increased security.

Local authorization is sufficient for local users

When this option is enabled, the system will not check authorization from network services (such as LDAP or Kerberos) for user accounts maintained in its **/etc/passwd** file.

Authenticate system accounts by network services

Enabling this option configures the system to allow network services (such as LDAP or Kerberos) to authenticate system accounts (including root) in the machine.

15.4. Command Line Version

The **Authentication Configuration Tool** can also be run as a command line tool with no interface. The command line version can be used in a configuration script or a kickstart script. The authentication options are summarized in *Table 15.1, “Command Line Options”*.

Tip

These options can also be found in the **authconfig** man page or by typing **authconfig --help** at a shell prompt.

Option	Description
--enableshadow	Enable shadow passwords
--disableshadow	Disable shadow passwords
--enablemd5	Enable MD5 passwords
--disablemd5	Disable MD5 passwords
--enablenis	Enable NIS
--disablenis	Disable NIS
--nisdomain=<*domain*>	Specify NIS domain
--nisserver=<*server*>	Specify NIS server
--enableldap	Enable LDAP for user information
--disableldap	Disable LDAP for user information
--enableldaptls	Enable use of TLS with LDAP
--disableldaptls	Disable use of TLS with LDAP
--enableldapauth	Enable LDAP for authentication
--disableldapauth	Disable LDAP for authentication
--ldapserver=<*server*>	Specify LDAP server

Option	Description
`--ldapbasedn=<dn>`	Specify LDAP base DN
`--enablekrb5`	Enable Kerberos
`--disablekrb5`	Disable Kerberos
`--krb5kdc=<kdc>`	Specify Kerberos KDC
`--krb5adminserver=<server>`	Specify Kerberos administration server
`--krb5realm=<realm>`	Specify Kerberos realm
`--enablekrb5kdcdns`	Enable use of DNS to find Kerberos KDCs
`--disablekrb5kdcdns`	Disable use of DNS to find Kerberos KDCs
`--enablekrb5realmdns`	Enable use of DNS to find Kerberos realms
`--disablekrb5realmdns`	Disable use of DNS to find Kerberos realms
`--enablesmbauth`	Enable SMB
`--disablesmbauth`	Disable SMB
`--smbworkgroup=<workgroup>`	Specify SMB workgroup
`--smbservers=<server>`	Specify SMB servers
`--enablewinbind`	Enable winbind for user information by default
`--disablewinbind`	Disable winbind for user information by default
`--enablewinbindauth`	Enable winbindauth for authentication by default
`--disablewinbindauth`	Disable winbindauth for authentication by default
`--smbsecurity=<user|server|domain|ads>`	Security mode to use for Samba and winbind
`--smbrealm=<STRING>`	Default realm for Samba and winbind when **`security=ads`**
`--smbidmapuid=<lowest-highest>`	UID range winbind assigns to domain or ADS users
`--smbidmapgid=<lowest-highest>`	GID range winbind assigns to domain or ADS users
`--winbindseparator=<\>`	Character used to separate the domain and user part of winbind usernames if **`winbindusedefaultdomain`** is not enabled
`--winbindtemplatehomedir=</home/%D/%U>`	Directory that winbind users have as their home

Option	Description
`--winbindtemplateprimarygroup=<nobody>`	Group that winbind users have as their primary group
`--winbindtemplateshell=</bin/false>`	Shell that winbind users have as their default login shell
`--enablewinbindusedefaultdomain`	Configures winbind to assume that users with no domain in their usernames are domain users
`--disablewinbindusedefaultdomain`	Configures winbind to assume that users with no domain in their usernames are not domain users
`--winbindjoin=<Administrator>`	Joins the winbind domain or ADS realm now as this administrator
`--enablewins`	Enable WINS for hostname resolution
`--disablewins`	Disable WINS for hostname resolution
`--enablehesiod`	Enable Hesiod
`--disablehesiod`	Disable Hesiod
`--hesiodlhs=<lhs>`	Specify Hesiod LHS
`--hesiodrhs=<rhs>`	Specify Hesiod RHS
`--enablecache`	Enable **`nscd`**
`--disablecache`	Disable **`nscd`**
`--nostart`	Do not start or stop the **`portmap`**, **`ypbind`**, or **`nscd`** services even if they are configured
`--kickstart`	Do not display the user interface
`--probe`	Probe and display network defaults

Table 15.1. Command Line Options

Part III. System Configuration

Part of a system administrator's job is configuring the system for various tasks, types of users, and hardware configurations. This section explains how to configure a Fedora system.

Console Access

When normal (non-root) users log into a computer locally, they are given two types of special permissions:

1. They can run certain programs that they would otherwise be unable to run.

2. They can access certain files (normally special device files used to access diskettes, CD-ROMs, and so on) that they would otherwise be unable to access.

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

In contrast, *every* user who logs in at the console is allowed to run programs that accomplish tasks normally restricted to the root user. If X is running, these actions can be included as menu items in a graphical user interface. As shipped, these console-accessible programs include **`halt`**, **`poweroff`**, and **`reboot`**.

16.1. Disabling Shutdown Via `Ctrl+Alt+Del`

By default, **`/etc/inittab`** specifies that your system is set to shutdown and reboot in response to a **`Ctrl`**+**`Alt`**+**`Del`** key combination used at the console. To completely disable this ability, comment out the following line in **`/etc/inittab`** by putting a hash mark (#) in front of it:

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

Alternatively, you may want to allow certain non-root users the right to shutdown or reboot the system from the console using **`Ctrl`**+**`Alt`**+**`Del`** . You can restrict this privilege to certain users, by taking the following steps:

1. Add the **`-a`** option to the **`/etc/inittab`** line shown above, so that it reads:

   ```
   ca::ctrlaltdel:/sbin/shutdown -a -t3 -r now
   ```

   The **`-a`** flag tells **`shutdown`** to look for the **`/etc/shutdown.allow`** file.

2. Create a file named **`shutdown.allow`** in **`/etc`**. The **`shutdown.allow`** file should list the usernames of any users who are allowed to shutdown the system using **`Ctrl`**+**`Alt`**+**`Del`** . The format of the **`shutdown.allow`** file is a list of usernames, one per line, like the following:

   ```
   stephen jack sophie
   ```

According to this example **`shutdown.allow`** file, the users **`stephen`**, **`jack`**, and **`sophie`** are allowed to shutdown the system from the console using **`Ctrl`**+**`Alt`**+**`Del`** . When that key combination is used, the **`shutdown -a`** command in **`/etc/inittab`** checks to see if any of the users in **`/etc/shutdown.allow`** (or root) are logged in on a virtual console. If one of them is, the shutdown of the system continues; if not, an error message is written to the system console instead.

For more information on **`shutdown.allow`**, refer to the **`shutdown`** man page.

16.2. Disabling Console Program Access

To disable access by users to console programs, run the following command as root:

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

In environments where the console is otherwise secured (BIOS and boot loader passwords are set, **Ctrl**+**Alt**+**Delete** is disabled, the power and reset switches are disabled, and so forth), you may not want to allow any user at the console to run **poweroff**, **halt**, and **reboot**, which are accessible from the console by default.

To disable these abilities, run the following commands as root:

```
rm -f /etc/security/console.apps/poweroff
      rm -f /etc/security/console.apps/halt
      rm -f /etc/security/console.apps/reboot
```

16.3. Defining the Console

The **pam_console.so** module uses the **/etc/security/console.perms** file to determine the permissions for users at the system console. The syntax of the file is very flexible; you can edit the file so that these instructions no longer apply. However, the default file has a line that looks like this:

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

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

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

16.4. Making Files Accessible From the Console

The default settings for individual device classes and permission definitions are defined in **/etc/security/console.perms.d/50-default.perms**. To edit file and device permissions, it is advisable to create a new default file in **/etc/security/console.perms.d/** containing your preferred settings for a specified set of files or devices. The name of the new default file must begin with a number higher than 50 (for example, **51-default.perms**) in order to override **50-default.perms**.

To do this, create a new file named **51-default.perms** in **/etc/security/console.perms.d/**:

```
touch /etc/security/console.perms.d/51-default.perms
```

Open the original default **perms** file, **50-default.perms**. The first section defines *device classes*, with lines similar to the following:

```
<floppy>=/dev/fd[0-1]* \ /dev/floppy/* /mnt/floppy* <sound>=/dev/dsp* /dev/
audio* /dev/midi* \ /dev/mixer* /dev/sequencer \ /dev/sound/* /dev/beep \ /
dev/snd/* <cdrom>=/dev/cdrom* /dev/cdroms/* /dev/cdwriter* /mnt/cdrom*
```

Items enclosed in brackets name the device; in the above example, `<cdrom>` refers to the CD-ROM drive. To add a new device, do not define it in the default **50-default.perms** file; instead, define it in **51-default.perms**. For example, to define a scanner, add the following line to **51-default.perms**:

```
<scanner>=/dev/scanner /dev/usb/scanner*
```

Of course, you must use the appropriate name for the device. Ensure that **/dev/scanner** is really your scanner and not some other device, such as your hard drive.

Once you have properly defined a device or file, the second step is to specify its *permission definitions*. The second section of **/etc/security/console.perms.d/50-default.perms** defines this, with lines similar to the following:

```
<console> 0660 <floppy> 0660 root.floppy <console> 0600 <sound> 0640 root
 <console> 0600 <cdrom> 0600 root.disk
```

To define permissions for a scanner, add a line similar to the following in **51-default.perms**:

```

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

Then, when you log in at the console, you are given ownership of the **/dev/scanner** device with the permissions of 0600 (readable and writable by you only). When you log out, the device is owned by root, and still has the permissions 0600 (now readable and writable by root only).

> **Warning**
>
> You must *never* edit the default **50-default.perms** file. To edit permissions for a device already defined in **50-default.perms**, add the desired permission definition for that device in **51-default.perms**. This will override whatever permissions are defined in **50-default.perms**.

16.5. Enabling Console Access for Other Applications

To make other applications accessible to console users, a bit more work is required.

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

1. Create a link from the name of your application, such as our sample **foo** program, to the **/usr/bin/consolehelper** application:

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

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

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

3. Create a PAM configuration file for the **`foo`** service in **`/etc/pam.d/`**. An easy way to do this is to copy the PAM configuration file of the **`halt`** service, and then modify the copy if you want to change the behavior:

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

Now, when **`/usr/bin/`*`foo`*** is executed, **`consolehelper`** is called, which authenticates the user with the help of **`/usr/sbin/userhelper`**. To authenticate the user, **`consolehelper`** asks for the user's password if **`/etc/pam.d/`*`foo`*** is a copy of **`/etc/pam.d/halt`** (otherwise, it does precisely what is specified in **`/etc/pam.d/`*`foo`***) and then runs **`/usr/sbin/`*`foo`*** with root permissions.

In the PAM configuration file, an application can be configured to use the *pam_timestamp* module to remember (or cache) a successful authentication attempt. When an application is started and proper authentication is provided (the root password), a timestamp file is created. By default, a successful authentication is cached for five minutes. During this time, any other application that is configured to use **`pam_timestamp`** and run from the same session is automatically authenticated for the user — the user does not have to enter the root password again.

This module is included in the **`pam`** package. To enable this feature, add the following lines to your PAM configuration file in **`etc/pam.d/`**:

```
auth include config-util account include config-util session include
 config-util
```

These lines can be copied from any of the **`/etc/pam.d/system-config-*`** configuration files. Note that these lines must be added *below* any other `auth sufficientsession optional` lines in your PAM configuration file.

If an application configured to use **`pam_timestamp`** is successfully authenticated from the Applications (the main menu on the panel), the

🔒

icon is displayed in the notification area of the panel if you are running the **GNOME** or **KDE** desktop environment. After the authentication expires (the default is five minutes), the icon disappears.

The user can select to forget the cached authentication by clicking on the icon and selecting the option to forget authentication.

16.6. The `floppy` Group

If, for whatever reason, console access is not appropriate for you and your non-root users require access to your system's diskette drive, this can be done using the **`floppy`** group. Add the user(s) to the **`floppy`** group using the tool of your choice. For example, the **`gpasswd`** command can be used to add user **`fred`** to the **`floppy`** group:

```
gpasswd -a fred floppy
```

Now, user **`fred`** is able to access the system's diskette drive from the console.

The sysconfig Directory

The **/etc/sysconfig/** directory contains a variety of system configuration files for Fedora.

This chapter outlines some of the files found in the **/etc/sysconfig/** directory, their function, and their contents. The information in this chapter is not intended to be complete, as many of these files have a variety of options that are only used in very specific or rare circumstances.

17.1. Files in the /etc/sysconfig/ Directory

The following sections offer descriptions of files normally found in the **/etc/sysconfig/** directory. Files not listed here, as well as extra file options, are found in the **/usr/share/doc/initscripts-<*version-number*>/sysconfig.txt** file (replace <*version-number*> with the version of the **initscripts** package). Alternatively, looking through the initscripts in the **/etc/rc.d/** directory can prove helpful.

Note

If some of the files listed here are not present in the **/etc/sysconfig/** directory, then the corresponding program may not be installed.

17.1.1. /etc/sysconfig/amd

The **/etc/sysconfig/amd** file contains various parameters used by **amd**; these parameters allow for the automatic mounting and unmounting of file systems.

17.1.2. /etc/sysconfig/apmd

The **/etc/sysconfig/apmd** file is used by **apmd** to configure what power settings to start/stop/change on suspend or resume. This file configures how **apmd** functions at boot time, depending on whether the hardware supports *Advanced Power Management* (*APM*) or whether the user has configured the system to use it. The **apm** daemon is a monitoring program that works with power management code within the Linux kernel. It is capable of alerting users to low battery power on laptops and other power-related settings.

17.1.3. /etc/sysconfig/arpwatch

The **/etc/sysconfig/arpwatch** file is used to pass arguments to the **arpwatch** daemon at boot time. The **arpwatch** daemon maintains a table of Ethernet MAC addresses and their IP address pairings. By default, this file sets the owner of the **arpwatch** process to the user pcap and sends any messages to the **root** mail queue. For more information regarding available parameters for this file, refer to the **arpwatch** man page.

17.1.4. /etc/sysconfig/authconfig

The **/etc/sysconfig/authconfig** file sets the authorization to be used on the host. It contains one or more of the following lines:

- **USEMD5=<*value*>**, where **<*value*>** is one of the following:
  - **yes** — MD5 is used for authentication.

  - **no** — MD5 is not used for authentication.

- **USEKERBEROS=<*value*>**, where **<*value*>** is one of the following:
  - **yes** — Kerberos is used for authentication.
  - **no** — Kerberos is not used for authentication.

- **USELDAPAUTH=<*value*>**, where **<*value*>** is one of the following:
  - **yes** — LDAP is used for authentication.
  - **no** — LDAP is not used for authentication.

17.1.5. /etc/sysconfig/autofs

The **/etc/sysconfig/autofs** file defines custom options for the automatic mounting of devices. This file controls the operation of the automount daemons, which automatically mount file systems when you use them and unmount them after a period of inactivity. File systems can include network file systems, CD-ROMs, diskettes, and other media.

The **/etc/sysconfig/autofs** file may contain the following:

- **LOCALOPTIONS="<*value*>"**, where <*value*> is a string for defining machine-specific automount rules. The default value is an empty string (**""**).

- **DAEMONOPTIONS="<*value*>"**, where <*value*> is the timeout length in seconds before unmounting the device. The default value is 60 seconds (**"--timeout=60"**).

- **UNDERSCORETODOT=<*value*>**, where <*value*> is a binary value that controls whether to convert underscores in file names into dots. For example, **auto_home** to **auto.home** and **auto_mnt** to **auto.mnt**. The default value is 1 (true).

- **DISABLE_DIRECT=<*value*>**, where <*value*> is a binary value that controls whether to disable direct mount support, as the Linux implementation does not conform to the Sun Microsystems' automounter behavior. The default value is 1 (true), and allows for compatibility with the Sun automounter options specification syntax.

17.1.6. /etc/sysconfig/clock

The **/etc/sysconfig/clock** file controls the interpretation of values read from the system hardware clock.

The correct values are:

- **UTC=<*value*>**, where **<*value*>** is one of the following boolean values:
  - **true** or **yes** — The hardware clock is set to Universal Time.
  - **false** or **no** — The hardware clock is set to local time.

- **ARC=<*value*>**, where **<*value*>** is the following:
  - **false** or **no** — This value indicates that the normal UNIX epoch is in use. Other values are used by systems not supported by Fedora.

- **SRM=<*value*>**, where **<*value*>** is the following:
  - **false** or **no** — This value indicates that the normal UNIX epoch is in use. Other values are used by systems not supported by Fedora.
- **ZONE=<*filename*>** — The time zone file under **/usr/share/zoneinfo** that **/etc/localtime** is a copy of. The file contains information such as:

  ```
  ZONE="America/New York"
  ```

  Note that the **ZONE** parameter is read by the **Time and Date Properties Tool** (**system-config-date**), and manually editing it does not change the system timezone.

Earlier releases of Fedora used the following values (which are deprecated):

- **CLOCKMODE=<*value*>**, where **<*value*>** is one of the following:
  - **GMT** — The clock is set to Universal Time (Greenwich Mean Time).
  - **ARC** — The ARC console's 42-year time offset is in effect (for Alpha-based systems only).

17.1.7. /etc/sysconfig/desktop

The **/etc/sysconfig/desktop** file specifies the desktop for new users and the display manager to run when entering runlevel 5.

Correct values are:

- **DESKTOP="<*value*>"**, where **"<*value*>"** is one of the following:
  - **GNOME** — Selects the **GNOME** desktop environment.
  - **KDE** — Selects the **KDE** desktop environment.
- **DISPLAYMANAGER="<*value*>"**, where **"<*value*>"** is one of the following:
  - **GNOME** — Selects the **GNOME Display Manager**.
  - **KDE** — Selects the **KDE Display Manager**.
  - **XDM** — Selects the **X Display Manager**.

For more information, refer to *Chapter 20, The X Window System*.

17.1.8. /etc/sysconfig/dhcpd

The **/etc/sysconfig/dhcpd** file is used to pass arguments to the **dhcpd** daemon at boot time. The **dhcpd** daemon implements the Dynamic Host Configuration Protocol (DHCP) and the Internet Bootstrap Protocol (BOOTP). DHCP and BOOTP assign hostnames to machines on the network. For more information about what parameters are available in this file, refer to the **dhcpd** man page.

17.1.9. /etc/sysconfig/exim

The **/etc/sysconfig/exim** file allows messages to be sent to one or more clients, routing the messages over whatever networks are necessary. The file sets the default values for exim to run.

Its default values are set to run as a background daemon and to check its queue each hour in case something has backed up.

The values include:

- **`DAEMON=<value>`**, where **`<value>`** is one of the following:
  - **`yes`** — **`exim`** should be configured to listen to port 25 for incoming mail. **`yes`** implies the use of the Exim's **`-bd`** options.
  - **`no`** — **`exim`** should not be configured to listen to port 25 for incoming mail.
- **`QUEUE=1h`** which is given to **`exim`** as **`-q$QUEUE`**. The **`-q`** option is not given to **`exim`** if **`/etc/sysconfig/exim`** exists and **`QUEUE`** is empty or undefined.

17.1.10. `/etc/sysconfig/firstboot`

The first time the system boots, the **`/sbin/init`** program calls the **`etc/rc.d/init.d/firstboot`** script, which in turn launches the **Setup Agent**. This application allows the user to install the latest updates as well as additional applications and documentation.

The **`/etc/sysconfig/firstboot`** file tells the **Setup Agent** application not to run on subsequent reboots. To run it the next time the system boots, remove **`/etc/sysconfig/firstboot`** and execute **`chkconfig --level 5 firstboot on`**.

17.1.11. `/etc/sysconfig/gpm`

The **`/etc/sysconfig/gpm`** file is used to pass arguments to the **`gpm`** daemon at boot time. The **`gpm`** daemon is the mouse server which allows mouse acceleration and middle-click pasting. For more information about what parameters are available for this file, refer to the **`gpm`** man page. By default, the **`DEVICE`** directive is set to **`/dev/input/mice`**.

17.1.12. `/etc/sysconfig/hwconf`

The **`/etc/sysconfig/hwconf`** file lists all the hardware that **`kudzu`** detected on the system, as well as the drivers used, vendor ID, and device ID information. The **`kudzu`** program detects and configures new and/or changed hardware on a system. The **`/etc/sysconfig/hwconf`** file is not meant to be manually edited. If edited, devices could suddenly show up as being added or removed.

17.1.13. `/etc/sysconfig/i18n`

The **`/etc/sysconfig/i18n`** file sets the default language, any supported languages, and the default system font. For example:

```
LANG="en_US.UTF-8"
SUPPORTED="en_US.UTF-8:en_US:en"
SYSFONT="latarcyrheb-sun16"
```

17.1.14. `/etc/sysconfig/init`

The **`/etc/sysconfig/init`** file controls how the system appears and functions during the boot process.

The following values may be used:

- **BOOTUP=<*value*>**, where **<*value*>** is one of the following:
  - **color** — The standard color boot display, where the success or failure of devices and services starting up is shown in different colors.
  - **verbose** — An old style display which provides more information than purely a message of success or failure.
  - Anything else means a new display, but without ANSI-formatting.
- **RES_COL=<*value*>**, where **<*value*>** is the number of the column of the screen to start status labels. The default is set to 60.
- **MOVE_TO_COL=<*value*>**, where **<*value*>** moves the cursor to the value in the **RES_COL** line via the **echo -en** command.
- **SETCOLOR_SUCCESS=<*value*>**, where **<*value*>** sets the success color via the **echo -en** command. The default color is set to green.
- **SETCOLOR_FAILURE=<*value*>**, where **<*value*>** sets the failure color via the **echo -en** command. The default color is set to red.
- **SETCOLOR_WARNING=<*value*>**, where **<*value*>** sets the warning color via the **echo -en** command. The default color is set to yellow.
- **SETCOLOR_NORMAL=<*value*>**, where **<*value*>** resets the color to "normal" via the **echo -en**.
- **LOGLEVEL=<*value*>**, where **<*value*>** sets the initial console logging level for the kernel. The default is 3; 8 means everything (including debugging), while 1 means only kernel panics. The **syslogd** daemon overrides this setting once started.
- **PROMPT=<*value*>**, where **<*value*>** is one of the following boolean values:
  - **yes** — Enables the key check for interactive mode.
  - **no** — Disables the key check for interactive mode.

17.1.15. /etc/sysconfig/ip6tables-config

The **/etc/sysconfig/ip6tables-config** file stores information used by the kernel to set up IPv6 packet filtering at boot time or whenever the **ip6tables** service is started.

Do not modify this file by hand unless familiar with how to construct **ip6tables** rules. Rules also can be created manually using the **/sbin/ip6tables** command. Once created, add the rules to the **/etc/sysconfig/ip6tables** file by typing the following command:

```
/sbin/service ip6tables save
```

Once this file exists, any firewall rules saved in it persists through a system reboot or a service restart.

For more information on **ip6tables**, refer to .

17.1.16. /etc/sysconfig/iptables-config

The **/etc/sysconfig/iptables-config** file stores information used by the kernel to set up packet filtering services at boot time or whenever the service is started.

Do not modify this file by hand unless you are familiar with constructing **iptables** rules. The easiest way to add rules is to use the **Security Level Configuration Tool** (**system-config-securitylevel**) application to create a firewall. These applications automatically edit this file at the end of the process.

Rules can also be created manually using the **/sbin/iptables** command. Once created, add the rule(s) to the **/etc/sysconfig/iptables** file by typing the following command:

```
/sbin/service iptables save
```

Once this file exists, any firewall rules saved in it persists through a system reboot or a service restart.

For more information on **iptables**, refer to .

17.1.17. /etc/sysconfig/irda

The **/etc/sysconfig/irda** file controls how infrared devices on the system are configured at startup.

The following values may be used:

- **IRDA=<*value*>**, where **<*value*>** is one of the following boolean values:
  - **yes** — **irattach** runs and periodically checks to see if anything is trying to connect to the infrared port, such as another notebook computer trying to make a network connection. For infrared devices to work on the system, this line must be set to **yes**.
  - **no** — **irattach** does not run, preventing infrared device communication.
- **DEVICE=<*value*>**, where **<*value*>** is the device (usually a serial port) that handles infrared connections. A sample serial device entry could be **/dev/ttyS2**.
- **DONGLE=<*value*>**, where **<*value*>** specifies the type of dongle being used for infrared communication. This setting exists for people who use serial dongles rather than real infrared ports. A dongle is a device that is attached to a traditional serial port to communicate via infrared. This line is commented out by default because notebooks with real infrared ports are far more common than computers with add-on dongles. A sample dongle entry could be **actisys+**.
- **DISCOVERY=<*value*>**, where **<*value*>** is one of the following boolean values:
  - **yes** — Starts **irattach** in discovery mode, meaning it actively checks for other infrared devices. This must be turned on for the machine to actively look for an infrared connection (meaning the peer that does not initiate the connection).
  - **no** — Does not start **irattach** in discovery mode.

17.1.18. `/etc/sysconfig/keyboard`

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

- **`KEYBOARDTYPE="sun|pc"`** where **`sun`** means a Sun keyboard is attached on **`/dev/kbd`**, or **`pc`** means a PS/2 keyboard connected to a PS/2 port.

- **`KEYTABLE="<file>"`**, where **`<file>`** is the name of a keytable file.

  For example: **`KEYTABLE="us"`**. The files that can be used as keytables start in **`/lib/kbd/keymaps/i386`** and branch into different keyboard layouts from there, all labeled **`<file>.kmap.gz`**. The first file found beneath **`/lib/kbd/keymaps/i386`** that matches the **`KEYTABLE`** setting is used.

17.1.19. `/etc/sysconfig/kudzu`

The **`/etc/sysconfig/kuzdu`** file triggers a safe probe of the system hardware by **`kudzu`** at boot time. A safe probe is one that disables serial port probing.

- **`SAFE=<value>`**, where **`<value>`** is one of the following:

  - **`yes`** — **`kuzdu`** does a safe probe.

  - **`no`** — **`kuzdu`** does a normal probe.

17.1.20. `/etc/sysconfig/named`

The **`/etc/sysconfig/named`** file is used to pass arguments to the **`named`** daemon at boot time. The **`named`** daemon is a *Domain Name System* (*DNS*) server which implements the *Berkeley Internet Name Domain* (*BIND*) version 9 distribution. This server maintains a table of which hostnames are associated with IP addresses on the network.

Currently, only the following values may be used:

- **`ROOTDIR="</some/where>"`**, where **`</some/where>`** refers to the full directory path of a configured chroot environment under which **`named`** runs. This chroot environment must first be configured. Type **`info chroot`** for more information.

- **`OPTIONS="<value>"`**, where **`<value>`** is any option listed in the man page for **`named`** except `-t`. In place of `-t`, use the **`ROOTDIR`** line above.

For more information about available parameters for this file, refer to the **`named`** man page. For detailed information on how to configure a BIND DNS server, refer to *Chapter 7, Berkeley Internet Name Domain (BIND)*. By default, the file contains no parameters.

17.1.21. `/etc/sysconfig/network`

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

- **`NETWORKING=<value>`**, where **`<value>`** is one of the following boolean values:

  - **`yes`** — Networking should be configured.

  - **`no`** — Networking should not be configured.

- **HOSTNAME=<*value*>**, where **<*value*>** should be the *Fully Qualified Domain Name* (*FQDN*), such as **hostname.expample.com**, but can be whatever hostname is necessary.
- **GATEWAY=<*value*>**, where **<*value*>** is the IP address of the network's gateway.
- **GATEWAYDEV=<*value*>**, where **<*value*>** is the gateway device, such as **eth0**. Configure this option if you have multiple interfaces on the same subnet, and require one of those interfaces to be the preferred route to the default gateway.
- **NISDOMAIN=<*value*>**, where **<*value*>** is the NIS domain name.
- **NOZEROCONF=<*value*>**, where setting **<*value*>** to **true** disables the zeroconf route.

  By default, the zeroconf route (169.254.0.0) is enabled when the system boots. For more information about zeroconf, refer to http://www.zeroconf.org/.

Warning

Do not use custom initscripts to configure network settings. When performing a post-boot network service restart, custom initscripts configuring network settings that are run outside of the network init script lead to unpredictable results.

17.1.22. /etc/sysconfig/ntpd

The **/etc/sysconfig/ntpd** file is used to pass arguments to the **ntpd** daemon at boot time. The **ntpd** daemon sets and maintains the system clock to synchronize with an Internet standard time server. It implements version 4 of the Network Time Protocol (NTP). For more information about what parameters are available for this file, use a Web browser to view the following file: **/usr/share/doc/ntp-<*version*>/ntpd.htm** (where <*version*> is the version number of **ntpd**). By default, this file sets the owner of the **ntpd** process to the user ntp.

17.1.23. /etc/sysconfig/radvd

The **/etc/sysconfig/radvd** file is used to pass arguments to the **radvd** daemon at boot time. The **radvd** daemon listens for router requests and sends router advertisements for the IP version 6 protocol. This service allows hosts on a network to dynamically change their default routers based on these router advertisements. For more information about available parameters for this file, refer to the **radvd** man page. By default, this file sets the owner of the **radvd** process to the user radvd.

17.1.24. /etc/sysconfig/samba

The **/etc/sysconfig/samba** file is used to pass arguments to the **smbd** and the **nmbd** daemons at boot time. The **smbd** daemon offers file sharing connectivity for Windows clients on the network. The **nmbd** daemon offers NetBIOS over IP naming services. For more information about what parameters are available for this file, refer to the **smbd** man page. By default, this file sets **smbd** and **nmbd** to run in daemon mode.

17.1.25. /etc/sysconfig/selinux

The **/etc/sysconfig/selinux** file contains the basic configuration options for SELinux. This file is a symbolic link to **/etc/selinux/config**.

17.1.26. /etc/sysconfig/sendmail

The **/etc/sysconfig/sendmail** file allows messages to be sent to one or more clients, routing the messages over whatever networks are necessary. The file sets the default values for the **Sendmail** application to run. Its default values are set to run as a background daemon and to check its queue each hour in case something has backed up.

Values include:

- **DAEMON=<*value*>**, where **<*value*>** is one of the following:
  - **yes** — **Sendmail** should be configured to listen to port 25 for incoming mail. **yes** implies the use of **Sendmail**'s **-bd** options.
  - **no** — **Sendmail** should not be configured to listen to port 25 for incoming mail.
- **QUEUE=1h** which is given to **Sendmail** as **-q$QUEUE**. The **-q** option is not given to **Sendmail** if **/etc/sysconfig/sendmail** exists and **QUEUE** is empty or undefined.

17.1.27. /etc/sysconfig/spamassassin

The **/etc/sysconfig/spamassassin** file is used to pass arguments to the **spamd** daemon (a daemonized version of **Spamassassin**) at boot time. **Spamassassin** is an email spam filter application. For a list of available options, refer to the **spamd** man page. By default, it configures **spamd** to run in daemon mode, create user preferences, and auto-create whitelists (allowed bulk senders).

For more information about **Spamassassin**, refer to *Section 13.5.2.6, “Spam Filters”*.

17.1.28. /etc/sysconfig/squid

The **/etc/sysconfig/squid** file is used to pass arguments to the **squid** daemon at boot time. The **squid** daemon is a proxy caching server for Web client applications. For more information on configuring a **squid** proxy server, use a Web browser to open the **/usr/share/doc/squid-<*version*>/** directory (replace <*version*> with the **squid** version number installed on the system). By default, this file sets **squid** to start in daemon mode and sets the amount of time before it shuts itself down.

17.1.29. /etc/sysconfig/system-config-securitylevel

The **/etc/sysconfig/system-config-securitylevel** file contains all options chosen by the user the last time the **Security Level Configuration Tool** (**system-config-securitylevel**) was run. Users should not modify this file by hand. For more information about the **Security Level Configuration Tool**, refer to .

17.1.30. /etc/sysconfig/system-config-selinux

The **/etc/sysconfig/system-config-selinux** file contains all options chosen by the user the last time the **SELinux Administration Tool** (**system-config-selinux**) was run. Users should not modify this file by hand. For more information about the **SELinux Administration Tool** and SELinux in general, refer to .

17.1.31. `/etc/sysconfig/system-config-users`

The **`/etc/sysconfig/system-config-users`** file is the configuration file for the graphical application, **User Manager**. This file is used to filter out system users such as **`root`**, **`daemon`**, or **`lp`**. This file is edited by the **Preferences** > **Filter system users and groups** pull-down menu in the **User Manager** application and should never be edited by hand. For more information on using this application, refer to *Section 22.1, “User and Group Configuration”*.

17.1.32. `/etc/sysconfig/system-logviewer`

The **`/etc/sysconfig/system-logviewer`** file is the configuration file for the graphical, interactive log viewing application, **Log Viewer**. This file is edited by the **Edit** > **Preferences** pull-down menu in the **Log Viewer** application and should not be edited by hand. For more information on using this application, refer to *Chapter 25, Log Files*.

17.1.33. `/etc/sysconfig/tux`

The **`/etc/sysconfig/tux`** file is the configuration file for the Red Hat Content Accelerator (formerly known as **TUX**), the kernel-based Web server. For more information on configuring the Red Hat Content Accelerator, use a Web browser to open the **`/usr/share/doc/tux-<*version*>/tux/index.html`** file (replace <*version*> with the version number of **TUX** installed on the system). The parameters available for this file are listed in **`/usr/share/doc/tux-<*version*>/tux/parameters.html`**.

17.1.34. `/etc/sysconfig/vncservers`

The **`/etc/sysconfig/vncservers`** file configures the way the *Virtual Network Computing* (*VNC*) server starts up.

VNC is a remote display system which allows users to view the desktop environment not only on the machine where it is running but across different networks on a variety of architectures.

It may contain the following:

- **`VNCSERVERS=<*value*>`**, where **<*value*>** is set to something like **`"1:fred"`**, to indicate that a VNC server should be started for user fred on display :1. User fred must have set a VNC password using the **`vncpasswd`** command before attempting to connect to the remote VNC server.

17.1.35. `/etc/sysconfig/xinetd`

The **`/etc/sysconfig/xinetd`** file is used to pass arguments to the **`xinetd`** daemon at boot time. The **`xinetd`** daemon starts programs that provide Internet services when a request to the port for that service is received. For more information about available parameters for this file, refer to the **`xinetd`** man page. For more information on the **`xinetd`** service, refer to .

17.2. Directories in the `/etc/sysconfig/` Directory

The following directories are normally found in **`/etc/sysconfig/`**.

`apm-scripts/`

This directory contains the APM suspend/resume script. Do not edit the files directly. If customization is necessary, create a file called **`/etc/sysconfig/apm-scripts/`**

apmcontinue which is called at the end of the script. It is also possible to control the script by editing **/etc/sysconfig/apmd**.

cbq/

This directory contains the configuration files needed to do *Class Based Queuing* for bandwidth management on network interfaces. CBQ divides user traffic into a hierarchy of classes based on any combination of IP addresses, protocols, and application types.

networking/

This directory is used by the **Network Administration Tool** (**system-config-network**), and its contents should not be edited manually. For more information about configuring network interfaces using the **Network Administration Tool**, refer to *Chapter 5, Network Configuration*.

network-scripts/

This directory contains the following network-related configuration files:

- Network configuration files for each configured network interface, such as **ifcfg-eth0** for the **eth0** Ethernet interface.
- Scripts used to bring network interfaces up and down, such as **ifup** and **ifdown**.
- Scripts used to bring ISDN interfaces up and down, such as **ifup-isdn** and **ifdown-isdn**.
- Various shared network function scripts which should not be edited directly.

For more information on the **network-scripts** directory, refer to *Chapter 4, Network Interfaces*.

rhn/

This directory contains the configuration files and GPG keys for Red Hat Network. No files in this directory should be edited by hand. For more information on Red Hat Network, refer to the Red Hat Network website online at *https://rhn.redhat.com/*.

17.3. Additional Resources

This chapter is only intended as an introduction to the files in the **/etc/sysconfig/** directory. The following source contains more comprehensive information.

17.3.1. Installed Documentation

- **/usr/share/doc/initscripts-<*version-number*>/sysconfig.txt** — This file contains a more authoritative listing of the files found in the **/etc/sysconfig/** directory and the configuration options available for them. The <*version-number*> in the path to this file corresponds to the version of the **initscripts** package installed.

Date and Time Configuration

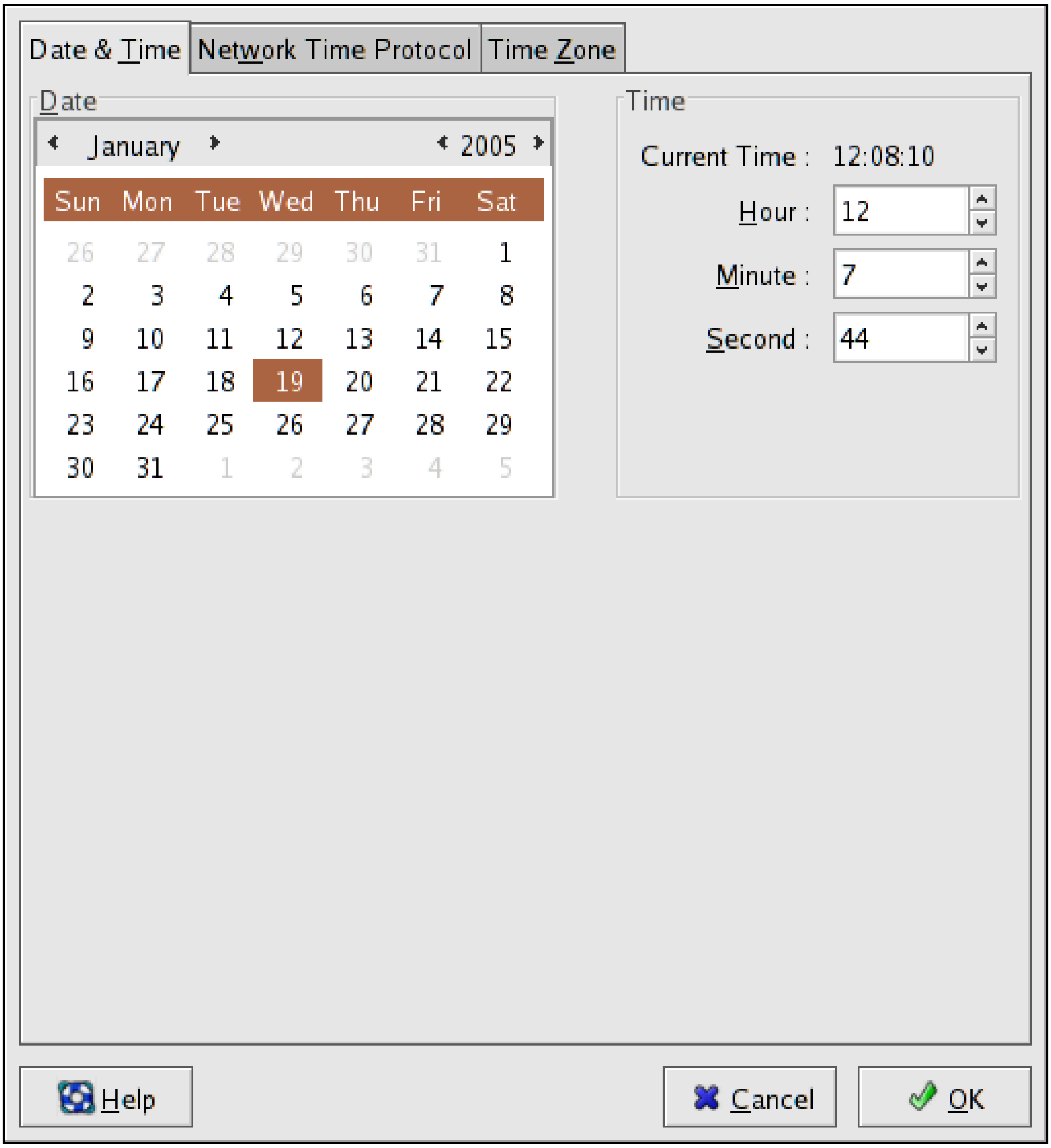
The **Time and Date Properties Tool** allows the user to change the system date and time, to configure the time zone used by the system, and to setup the Network Time Protocol (NTP) daemon to synchronize the system clock with a time server.

You must be running the **X** Window System and have root privileges to use the tool. There are three ways to start the application:

- From the desktop, go to Applications (the main menu on the panel) > **System Settings** > **Date & Time**
- From the desktop, right-click on the time in the toolbar and select **Adjust Date and Time**.
- Type the command **`system-config-date`**, **`system-config-time`**, or **`dateconfig`** at a shell prompt (for example, in an **XTerm** or a **GNOME** terminal).

18.1. Time and Date Properties

As shown in *Figure 18.1, “Time and Date Properties”*, the first tabbed window that appears is for configuring the system date and time.



January 2005

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Current Time : 12:08:10

Hour : 12

Minute : 7

Second : 44

Figure 18.1. Time and Date Properties

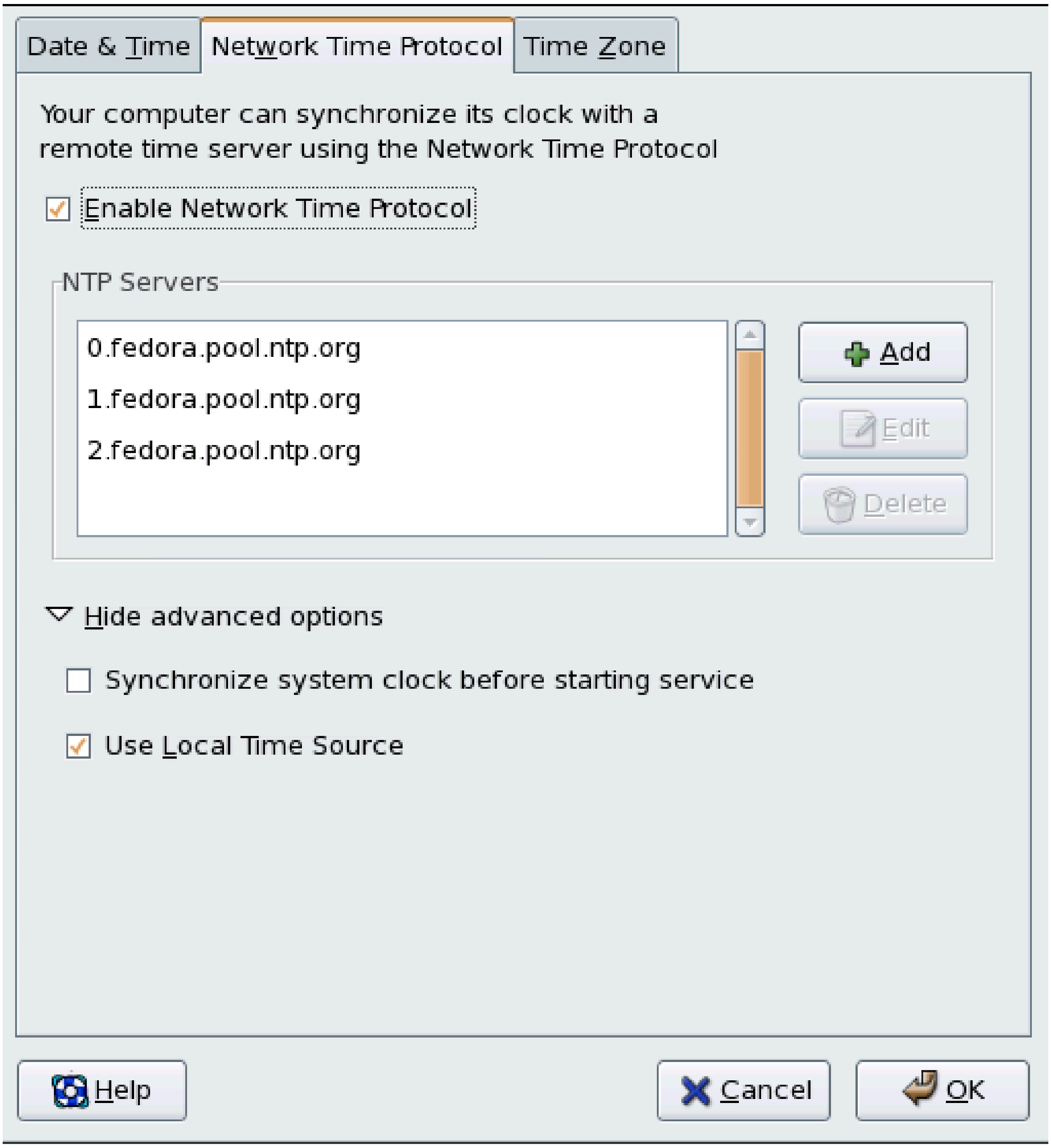
To change the date, use the arrows to the left and right of the month to change the month, use the arrows to the left and right of the year to change the year, and click on the day of the week to change the day of the week.

To change the time, use the up and down arrow buttons beside the **Hour**, **Minute**, and **Second** in the **Time** section.

Clicking the **OK** button applies any changes made to the date and time, the NTP daemon settings, and the time zone settings. It also exits the program.

18.2. Network Time Protocol (NTP) Properties

As shown in *Figure 18.2, “NTP Properties”*, the second tabbed window that appears is for configuring NTP.



Your computer can synchronize its clock with a remote time server using the Network Time Protocol

☑ Enable Network Time Protocol

0.fedora.pool.ntp.org
1.fedora.pool.ntp.org
2.fedora.pool.ntp.org

□ Synchronize system clock before starting service

☑ Use Local Time Source

Figure 18.2. NTP Properties

The Network Time Protocol (NTP) daemon synchronizes the system clock with a remote time server or time source. The application allows you to configure an NTP daemon to synchronize your system clock with a remote server. To enable this feature, select **Enable Network Time Protocol**. This enables the **NTP Servers** list and other options. You can choose one of the predefined servers, edit a predefined server by clicking the **Edit** or add a new server name by clicking **Add**. Your system does not start synchronizing with the NTP server until you click **OK**. After clicking **OK**, the configuration is saved and the NTP daemon is started (or restarted if it is already running).

Clicking the **OK** button applies any changes made to the date and time, the NTP daemon settings, and the time zone settings. It also exits the program.

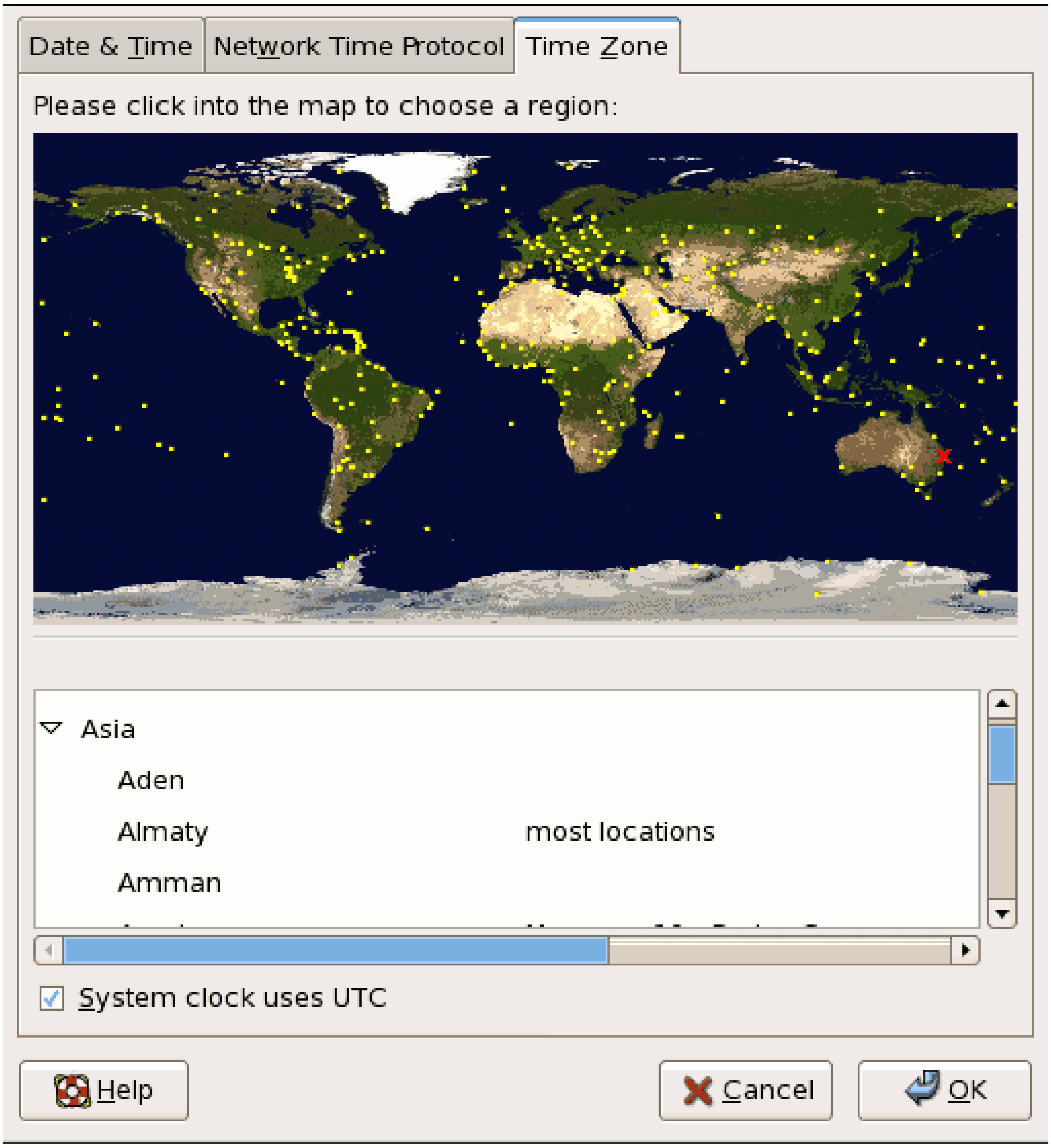
18.3. Time Zone Configuration

As shown in *Figure 18.3, “Timezone Properties”*, the third tabbed window that appears is for configuring the system time zone.

To configure the system time zone, click the **Time Zone** tab. The time zone can be changed by either using the interactive map or by choosing the desired time zone from the list below the map. To use the map, click on the desired region. The map zooms into the region selected, after which you may choose the city specific to your time zone. A red **X** appears and the time zone selection changes in the list below the map.

Alternatively, you can also use the list below the map. In the same way that the map lets you choose a region before choosing a city, the list of time zones is now a treelist, with cities and countries grouped within their specific continents. Non-geographic time zones have also been added to address needs in the scientific community.

Click **OK** to apply the changes and exit the program.



Please click into the map to choose a region:

- Asia
  - Aden
  - Almaty — most locations
  - Amman

☑ System clock uses UTC

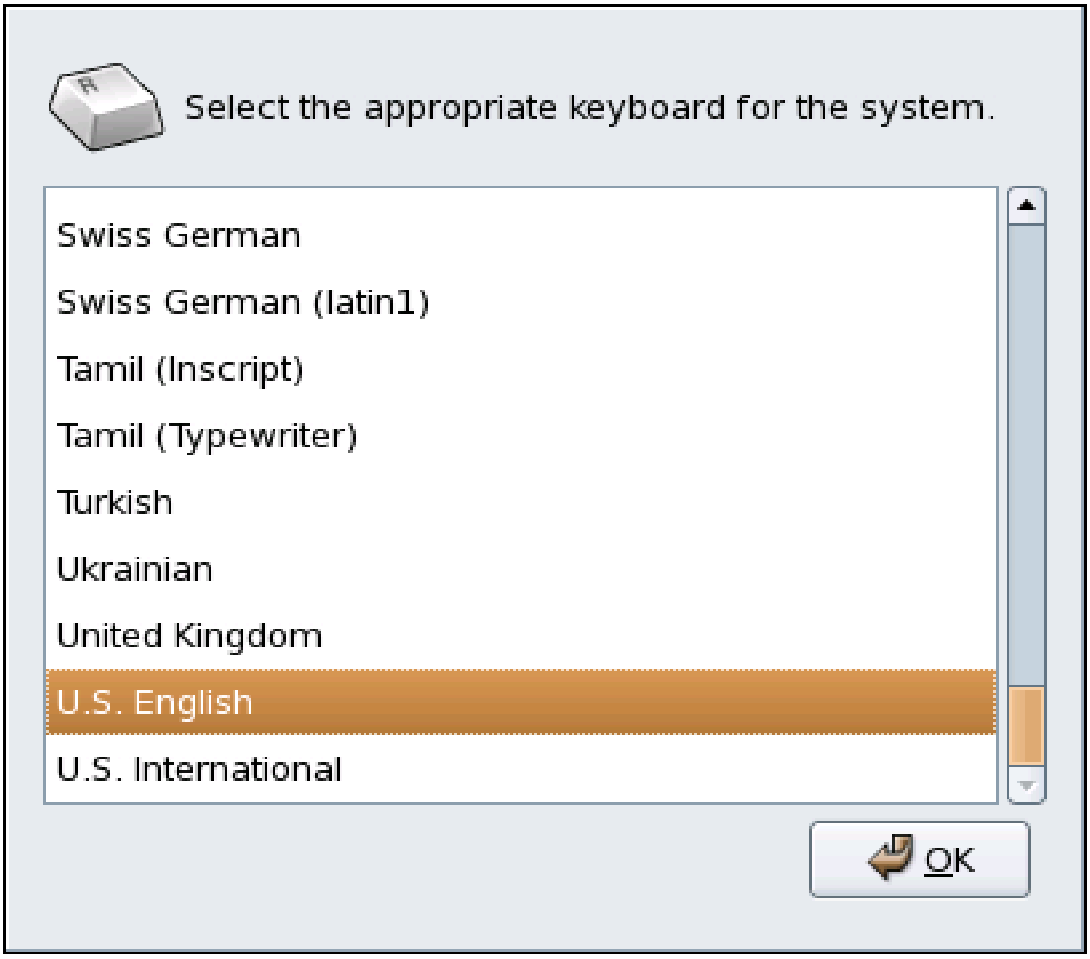
Figure 18.3. Timezone Properties

If your system clock is set to use UTC, select the **System clock uses UTC** option. UTC stands for the *Universal Time, Coordinated*, also known as Greenwich Mean Time (GMT). Other time zones are determined by adding or subtracting from the UTC time.

Keyboard Configuration

The installation program allows you to configure a keyboard layout for your system. To configure a different keyboard layout after installation, use the **Keyboard Configuration Tool**.

To start the **Keyboard Configuration Tool**, select System (on the panel) > **Administration** > **Keyboard**, or type the command **`system-config-keyboard`** at a shell prompt.



Select the appropriate keyboard for the system.

- Swiss German
- Swiss German (latin1)
- Tamil (Inscript)
- Tamil (Typewriter)
- Turkish
- Ukrainian
- United Kingdom
- U.S. English
- U.S. International

Figure 19.1. **Keyboard Configuration Tool**

Select a keyboard layout from the list (for example, **U.S. English**) and click **OK**.

Changes take effect immediately.

The X Window System

While the heart of Fedora is the kernel, for many users, the face of the operating system is the graphical environment provided by the *X Window System*, also called *X*.

Other windowing environments have existed in the UNIX world, including some that predate the release of the X Window System in June 1984. Nonetheless, X has been the default graphical environment for most UNIX-like operating systems, including Fedora, for many years.

The graphical environment for Fedora is supplied by the *X.Org Foundation*, an open source organization created to manage development and strategy for the X Window System and related technologies. X.Org is a large-scale, rapidly developing project with hundreds of developers around the world. It features a wide degree of support for a variety of hardware devices and architectures, and can run on a variety of different operating systems and platforms. This release for Fedora specifically includes the X11R7.1 release of the X Window System.

The X Window System uses a client-server architecture. The *X server* (the **`Xorg`** binary) listens for connections from *X client* applications via a network or local loopback interface. The server communicates with the hardware, such as the video card, monitor, keyboard, and mouse. X client applications exist in the user-space, creating a *graphical user interface* (*GUI*) for the user and passing user requests to the X server.

20.1. The X11R7.1 Release

Fedora 12 uses the X11R7.1 release as the base X Window System, which includes several video driver, EXA, and platform support enhancements over the previous release, among others. In addition, this release also includes several automatic configuration features for the X server.

X11R7.1 is the first release to take specific advantage of the modularization of the X Window System. This modularization, which splits X into logically distinct modules, makes it easier for open source developers to contribute code to the system.

Important

Fedora no longer provides the XFree86™ server packages. Before upgrading a system to the latest version of Fedora, be sure that the system's video card is compatible with the X11R7.1 release by checking the Red Hat Hardware Compatibility List located online at *http://hardware.redhat.com/*.

In the X11R7.1 release, all libraries, headers, and binaries now live under **`/usr/`** instead of **`/usr/X11R6`**. The **`/etc/X11/`** directory contains configuration files for X client and server applications. This includes configuration files for the X server itself, the **`xfs`** font server, the X display managers, and many other base components.

The configuration file for the newer Fontconfig-based font architecture is still **`/etc/fonts/fonts.conf`**. For more on configuring and adding fonts, refer to *Section 20.4, “Fonts”*.

Because the X server performs advanced tasks on a wide array of hardware, it requires detailed information about the hardware it works on. The X server automatically detects some of this information; other details must be configured.

The installation program installs and configures X automatically, unless the X11R7.1 release packages are not selected for installation. However, if there are any changes to the monitor, video card or other devices managed by the X server, X must be reconfigured. The best way to do this is to use the **X Configuration Tool** (`system-config-display`), particularly for devices that are not detected manually.

In Fedora's default graphical environment, the **X Configuration Tool** is available at System (on the panel) > **Administration** > **Display**.

Changes made with the **X Configuration Tool** take effect after logging out and logging back in.

For more information about **X Configuration Tool**, refer to *Chapter 21, X Window System Configuration*.

In some situations, reconfiguring the X server may require manually editing its configuration file, `/etc/X11/xorg.conf`. For information about the structure of this file, refer to *Section 20.3, "X Server Configuration Files"*.

20.2. Desktop Environments and Window Managers

Once an X server is running, X client applications can connect to it and create a GUI for the user. A range of GUIs are possible with Fedora, from the rudimentary *Tab Window Manager* to the highly developed and interactive *GNOME* desktop environment that most Fedora users are familiar with.

To create the latter, more comprehensive GUI, two main classes of X client application must connect to the X server: a *desktop environment* and a *window manager*.

20.2.1. Desktop Environments

A desktop environment integrates various X clients to create a common graphical user environment and development platform.

Desktop environments have advanced features allowing X clients and other running processes to communicate with one another, while also allowing all applications written to work in that environment to perform advanced tasks, such as drag and drop operations.

Fedora provides two desktop environments:

- *GNOME* — The default desktop environment for Fedora based on the GTK+ 2 graphical toolkit.
- *KDE* — An alternative desktop environment based on the Qt 4 graphical toolkit.

Both GNOME and KDE have advanced productivity applications, such as word processors, spreadsheets, and Web browsers; both also provide tools to customize the look and feel of the GUI. Additionally, if both the GTK+ 2 and the Qt libraries are present, KDE applications can run in GNOME and vice-versa.

20.2.2. Window Managers

Window managers are X client programs which are either part of a desktop environment or, in some cases, stand-alone. Their primary purpose is to control the way graphical windows are positioned, resized, or moved. Window managers also control title bars, window focus behavior, and user-specified key and mouse button bindings.

Four window managers are included with Fedora:

kwin

The *KWin* window manager is the default window manager for KDE. It is an efficient window manager which supports custom themes.

metacity

The *Metacity* window manager is the default window manager for GNOME. It is a simple and efficient window manager which also supports custom themes. To run this window manager, you need to install the **metacity** package.

mwm

The *Motif Window Manager* (**mwm**) is a basic, stand-alone window manager. Since it is designed to be a stand-alone window manager, it should not be used in conjunction with GNOME or KDE. To run this window manager, you need to install the **openmotif** package.

twm

The minimalist *Tab Window Manager* (**twm**, which provides the most basic tool set of any of the window managers, can be used either as a stand-alone or with a desktop environment. It is installed as part of the X11R7.1 release.

To run any of the aforementioned window managers, you will first need to boot into Runlevel 3. For instructions on how to do this, refer to *Section 6.1, "Runlevels"*.

Once you are logged in to Runlevel 3, you will be presented with a terminal prompt, not a graphical environment. To start a window manager, type **xinit -e *<path-to-window-manager>*** at the prompt.

<path-to-window-manager> is the location of the window manager binary file. The binary file can be located by typing **which *window-manager-name***, where ***window-manager-name*** is the name of the window manager you want to run.

For example:

```
user@host# which twm
          /usr/bin/twm
user@host# xinit -e /usr/bin/twm
```

The first command above returns the absolute path to the **twm** window manager, the second command starts **twm**.

To exit a window manager, close the last window or press **Ctrl**+**Alt**+**Backspace**. Once you have exited the window manager, you can log back into Runlevel 5 by typing **startx** at the prompt.

20.3. X Server Configuration Files

The X server is a single binary executable (**/usr/bin/Xorg**). Associated configuration files are stored in the **/etc/X11/** directory (as is a symbolic link — X — which points to **/usr/bin/Xorg**). The configuration file for the X server is **/etc/X11/xorg.conf**.

The directory **/usr/lib/xorg/modules/** contains X server modules that can be loaded dynamically at runtime. By default, only some modules in **/usr/lib/xorg/modules/** are automatically loaded by the X server.

To load optional modules, they must be specified in the X server configuration file, **/etc/X11/xorg.conf**. For more information about loading modules, refer to *Section 20.3.1.5, "**Module**"*.

When Fedora 12 is installed, the configuration files for X are created using information gathered about the system hardware during the installation process.

20.3.1. `xorg.conf`

While there is rarely a need to manually edit the **`/etc/X11/xorg.conf`** file, it is useful to understand the various sections and optional parameters available, especially when troubleshooting.

20.3.1.1. The Structure

The **`/etc/X11/xorg.conf`** file is comprised of many different sections which address specific aspects of the system hardware.

Each section begins with a **`Section "<section-name>"`** line (where <*section-name*> is the title for the section) and ends with an **`EndSection`** line. Each section contains lines that include option names and one or more option values. These are sometimes enclosed in double quotes (**`"`**).

Lines beginning with a hash mark (**`#`**) are not read by the X server and are used for human-readable comments.

Some options within the **`/etc/X11/xorg.conf`** file accept a boolean switch which turns the feature on or off. Acceptable boolean values are:

- **`1`**, **`on`**, **`true`**, or **`yes`** — Turns the option on.
- **`0`**, **`off`**, **`false`**, or **`no`** — Turns the option off.

The following are some of the more important sections in the order in which they appear in a typical **`/etc/X11/xorg.conf`** file. More detailed information about the X server configuration file can be found in the **`xorg.conf`** man page.

20.3.1.2. `ServerFlags`

The optional **`ServerFlags`** section contains miscellaneous global X server settings. Any settings in this section may be overridden by options placed in the **`ServerLayout`** section (refer to *Section 20.3.1.3, “`ServerLayout`”* for details).

Each entry within the **`ServerFlags`** section is on its own line and begins with the term **`Option`** followed by an option enclosed in double quotation marks (**`"`**).

The following is a sample **`ServerFlags`** section:

```
Section "ServerFlags" Option "DontZap" "true" EndSection
```

The following lists some of the most useful options:

- **`"DontZap" "<boolean>"`** — When the value of <*boolean*> is set to true, this setting prevents the use of the **`Ctrl`**+**`Alt`**+**`Backspace`** key combination to immediately terminate the X server.
- **`"DontZoom" "<boolean>"`** — When the value of <*boolean*> is set to true, this setting prevents cycling through configured video resolutions using the **`Ctrl`**+**`Alt`**+**`Keypad-Plus`** and **`Ctrl`**+**`Alt`**+**`Keypad-Minus`** key combinations.

20.3.1.3. ServerLayout

The **ServerLayout** section binds together the input and output devices controlled by the X server. At a minimum, this section must specify one output device and one input device. By default, a monitor (output device) and keyboard (input device) are specified.

The following example illustrates a typical **ServerLayout** section:

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

The following entries are commonly used in the **ServerLayout** section:

- **Identifier** — Specifies a unique name for this **ServerLayout** section.
- **Screen** — Specifies the name of a **Screen** section to be used with the X server. More than one **Screen** option may be present.

  The following is an example of a typical **Screen** entry:

  ```
  Screen 0 "Screen0" 0 0
  ```

  The first number in this example **Screen** entry (**0**) indicates that the first monitor connector or *head* on the video card uses the configuration specified in the **Screen** section with the identifier **"Screen0"**.

  An example of a **Screen** section with the identifier **"Screen0"** can be found in *Section 20.3.1.9, "**Screen**"*.

  If the video card has more than one head, another **Screen** entry with a different number and a different **Screen** section identifier is necessary .

  The numbers to the right of **"Screen0"** give the absolute X and Y coordinates for the upper-left corner of the screen (**0 0** by default).

- **InputDevice** — Specifies the name of an **InputDevice** section to be used with the X server.

  It is advisable that there be at least two **InputDevice** entries: one for the default mouse and one for the default keyboard. The options **CorePointer** and **CoreKeyboard** indicate that these are the primary mouse and keyboard.

- **Option "*<option-name>*"** — An optional entry which specifies extra parameters for the section. Any options listed here override those listed in the **ServerFlags** section.

  Replace *<option-name>* with a valid option listed for this section in the **xorg.conf** man page.

It is possible to put more than one **ServerLayout** section in the **/etc/X11/xorg.conf** file. By default, the server only reads the first one it encounters, however.

If there is an alternative **ServerLayout** section, it can be specified as a command line argument when starting an X session.

20.3.1.4. `Files`

The **`Files`** section sets paths for services vital to the X server, such as the font path. This is an optional section, these paths are normally detected automatically. This section may be used to override any automatically detected defaults.

The following example illustrates a typical **`Files`** section:

```
Section "Files" RgbPath "/usr/share/X11/rgb.txt" FontPath "unix/:7100"
 EndSection
```

The following entries are commonly used in the **`Files`** section:

- **`RgbPath`** — Specifies the location of the RGB color database. This database defines all valid color names in X and ties them to specific RGB values.
- **`FontPath`** — Specifies where the X server must connect to obtain fonts from the **`xfs`** font server.

  By default, the **`FontPath`** is **`unix/:7100`**. This tells the X server to obtain font information using UNIX-domain sockets for inter-process communication (IPC) on port 7100.

  Refer to *Section 20.4, “Fonts”* for more information concerning X and fonts.
- **`ModulePath`** — An optional parameter which specifies alternate directories which store X server modules.

20.3.1.5. `Module`

By default, the X server automatically loads the following modules from the **`/usr/lib/xorg/modules/`** directory:

- **`extmod`**
- **`dbe`**
- **`glx`**
- **`freetype`**
- **`type1`**
- **`record`**
- **`dri`**

The default directory for loading these modules can be changed by specifying a different directory with the optional **`ModulePath`** parameter in the **`Files`** section. Refer to *Section 20.3.1.4, “**`Files`**”* for more information on this section.

Adding a **`Module`** section to **`/etc/X11/xorg.conf`** instructs the X server to load the modules listed in this section *instead* of the default modules.

For example, the following typical **`Module`** section:

```
```

```
Section "Module" Load "fbdevhw" EndSection
```

instructs the X server to load the **fbdevhw** instead of the default modules.

As such, if you add a **Module** section to **/etc/X11/xorg.conf**, you will need to specify any default modules you want to load as well as any extra modules.

20.3.1.6. InputDevice

Each **InputDevice** section configures one input device for the X server. Systems typically have at least one **InputDevice** section for the keyboard. It is perfectly normal to have no entry for a mouse, as most mouse settings are automatically detected.

The following example illustrates a typical **InputDevice** section for a keyboard:

```

Section "InputDevice" Identifier "Keyboard0" Driver "kbd" Option "XkbModel"
 "pc105" Option "XkbLayout" "us" EndSection
```

The following entries are commonly used in the **InputDevice** section:

- **Identifier** — Specifies a unique name for this **InputDevice** section. This is a required entry.
- **Driver** — Specifies the name of the device driver X must load for the device.
- **Option** — Specifies necessary options pertaining to the device.

  A mouse may also be specified to override any autodetected defaults for the device. The following options are typically included when adding a mouse in the **xorg.conf**:

  - **Protocol** — Specifies the protocol used by the mouse, such as **IMPS/2**.
  - **Device** — Specifies the location of the physical device.
  - **Emulate3Buttons** — Specifies whether to allow a two-button mouse to act like a three-button mouse when both mouse buttons are pressed simultaneously.

  Consult the **xorg.conf** man page for a list of valid options for this section.

20.3.1.7. Monitor

Each **Monitor** section configures one type of monitor used by the system. This is an optional entry as well, as most monitors are now automatically detected.

The easiest way to configure a monitor is to configure X during the installation process or by using the **X Configuration Tool**. For more information about using the **X Configuration Tool**, refer to *Chapter 21, X Window System Configuration*.

This example illustrates a typical **Monitor** section for a monitor:

```

Section "Monitor" Identifier "Monitor0" VendorName "Monitor Vendor"
 ModelName "DDC Probed Monitor - ViewSonic G773-2" DisplaySize 320 240
 HorizSync 30.0 - 70.0 VertRefresh 50.0 - 180.0 EndSection
```

Warning

Be careful when manually editing values in the **`Monitor`** section of **`/etc/X11/xorg.conf`**. Inappropriate values can damage or destroy a monitor. Consult the monitor's documentation for a listing of safe operating parameters.

The following are commonly entries used in the **`Monitor`** section:

- **`Identifier`** — Specifies a unique name for this **`Monitor`** section. This is a required entry.
- **`VendorName`** — An optional parameter which specifies the vendor of the monitor.
- **`ModelName`** — An optional parameter which specifies the monitor's model name.
- **`DisplaySize`** — An optional parameter which specifies, in millimeters, the physical size of the monitor's picture area.
- **`HorizSync`** — Specifies the range of horizontal sync frequencies compatible with the monitor in kHz. These values help the X server determine the validity of built-in or specified **`Modeline`** entries for the monitor.
- **`VertRefresh`** — Specifies the range of vertical refresh frequencies supported by the monitor, in kHz. These values help the X server determine the validity of built in or specified **`Modeline`** entries for the monitor.
- **`Modeline`** — An optional parameter which specifies additional video modes for the monitor at particular resolutions, with certain horizontal sync and vertical refresh resolutions. Refer to the **`xorg.conf`** man page for a more detailed explanation of **`Modeline`** entries.
- **`Option "<option-name>"`** — An optional entry which specifies extra parameters for the section. Replace *`<option-name>`* with a valid option listed for this section in the **`xorg.conf`** man page.

20.3.1.8. Device

Each **`Device`** section configures one video card on the system. While one **`Device`** section is the minimum, additional instances may occur for each video card installed on the machine.

The best way to configure a video card is to configure X during the installation process or by using the **X Configuration Tool**. For more about using the **X Configuration Tool**, refer to *Chapter 21, X Window System Configuration*.

The following example illustrates a typical **`Device`** section for a video card:

```
Section "Device" Identifier "Videocard0" Driver "mga" VendorName "Videocard
 vendor" BoardName "Matrox Millennium G200" VideoRam 8192 Option "dpms"
 EndSection
```

The following entries are commonly used in the **`Device`** section:

- **`Identifier`** — Specifies a unique name for this **`Device`** section. This is a required entry.

- **Driver** — Specifies which driver the X server must load to utilize the video card. A list of drivers can be found in **/usr/share/hwdata/videodrivers**, which is installed with the **hwdata** package.

- **VendorName** — An optional parameter which specifies the vendor of the video card.

- **BoardName** — An optional parameter which specifies the name of the video card.

- **VideoRam** — An optional parameter which specifies the amount of RAM available on the video card in kilobytes. This setting is only necessary for video cards the X server cannot probe to detect the amount of video RAM.

- **BusID** — An entry which specifies the bus location of the video card. On systems with only one video card a **BusID** entry is optional and may not even be present in the default **/etc/X11/xorg.conf** file. On systems with more than one video card, however, a **BusID** entry must be present.

- **Screen** — An optional entry which specifies which monitor connector or head on the video card the **Device** section configures. This option is only useful for video cards with multiple heads.

  If multiple monitors are connected to different heads on the same video card, separate **Device** sections must exist and each of these sections must have a different **Screen** value.

  Values for the **Screen** entry must be an integer. The first head on the video card has a value of **0**. The value for each additional head increments this value by one.

- **Option "*<option-name>*"** — An optional entry which specifies extra parameters for the section. Replace *<option-name>* with a valid option listed for this section in the **xorg.conf** man page.

  One of the more common options is **"dpms"** (for Display Power Management Signaling, a VESA standard), which activates the Service Star energy compliance setting for the monitor.

20.3.1.9. Screen

Each **Screen** section binds one video card (or video card head) to one monitor by referencing the **Device** section and the **Monitor** section for each. While one **Screen** section is the minimum, additional instances may occur for each video card and monitor combination present on the machine.

The following example illustrates a typical **Screen** section:

```
Section "Screen" Identifier "Screen0" Device "Videocard0" Monitor
 "Monitor0" DefaultDepth 16 SubSection "Display" Depth 24 Modes "1280x1024"
 "1280x960" "1152x864" "1024x768" "800x600" "640x480" EndSubSection
 SubSection "Display" Depth 16 Modes "1152x864" "1024x768" "800x600"
 "640x480" EndSubSection EndSection
```

The following entries are commonly used in the **Screen** section:

- **Identifier** — Specifies a unique name for this **Screen** section. This is a required entry.

- **Device** — Specifies the unique name of a **Device** section. This is a required entry.

- **`Monitor`** — Specifies the unique name of a **`Monitor`** section. This is only required if a specific **`Monitor`** section is defined in the **`xorg.conf`** file. Normally, monitors are automatically detected.
- **`DefaultDepth`** — Specifies the default color depth in bits. In the previous example, **`16`** (which provides thousands of colors) is the default. Only one **`DefaultDepth`** is permitted, although this can be overridden with the Xorg command line option **`-depth`** ***`<n>`***,where ***`<n>`*** is any additional depth specified.
- **`SubSection "Display"`** — Specifies the screen modes available at a particular color depth. The **`Screen`** section can have multiple **`Display`** subsections, which are entirely optional since screen modes are automatically detected.

  This subsection is normally used to override autodetected modes.
- **`Option "`*`<option-name>`*`"`** — An optional entry which specifies extra parameters for the section. Replace *`<option-name>`* with a valid option listed for this section in the **`xorg.conf`** man page.

20.3.1.10. DRI

The optional **`DRI`** section specifies parameters for the *Direct Rendering Infrastructure* (*DRI*). DRI is an interface which allows 3D software applications to take advantage of 3D hardware acceleration capabilities built into most modern video hardware. In addition, DRI can improve 2D performance via hardware acceleration, if supported by the video card driver.

This section rarely appears, as the DRI Group and Mode are automatically initialized to default values. If a different Group or Mode is desired, then adding this section to the **`xorg.conf`** file will override those defaults.

The following example illustrates a typical **`DRI`** section:

```
Section "DRI" Group 0 Mode 0666 EndSection
```

Since different video cards use DRI in different ways, do not add to this section without first referring to *http://dri.sourceforge.net/*.

20.4. Fonts

Fedora uses two subsystems to manage and display fonts under X: *Fontconfig* and **`xfs`**.

The newer Fontconfig font subsystem simplifies font management and provides advanced display features, such as anti-aliasing. This system is used automatically for applications programmed using the Qt 3 or GTK+ 2 graphical toolkit.

For compatibility, Fedora includes the original font subsystem, called the core X font subsystem. This system, which is over 15 years old, is based around the *X Font Server* (*xfs*).

This section discusses how to configure fonts for X using both systems.

20.4.1. Fontconfig

The Fontconfig font subsystem allows applications to directly access fonts on the system and use Xft or other rendering mechanisms to render Fontconfig fonts with advanced anti-aliasing. Graphical applications can use the Xft library with Fontconfig to draw text to the screen.

Over time, the Fontconfig/Xft font subsystem replaces the core X font subsystem.

Important

The Fontconfig font subsystem does not yet work for **OpenOffice.org**, which uses its own font rendering technology.

It is important to note that Fontconfig uses the `/etc/fonts/fonts.conf` configuration file, which should not be edited by hand.

Tip

Due to the transition to the new font system, GTK+ 1.2 applications are not affected by any changes made via the **Font Preferences** dialog (accessed by selecting System (on the panel) > **Preferences** > **Fonts**). For these applications, a font can be configured by adding the following lines to the file `~/.gtkrc.mine`:

```
style "user-font" { fontset = "<font-specification>" }
 widget_class "*" style "user-font"
```

Replace *<font-specification>* with a font specification in the style used by traditional X applications, such as `-adobe-helvetica-medium-r-normal--*-120-*-*-*-*-*-*`. A full list of core fonts can be obtained by running `xlsfonts` or created interactively using the `xfontsel` command.

20.4.1.1. Adding Fonts to Fontconfig

Adding new fonts to the Fontconfig subsystem is a straightforward process.

1. To add fonts system-wide, copy the new fonts into the `/usr/share/fonts/` directory. It is a good idea to create a new subdirectory, such as `local/` or similar, to help distinguish between user-installed and default fonts.

   To add fonts for an individual user, copy the new fonts into the `.fonts/` directory in the user's home directory.

2. Use the **fc-cache** command to update the font information cache, as in the following example:

   ```
   fc-cache <path-to-font-directory>
   ```

   In this command, replace *<path-to-font-directory>* with the directory containing the new fonts (either `/usr/share/fonts/local/` or `/home/<user>/.fonts/`).

Tip

Individual users may also install fonts graphically, by typing `fonts:///` into the **Nautilus** address bar, and dragging the new font files there.

Important

If the font file name ends with a **.gz** extension, it is compressed and cannot be used until uncompressed. To do this, use the **gunzip** command or double-click the file and drag the font to a directory in **Nautilus**.

20.4.2. Core X Font System

For compatibility, Fedora provides the core X font subsystem, which uses the X Font Server (**xfs**) to provide fonts to X client applications.

The X server looks for a font server specified in the **FontPath** directive within the **Files** section of the **/etc/X11/xorg.conf** configuration file. Refer to *Section 20.3.1.4, “**Files**”* for more information about the **FontPath** entry.

The X server connects to the **xfs** server on a specified port to acquire font information. For this reason, the **xfs** service must be running for X to start. For more about configuring services for a particular runlevel, refer to *Chapter 6, Controlling Access to Services*.

20.4.2.1. xfs Configuration

The **/etc/rc.d/init.d/xfs** script starts the **xfs** server. Several options can be configured within its configuration file, **/etc/X11/fs/config**.

The following lists common options:

- **alternate-servers** — Specifies a list of alternate font servers to be used if this font server is not available. A comma must separate each font server in a list.
- **catalogue** — Specifies an ordered list of font paths to use. A comma must separate each font path in a list.

  Use the string **:unscaled** immediately after the font path to make the unscaled fonts in that path load first. Then specify the entire path again, so that other scaled fonts are also loaded.
- **client-limit** — Specifies the maximum number of clients the font server services. The default is **10**.
- **clone-self** — Allows the font server to clone a new version of itself when the **client-limit** is hit. By default, this option is **on**.
- **default-point-size** — Specifies the default point size for any font that does not specify this value. The value for this option is set in decipoints. The default of **120** corresponds to a 12 point font.
- **default-resolutions** — Specifies a list of resolutions supported by the X server. Each resolution in the list must be separated by a comma.
- **deferglyphs** — Specifies whether to defer loading *glyphs* (the graphic used to visually represent a font). To disable this feature use **none**, to enable this feature for all fonts use **all**, or to turn this feature on only for 16-bit fonts use **16**.
- **error-file** — Specifies the path and file name of a location where **xfs** errors are logged.

- **`no-listen`** — Prevents **`xfs`** from listening to particular protocols. By default, this option is set to **`tcp`** to prevent **`xfs`** from listening on TCP ports for security reasons.

  Tip

  If **`xfs`** is used to serve fonts over the network, remove this line.

- **`port`** — Specifies the TCP port that **`xfs`** listens on if **`no-listen`** does not exist or is commented out.
- **`use-syslog`** — Specifies whether to use the system error log.

20.4.2.2. Adding Fonts to `xfs`

To add fonts to the core X font subsystem (**`xfs`**), follow these steps:

1. If it does not already exist, create a directory called **`/usr/share/fonts/local/`** using the following command as root:

   ```
   mkdir /usr/share/fonts/local/
   ```

   If creating the **`/usr/share/fonts/local/`** directory is necessary, it must be added to the **`xfs`** path using the following command as root:

   ```
   chkfontpath --add /usr/share/fonts/local/
   ```

2. Copy the new font file into the **`/usr/share/fonts/local/`** directory
3. Update the font information by issuing the following command as root:

   ```
   ttmkfdir -d /usr/share/fonts/local/ -o /usr/share/fonts/local/
   fonts.scale
   ```

4. Reload the **`xfs`** font server configuration file by issuing the following command as root:

   ```
   service xfs reload
   ```

20.5. Runlevels and X

In most cases, the Fedora installer configures a machine to boot into a graphical login environment, known as *Runlevel 5*. It is possible, however, to boot into a text-only multi-user mode called *Runlevel 3* and begin an X session from there.

For more information about runlevels, refer to *Section 6.1, “Runlevels”*.

The following subsections review how X starts up in both runlevel 3 and runlevel 5.

20.5.1. Runlevel 3

When in runlevel 3, the best way to start an X session is to log in and type **`startx`**. The **`startx`** command is a front-end to the **`xinit`** command, which launches the X server (**`Xorg`**) and connects X client applications to it. Because the user is already logged into the system at runlevel 3, **`startx`** does not launch a display manager or authenticate users. Refer to *Section 20.5.2, “Runlevel 5”* for more information about display managers.

When the **`startx`** command is executed, it searches for the **`.xinitrc`** file in the user's home directory to define the desktop environment and possibly other X client applications to run. If no **`.xinitrc`** file is present, it uses the system default **`/etc/X11/xinit/xinitrc`** file instead.

The default **`xinitrc`** script then searches for user-defined files and default system files, including **`.Xresources`**, **`.Xmodmap`**, and **`.Xkbmap`** in the user's home directory, and **`Xresources`**, **`Xmodmap`**, and **`Xkbmap`** in the **`/etc/X11/`** directory. The **`Xmodmap`** and **`Xkbmap`** files, if they exist, are used by the **`xmodmap`** utility to configure the keyboard. The **`Xresources`** file is read to assign specific preference values to applications.

After setting these options, the **`xinitrc`** script executes all scripts located in the **`/etc/X11/xinit/xinitrc.d/`** directory. One important script in this directory is **`xinput.sh`**, which configures settings such as the default language.

Next, the **`xinitrc`** script attempts to execute **`.Xclients`** in the user's home directory and turns to **`/etc/X11/xinit/Xclients`** if it cannot be found. The purpose of the **`Xclients`** file is to start the desktop environment or, possibly, just a basic window manager. The **`.Xclients`** script in the user's home directory starts the user-specified desktop environment in the **`.Xclients-default`** file. If **`.Xclients`** does not exist in the user's home directory, the standard **`/etc/X11/xinit/Xclients`** script attempts to start another desktop environment, trying GNOME first and then KDE followed by **`twm`**.

When in runlevel 3, the user is returned to a text mode user session after ending an X session.

20.5.2. Runlevel 5

When the system boots into runlevel 5, a special X client application called a *display manager* is launched. A user must authenticate using the display manager before any desktop environment or window managers are launched.

Depending on the desktop environments installed on the system, three different display managers are available to handle user authentication.

- **`GNOME`** — The default display manager for Fedora, **`GNOME`** allows the user to configure language settings, shutdown, restart or log in to the system.
- **`KDE`** — KDE's display manager which allows the user to shutdown, restart or log in to the system.
- **`xdm`** — A very basic display manager which only lets the user log in to the system.

When booting into runlevel 5, the **`prefdm`** script determines the preferred display manager by referencing the **`/etc/sysconfig/desktop`** file. A list of options for this file is available in this file:

```
/usr/share/doc/initscripts-<version-number>/sysconfig.txt
```

where *<version-number>* is the version number of the **initscripts** package.

Each of the display managers reference the **/etc/X11/xdm/Xsetup_0** file to set up the login screen. Once the user logs into the system, the **/etc/X11/xdm/GiveConsole** script runs to assign ownership of the console to the user. Then, the **/etc/X11/xdm/Xsession** script runs to accomplish many of the tasks normally performed by the **xinitrc** script when starting X from runlevel 3, including setting system and user resources, as well as running the scripts in the **/etc/X11/xinit/xinitrc.d/** directory.

Users can specify which desktop environment they want to utilize when they authenticate using the **GNOME** or **KDE** display managers by selecting it from the **Sessions** menu item (accessed by selecting System (on the panel) > **Preferences** > **More Preferences** > **Sessions**). If the desktop environment is not specified in the display manager, the **/etc/X11/xdm/Xsession** script checks the **.xsession** and **.Xclients** files in the user's home directory to decide which desktop environment to load. As a last resort, the **/etc/X11/xinit/Xclients** file is used to select a desktop environment or window manager to use in the same way as runlevel 3.

When the user finishes an X session on the default display (:0) and logs out, the **/etc/X11/xdm/TakeConsole** script runs and reassigns ownership of the console to the root user. The original display manager, which continues running after the user logged in, takes control by spawning a new display manager. This restarts the X server, displays a new login window, and starts the entire process over again.

The user is returned to the display manager after logging out of X from runlevel 5.

For more information on how display managers control user authentication, refer to the **/usr/share/doc/gdm-*<version-number>*/README** (where *<version-number>* is the version number for the **gdm** package installed) and the **xdm** man page.

20.6. Additional Resources

There is a large amount of detailed information available about the X server, the clients that connect to it, and the assorted desktop environments and window managers.

20.6.1. Installed Documentation

- **/usr/share/X11/doc/** — contains detailed documentation on the X Window System architecture, as well as how to get additional information about the Xorg project as a new user.
- **man xorg.conf** — Contains information about the **xorg.conf** configuration files, including the meaning and syntax for the different sections within the files.
- **man Xorg** — Describes the **Xorg** display server.

20.6.2. Useful Websites

- *http://www.X.org/* — Home page of the X.Org Foundation, which produces the X11R7.1 release of the X Window System. The X11R7.1 release is bundled with Fedora to control the necessary hardware and provide a GUI environment.
- *http://dri.sourceforge.net/* — Home page of the DRI (Direct Rendering Infrastructure) project. The DRI is the core hardware 3D acceleration component of X.

- *http://www.gnome.org/*[1] — Home of the GNOME project.
- *http://www.kde.org/*[2] — Home of the KDE desktop environment.

X Window System Configuration

During installation, the system's monitor, video card, and display settings are configured. To change any of these settings after installation, use the **X Configuration Tool**.

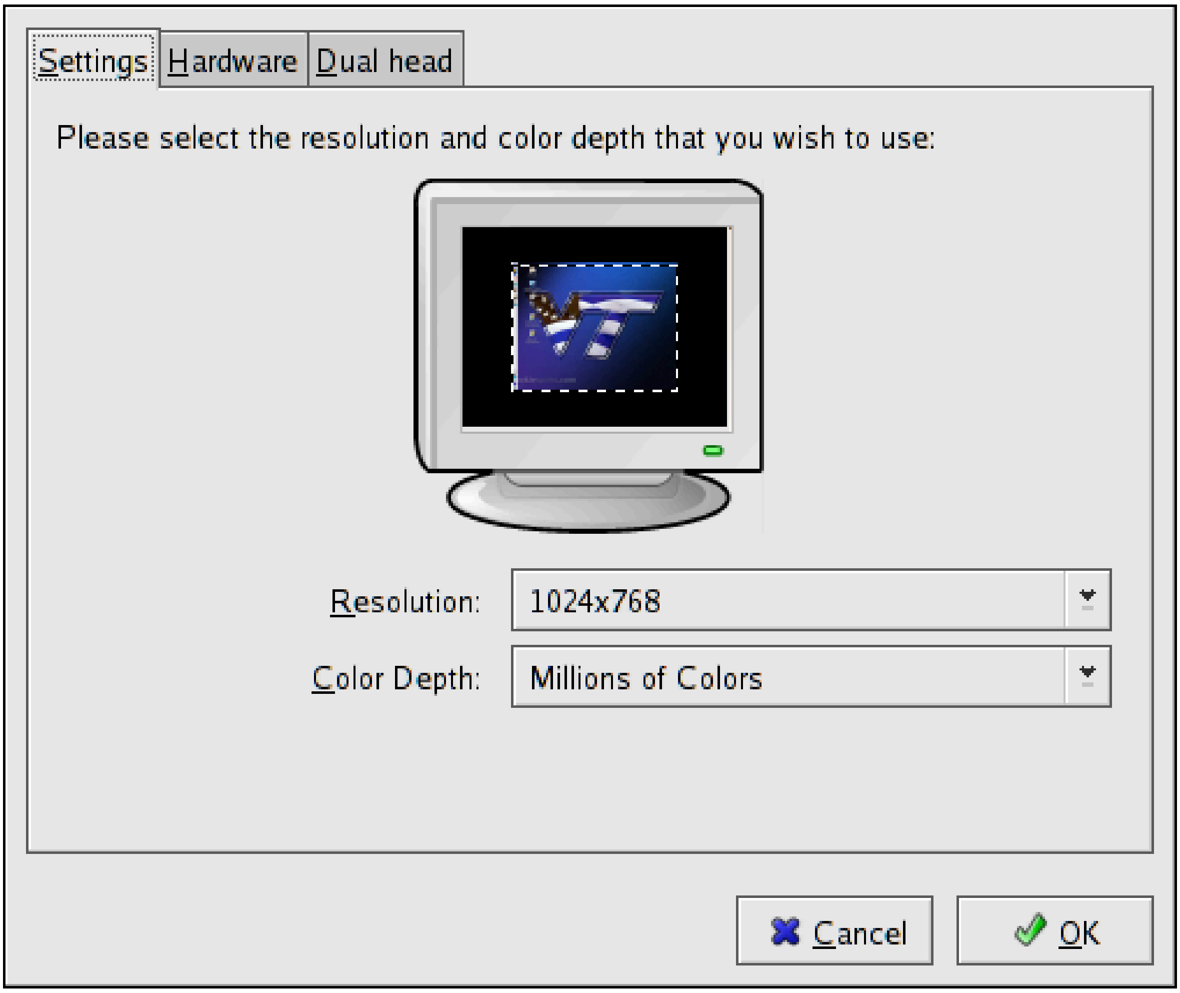
To start the **X Configuration Tool**, go to System (on the panel) > **Administration** > **Display**, or type the command **`system-config-display`** at a shell prompt (for example, in an XTerm or GNOME terminal). If the X Window System is not running, a small version of X is started to run the program.

After changing any of the settings, log out of the graphical desktop and log back in to enable the changes.

21.1. Display Settings

The **Settings** tab allows users to change the *resolution* and *color depth*. The display of a monitor consists of tiny dots called *pixels*. The number of pixels displayed at one time is called the resolution. For example, the resolution 1024x768 means that 1024 horizontal pixels and 768 vertical pixels are used. The higher the resolution values, the more images the monitor can display at one time.

The color depth of the display determines how many possible colors are displayed. A higher color depth means more contrast between colors.



Please select the resolution and color depth that you wish to use:

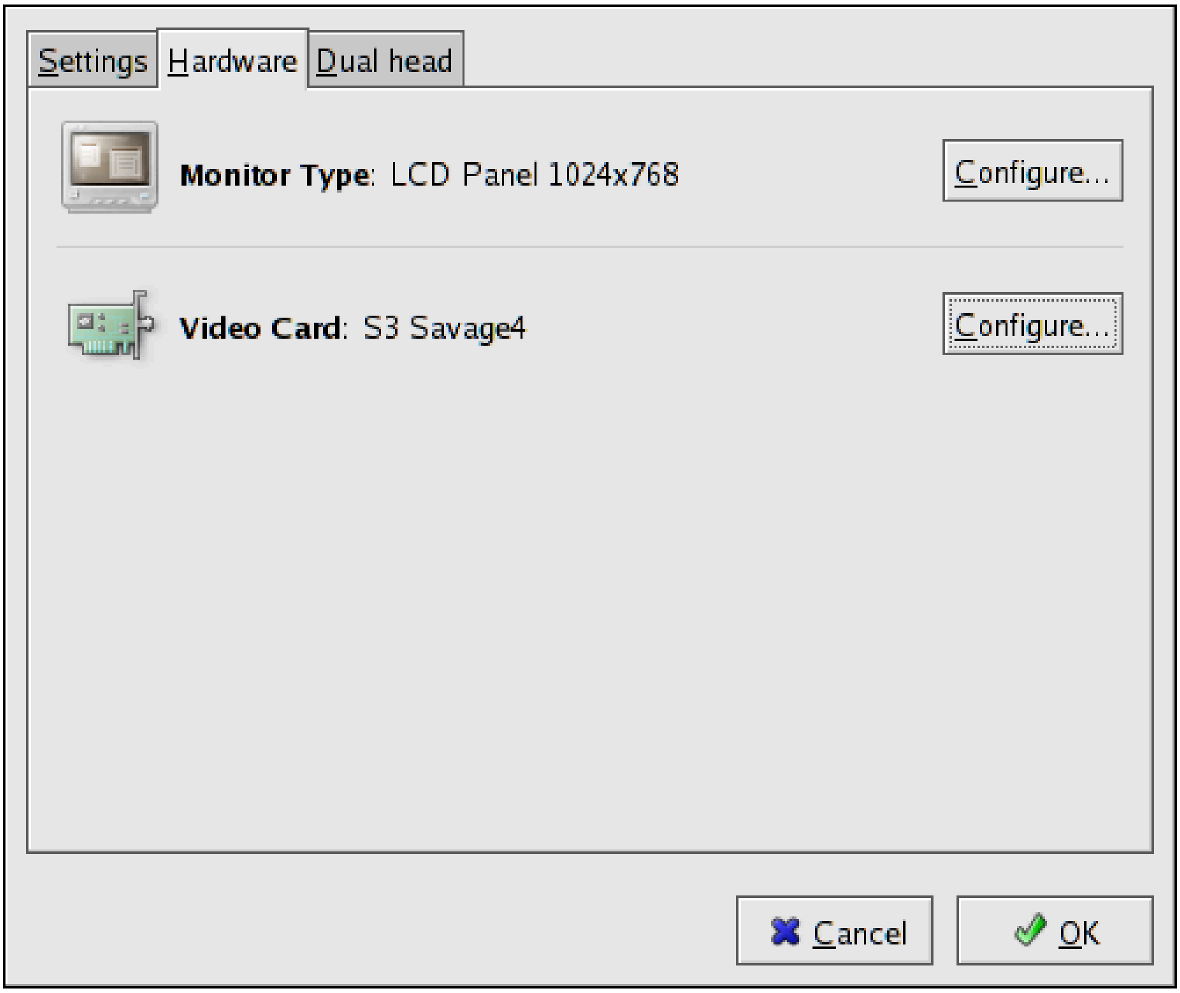
Resolution: 1024x768

Color Depth: Millions of Colors

Figure 21.1. Display Settings

21.2. Display Hardware Settings

When the **X Configuration Tool** is started, it probes the monitor and video card. If the hardware is probed properly, the information for it is shown on the **Hardware** tab as shown in *Figure 21.2, "Display Hardware Settings"*.



Monitor Type: LCD Panel 1024x768

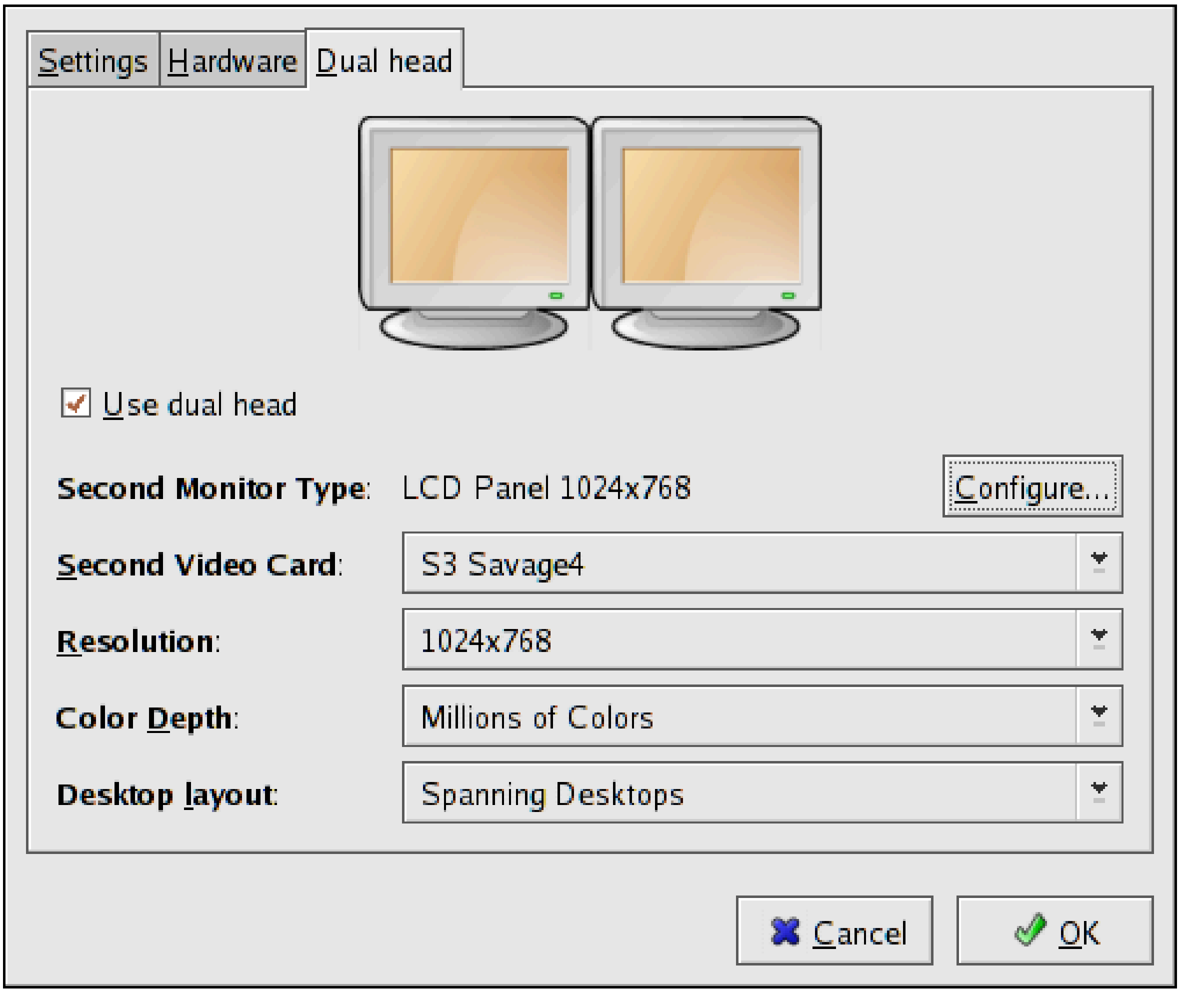
Video Card: S3 Savage4

Figure 21.2. Display Hardware Settings

To change the monitor type or any of its settings, click the corresponding **Configure** button. To change the video card type or any of its settings, click the **Configure** button beside its settings.

21.3. Dual Head Display Settings

If multiple video cards are installed on the system, dual head monitor support is available and is configured via the **Dual head** tab, as shown in *Figure 21.3, "Dual Head Display Settings"*.



☑ Use dual head

Second Monitor Type: LCD Panel 1024x768

Second Video Card: S3 Savage4

Resolution: 1024x768

Color Depth: Millions of Colors

Desktop layout: Spanning Desktops

Figure 21.3. Dual Head Display Settings

To enable use of Dual head, check the **Use dual head** checkbox.

To configure the second monitor type, click the corresponding **Configure** button. You can also configure the other Dual head settings by using the corresponding drop-down list.

For the **Desktop layout** option, selecting **Spanning Desktops** allows both monitors to use an enlarged usable workspace. Selecting **Individual Desktops** shares the mouse and keyboard among the displays, but restricts windows to a single display.

Users and Groups

The control of *users* and *groups* is a core element of Fedora system administration.

Users can be either people (meaning accounts tied to physical users) or accounts which exist for specific applications to use.

Groups are logical expressions of organization, tying users together for a common purpose. Users within a group can read, write, or execute files owned by that group.

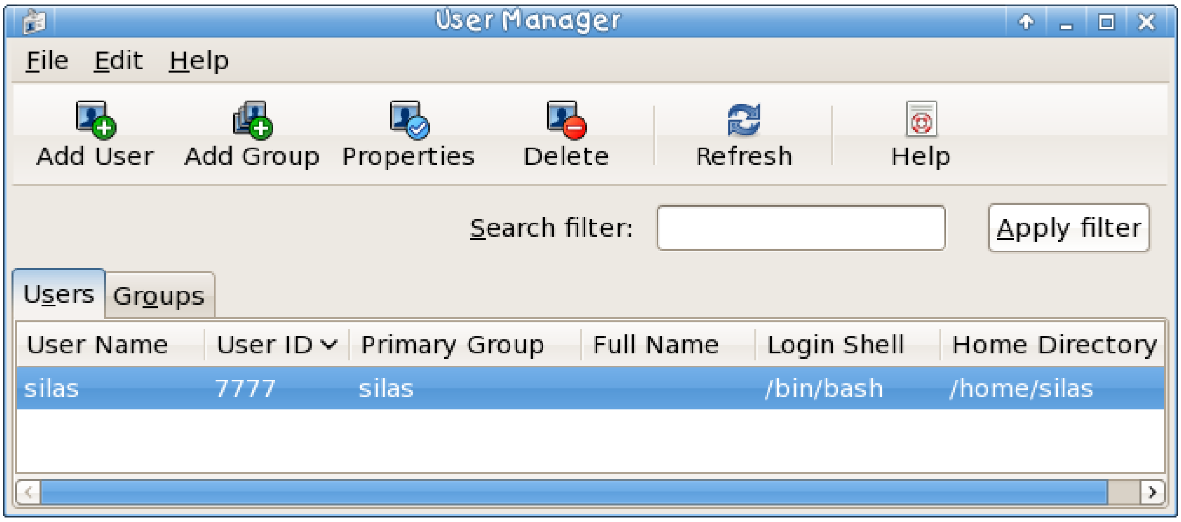
Each user is associated with a unique numerical identification number called a *userid* (*UID*); likewise, each group is associated with a *groupid* (*GID*).

A user who creates a file is also the owner and group owner of that file. The file is assigned separate read, write, and execute permissions for the owner, the group, and everyone else. The file owner can be changed only by the root user, and access permissions can be changed by both the root user and file owner.

Fedora also supports *access control lists* (*ACLs*) for files and directories which allow permissions for specific users outside of the owner to be set. For more information about ACLs, refer to

22.1. User and Group Configuration

The **User Manager** allows you to view, modify, add, and delete local users and groups.



User Name	User ID	Primary Group	Full Name	Login Shell	Home Directory
silas	7777	silas		/bin/bash	/home/silas

Figure 22.1. The GNOME **User Manager**

You can start the **User Manager** by clicking **System** → **Administration** → **Users and Groups**. Alternatively, you can enter **`system-config-users`** at the shell prompt to open the **User Manager**. Viewing and modifying user and group information requires superuser privileges. If you are not the superuser when you open the **User Manager**, it will prompt you for the superuser password.

To view a list of local users on the system, click the **Users** tab. To view a list of local groups on the system, click the **Groups** tab.

To find a specific user or group, type the first few letters of the name in the **Search filter** field. Press **`Enter`** or click the **Apply filter** button. The filtered list is displayed.

To sort the users or groups, click on the column name. The users or groups are sorted according to the value of that column.

Fedora reserves user IDs below 500 for system users. By default, the **User Manager** does not display system users. To view all users, including the system users, go to **Edit** > **Preferences** and uncheck **Hide system users and groups** from the dialog box.

22.1.1. Adding a New User

To add a new user, click the **Add User** button. A window as shown in *Figure 22.2, “Creating a new user”* appears. Type the username and full name for the new user in the appropriate fields. Type the user's password in the **Password** and **Confirm Password** fields. The password must be at least six characters.

Tip

It is advisable to use a much longer password, as this makes it more difficult for an intruder to guess it and access the account without permission. It is also recommended that the password not be based on a dictionary term; use a combination of letters, numbers and special characters.

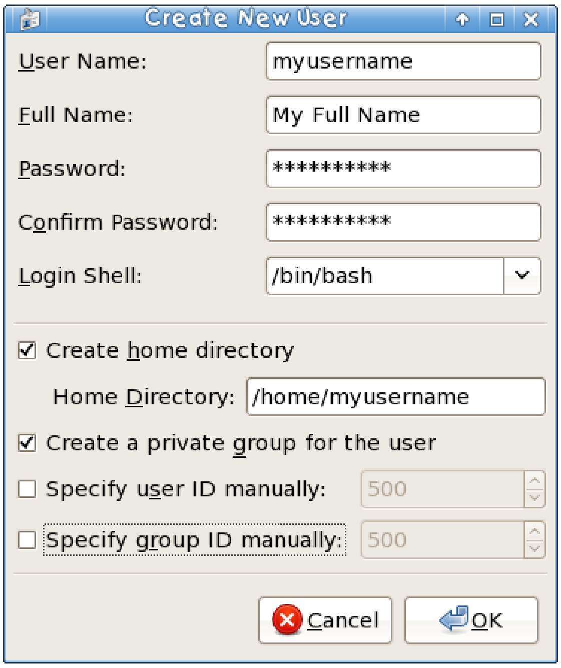
Select a login shell. If you are not sure which shell to select, accept the default value of `/bin/bash`. The default home directory is ***`/home/<username>/`***. You can change the home directory that is created for the user, or you can choose not to create the home directory by unselecting **Create home directory**.

If you select to create the home directory, default configuration files are copied from the **`/etc/skel/`** directory into the new home directory.

Fedora uses a *user private group* (UPG) scheme. The UPG scheme does not add or change anything in the standard UNIX way of handling groups; it offers a new convention. Whenever you create a new user, by default, a unique group with the same name as the user is created. If you do not want to create this group, unselect **Create a private group for the user**.

To specify a user ID for the user, select **Specify user ID manually**. If the option is not selected, the next available user ID above 500 is assigned to the new user. Because Fedora reserves user IDs below 500 for system users, it is not advisable to manually assign user IDs 1-499.

Click **OK** to create the user.



User Name: myusername

Full Name: My Full Name

Password: **********

Confirm Password: **********

Login Shell: /bin/bash

☑ Create home directory

Home Directory: /home/myusername

☑ Create a private group for the user

□ Specify user ID manually: 500

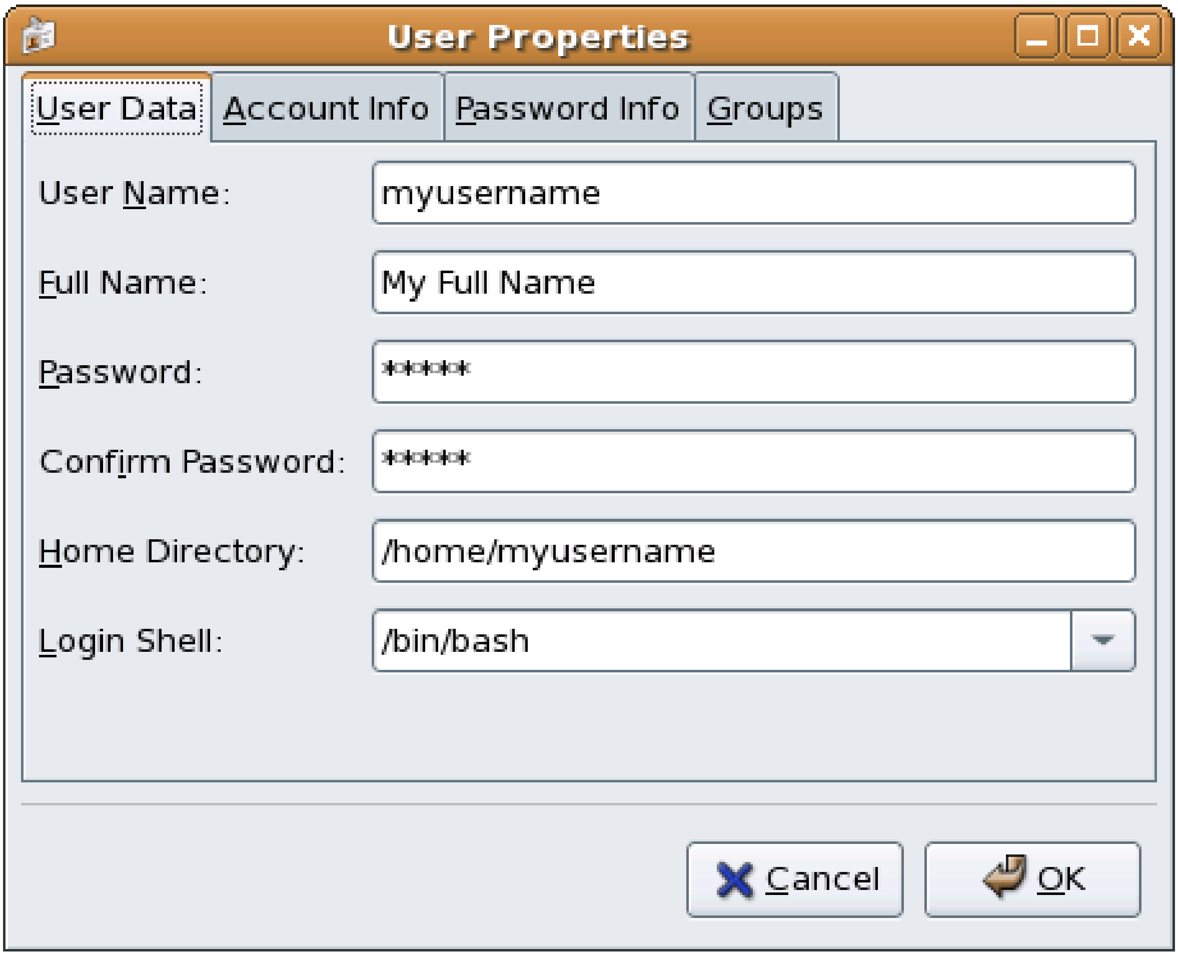
□ Specify group ID manually: 500

Figure 22.2. Creating a new user

To configure more advanced user properties, such as password expiration, modify the user's properties after adding the user. Refer to *Section 22.1.2, “Modifying User Properties”* for more information.

22.1.2. Modifying User Properties

To view the properties of an existing user, click on the **Users** tab, select the user from the user list, and click **Properties** from the menu (or choose **File** > **Properties** from the pulldown menu). A window similar to *Figure 22.3, “User Properties”* appears.



User Name: myusername

Full Name: My Full Name

Password: ******

Confirm Password: ******

Home Directory: /home/myusername

Login Shell: /bin/bash

Figure 22.3. User Properties

The **User Properties** window is divided into multiple tabbed pages:

- **User Data** — Shows the basic user information configured when you added the user. Use this tab to change the user's full name, password, home directory, or login shell.

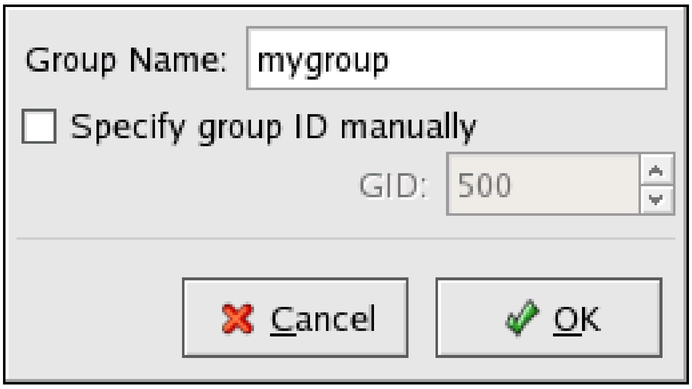
- **Account Info** —
  Select **Enable account expiration** if you want the account to expire on a certain date. Enter the date in the provided fields.
  Select **Local password is locked** to lock the user account and prevent the user from logging into the system.

- **Password Info** — Displays the date that the user's password last changed. To force the user to change passwords after a certain number of days, select **Enable password expiration** and enter a desired value in the **Days before change required:** field. The number of days before the user's password expires, the number of days before the user is warned to change passwords, and days before the account becomes inactive can also be changed.

- **Groups** — Allows you to view and configure the Primary Group of the user, as well as other groups that you want the user to be a member of.

22.1.3. Adding a New Group

To add a new user group, click the **Add Group** button. A window similar to *Figure 22.4, "New Group"* appears. Type the name of the new group to create. To specify a group ID for the new group, select **Specify group ID manually** and select the GID. Note that Fedora also reserves group IDs lower than 500 for system groups.



Group Name: mygroup

□ Specify group ID manually

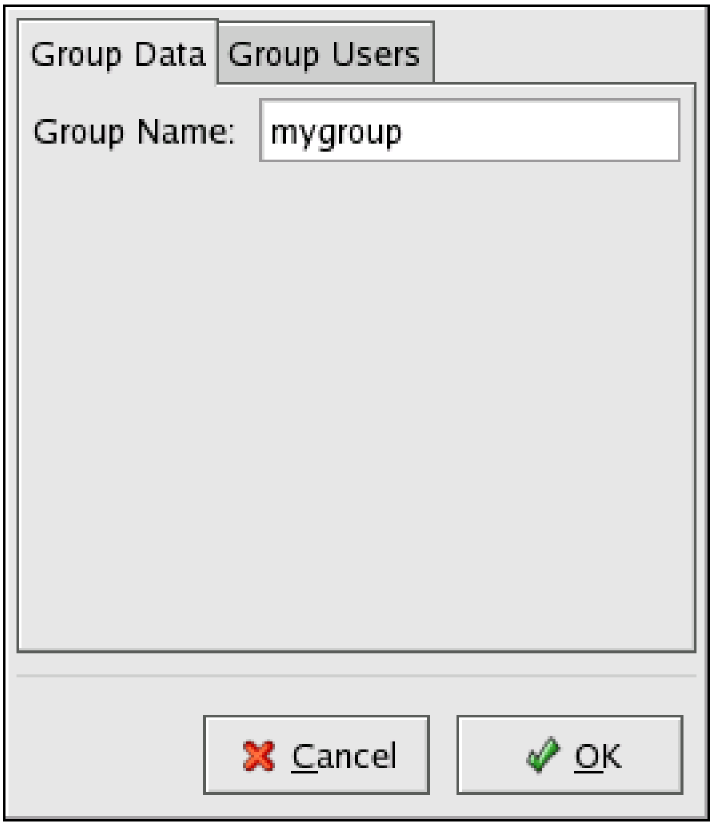
GID: 500

Figure 22.4. New Group

Click **OK** to create the group. The new group appears in the group list.

22.1.4. Modifying Group Properties

To view the properties of an existing group, select the group from the group list and click **Properties** from the menu (or choose **File** > **Properties** from the pulldown menu). A window similar to *Figure 22.5, "Group Properties"* appears.



Group Name: mygroup

Figure 22.5. Group Properties

The **Group Users** tab displays which users are members of the group. Use this tab to add or remove users from the group. Click **OK** to save your changes.

22.2. User and Group Management Tools

Managing users and groups can be a tedious task; this is why Fedora provides tools and conventions to make them easier to manage.

The easiest way to manage users and groups is through the graphical application, **User Manager** (`system-config-users`). For more information on **User Manager**, refer to *Section 22.1, “User and Group Configuration”*.

The following command line tools can also be used to manage users and groups:

- `useradd`, `usermod`, and `userdel` — Industry-standard methods of adding, deleting and modifying user accounts
- `groupadd`, `groupmod`, and `groupdel` — Industry-standard methods of adding, deleting, and modifying user groups
- `gpasswd` — Industry-standard method of administering the `/etc/group` file
- `pwck`, `grpck` — Tools used for the verification of the password, group, and associated shadow files
- `pwconv`, `pwunconv` — Tools used for the conversion of passwords to shadow passwords and back to standard passwords

22.2.1. Command Line Configuration

If you prefer command line tools or do not have the X Window System installed, use this section to configure users and groups.

22.2.2. Adding a User

To add a user to the system:

1. Issue the **`useradd`** command to create a locked user account:

   ```
   useradd <username>
   ```

2. Unlock the account by issuing the **`passwd`** command to assign a password and set password aging guidelines:

   ```
   passwd <username>
   ```

Command line options for **`useradd`** are detailed in *Table 22.1, “**`useradd`** Command Line Options”*.

Option	Description
`-c` '*`<comment>`*'	*`<comment>`* can be replaced with any string. This option is generally used to specify the full name of a user.
`-d`*`<home-dir>`*	Home directory to be used instead of default **`/home/<username>/`**
`-e`*`<date>`*	Date for the account to be disabled in the format YYYY-MM-DD
`-f`*`<days>`*	Number of days after the password expires until the account is disabled. If **`0`** is specified, the account is disabled immediately after the password expires. If **`-1`** is specified, the account is not be disabled after the password expires.
`-g`*`<group-name>`*	Group name or group number for the user's default group. The group must exist prior to being specified here.
`-G`*`<group-list>`*	List of additional (other than default) group names or group numbers, separated by commas, of which the user is a member. The groups must exist prior to being specified here.
`-m`	Create the home directory if it does not exist.
`-M`	Do not create the home directory.
`-n`	Do not create a user private group for the user.
`-r`	Create a system account with a UID less than 500 and without a home directory
`-p`*`<password>`*	The password encrypted with **`crypt`**
`-s`	User's login shell, which defaults to **`/bin/bash`**
`-u`*`<uid>`*	User ID for the user, which must be unique and greater than 499

Table 22.1. **`useradd`** Command Line Options

22.2.3. Adding a Group

To add a group to the system, use the command **`groupadd`**:

```
groupadd <group-name>
```

Command line options for **`groupadd`** are detailed in *Table 22.2, “**`groupadd`** Command Line Options”*.

Option	Description
`-g<gid>`	Group ID for the group, which must be unique and greater than 499
`-r`	Create a system group with a GID less than 500
`-f`	When used with `-g<gid>` and `<gid>` already exists, **`groupadd`** will choose another unique `<gid>` for the group.

Table 22.2. **`groupadd`** Command Line Options

22.2.4. Password Aging

For security reasons, it is advisable to require users to change their passwords periodically. This can be done when adding or editing a user on the **Password Info** tab of the **User Manager**.

To configure password expiration for a user from a shell prompt, use the **`chage`** command, followed by an option from *Table 22.3, “**`chage`** Command Line Options”*, followed by the username of the user.

Important

Shadow passwords must be enabled to use the **`chage`** command.

Option	Description
`-m<days>`	Specifies the minimum number of days between which the user must change passwords. If the value is 0, the password does not expire.
`-M<days>`	Specifies the maximum number of days for which the password is valid. When the number of days specified by this option plus the number of days specified with the `-d` option is less than the current day, the user must change passwords before using the account.
`-d<days>`	Specifies the number of days since January 1, 1970 the password was changed
`-I<days>`	Specifies the number of inactive days after the password expiration before locking the account. If the value is 0, the account is not locked after the password expires.
`-E<date>`	Specifies the date on which the account is locked, in the format YYYY-MM-DD. Instead of the date, the number of days since January 1, 1970 can also be used.
`-W<days>`	Specifies the number of days before the password expiration date to warn the user.

Table 22.3. **`chage`** Command Line Options

Tip

If the **chage** command is followed directly by a username (with no options), it displays the current password aging values and allows them to be changed.

You can configure a password to expire the first time a user logs in. This forces users to change passwords the first time they log in.

Note

This process will not work if the user logs in using the SSH protocol.

1. *Lock the user password* — If the user does not exist, use the **useradd** command to create the user account, but do not give it a password so that it remains locked.

   If the password is already enabled, lock it with the command:

   ```
   usermod -L username
   ```

2. *Force immediate password expiration* — Type the following command:

   ```
   chage -d 0 username
   ```

   This command sets the value for the date the password was last changed to the epoch (January 1, 1970). This value forces immediate password expiration no matter what password aging policy, if any, is in place.

3. *Unlock the account* — There are two common approaches to this step. The administrator can assign an initial password or assign a null password.

   ### Warning

   Do not use the **passwd** command to set the password as it disables the immediate password expiration just configured.

   To assign an initial password, use the following steps:

   - Start the command line Python interpreter with the **python** command. It displays the following:

     ```
     Python 2.4.3 (#1, Jul 21 2006, 08:46:09) [GCC 4.1.1 20060718 (Red Hat
     4.1.1-9)] on linux2 Type "help", "copyright", "credits" or "license"
     for more information. >>>
     ```

   - At the prompt, type the following commands. Replace <*password*> with the password to encrypt and <*salt*> with a random combination of at least 2 of the following: any alphanumeric character, the slash (/) character or a dot (.):

```
import crypt; print crypt.crypt("<password>","<salt>")
```

The output is the encrypted password, similar to `'12CsGd8FRcMSM'`.

- Press **`Ctrl-D`** to exit the Python interpreter.
- At the shell, enter the following command (replacing *`<encrypted-password>`* with the encrypted output of the Python interpreter):

```
usermod -p "<encrypted-password>" <username>
```

Alternatively, you can assign a null password instead of an initial password. To do this, use the following command:

```
usermod -p "" username
```

Caution

Using a null password, while convenient, is a highly unsecure practice, as any third party can log in first an access the system using the unsecure username. Always make sure that the user is ready to log in before unlocking an account with a null password.

In either case, upon initial log in, the user is prompted for a new password.

22.2.5. Explaining the Process

The following steps illustrate what happens if the command **`useradd juan`** is issued on a system that has shadow passwords enabled:

1. A new line for `juan` is created in **`/etc/passwd`**. The line has the following characteristics:
   - It begins with the username `juan`.
   - There is an `x` for the password field indicating that the system is using shadow passwords.
   - A UID greater than 499 is created. (Under Fedora, UIDs and GIDs below 500 are reserved for system use.)
   - A GID greater than 499 is created.
   - The optional GECOS information is left blank.
   - The home directory for `juan` is set to **`/home/juan/`**.
   - The default shell is set to **`/bin/bash`**.
2. A new line for `juan` is created in **`/etc/shadow`**. The line has the following characteristics:
   - It begins with the username `juan`.

- Two exclamation points (`!!`) appear in the password field of the **`/etc/shadow`** file, which locks the account.

> **Note**
>
> If an encrypted password is passed using the `-p` flag, it is placed in the **`/etc/shadow`** file on the new line for the user.

- The password is set to never expire.

3. A new line for a group named `juan` is created in **`/etc/group`**. A group with the same name as a user is called a *user private group*. For more information on user private groups, refer to *Section 22.1.1, “Adding a New User”*.

   The line created in **`/etc/group`** has the following characteristics:

   - It begins with the group name `juan`.
   - An `x` appears in the password field indicating that the system is using shadow group passwords.
   - The GID matches the one listed for user `juan` in **`/etc/passwd`**.

4. A new line for a group named `juan` is created in **`/etc/gshadow`**. The line has the following characteristics:

   - It begins with the group name `juan`.
   - An exclamation point (`!`) appears in the password field of the **`/etc/gshadow`** file, which locks the group.
   - All other fields are blank.

5. A directory for user `juan` is created in the **`/home/`** directory. This directory is owned by user `juan` and group `juan`. However, it has read, write, and execute privileges *only* for the user `juan`. All other permissions are denied.

6. The files within the **`/etc/skel/`** directory (which contain default user settings) are copied into the new **`/home/juan/`** directory.

At this point, a locked account called `juan` exists on the system. To activate it, the administrator must next assign a password to the account using the **`passwd`** command and, optionally, set password aging guidelines.

22.3. Standard Users

Table 22.4, “Standard Users” lists the standard users configured in the **`/etc/passwd`** file by an **Everything** installation. The groupid (GID) in this table is the *primary group* for the user. See *Section 22.4, “Standard Groups”* for a listing of standard groups.

User	UID	GID	Home Directory	Shell
root	0	0	**`/root`**	**`/bin/bash`**
bin	1	1	**`/bin`**	**`/sbin/nologin`**

User	UID	GID	Home Directory	Shell
daemon	2	2	**/sbin**	**/sbin/nologin**
adm	3	4	**/var/adm**	**/sbin/nologin**
lp	4	7	**/var/spool/lpd**	**/sbin/nologin**
sync	5	0	**/sbin**	**/bin/sync**
shutdown	6	0	**/sbin**	**/sbin/shutdown**
halt	7	0	**/sbin**	**/sbin/halt**
mail	8	12	**/var/spool/mail**	**/sbin/nologin**
news	9	13	**/etc/news**	
uucp	10	14	**/var/spool/uucp**	**/sbin/nologin**
operator	11	0	**/root**	**/sbin/nologin**
games	12	100	**/usr/games**	**/sbin/nologin**
gopher	13	30	**/var/gopher**	**/sbin/nologin**
ftp	14	50	**/var/ftp**	**/sbin/nologin**
nobody	99	99	**/**	**/sbin/nologin**
rpm	37	37	**/var/lib/rpm**	**/sbin/nologin**
vcsa	69	69	**/dev**	**/sbin/nologin**
dbus	81	81	**/**	**/sbin/nologin**
ntp	38	38	**/etc/ntp**	**/sbin/nologin**
canna	39	39	**/var/lib/canna**	**/sbin/nologin**
nscd	28	28	**/**	**/sbin/nologin**
rpc	32	32	**/**	**/sbin/nologin**
postfix	89	89	**/var/spool/postfix**	**/sbin/nologin**
mailman	41	41	**/var/mailman**	**/sbin/nologin**
named	25	25	**/var/named**	**/bin/false**
amanda	33	6	**var/lib/amanda/**	**/bin/bash**
postgres	26	26	**/var/lib/pgsql**	**/bin/bash**
exim	93	93	**/var/spool/exim**	**/sbin/nologin**
sshd	74	74	**/var/empty/sshd**	**/sbin/nologin**
rpcuser	29	29	**/var/lib/nfs**	**/sbin/nologin**
nsfnobody	65534	65534	**/var/lib/nfs**	**/sbin/nologin**
pvm	24	24	**/usr/share/pvm3**	**/bin/bash**
apache	48	48	**/var/www**	**/sbin/nologin**
xfs	43	43	**/etc/X11/fs**	**/sbin/nologin**
gdm	42	42	**/var/gdm**	**/sbin/nologin**
htt	100	101	**/usr/lib/im**	**/sbin/nologin**
mysql	27	27	**/var/lib/mysql**	**/bin/bash**
webalizer	67	67	**/var/www/usage**	**/sbin/nologin**
mailnull	47	47	**/var/spool/mqueue**	**/sbin/nologin**

User	UID	GID	Home Directory	Shell
smmsp	51	51	`/var/spool/mqueue`	`/sbin/nologin`
squid	23	23	`/var/spool/squid`	`/sbin/nologin`
ldap	55	55	`/var/lib/ldap`	`/bin/false`
netdump	34	34	`/var/crash`	`/bin/bash`
pcap	77	77	`/var/arpwatch`	`/sbin/nologin`
radiusd	95	95	`/`	`/bin/false`
radvd	75	75	`/`	`/sbin/nologin`
quagga	92	92	`/var/run/quagga`	`/sbin/login`
wnn	49	49	`/var/lib/wnn`	`/sbin/nologin`
dovecot	97	97	`/usr/libexec/dovecot`	`/sbin/nologin`

Table 22.4. Standard Users

22.4. Standard Groups

Table 22.5, “Standard Groups” lists the standard groups configured by an **Everything** installation. Groups are stored in the **`/etc/group`** file.

Group	GID	Members
root	0	root
bin	1	root, bin, daemon
daemon	2	root, bin, daemon
sys	3	root, bin, adm
adm	4	root, adm, daemon
tty	5	
disk	6	root
lp	7	daemon, lp
mem	8	
kmem	9	
wheel	10	root
mail	12	mail, postfix, exim
news	13	news
uucp	14	uucp
man	15	
games	20	
gopher	30	
dip	40	
ftp	50	
lock	54	
nobody	99	
users	100	

Group	GID	Members
rpm	37	
utmp	22	
floppy	19	
vcsa	69	
dbus	81	
ntp	38	
canna	39	
nscd	28	
rpc	32	
postdrop	90	
postfix	89	
mailman	41	
exim	93	
named	25	
postgres	26	
sshd	74	
rpcuser	29	
nfsnobody	65534	
pvm	24	
apache	48	
xfs	43	
gdm	42	
htt	101	
mysql	27	
webalizer	67	
mailnull	47	
smmsp	51	
squid	23	
ldap	55	
netdump	34	
pcap	77	
quaggavt	102	
quagga	92	
radvd	75	
slocate	21	
wnn	49	
dovecot	97	

Group	GID	Members
radiusd	95	

Table 22.5. Standard Groups

22.5. User Private Groups

Fedora uses a *user private group* (*UPG*) scheme, which makes UNIX groups easier to manage.

A UPG is created whenever a new user is added to the system. A UPG has the same name as the user for which it was created and that user is the only member of the UPG.

UPGs make it safe to set default permissions for a newly created file or directory, allowing both the user and *the group of that user* to make modifications to the file or directory.

The setting which determines what permissions are applied to a newly created file or directory is called a *umask* and is configured in the **`/etc/bashrc`** file. Traditionally on UNIX systems, the **`umask`** is set to **`022`**, which allows only the user who created the file or directory to make modifications. Under this scheme, all other users, *including members of the creator's group*, are not allowed to make any modifications. However, under the UPG scheme, this "group protection" is not necessary since every user has their own private group.

22.5.1. Group Directories

Many IT organizations like to create a group for each major project and then assign people to the group if they need to access that project's files. Using this traditional scheme, managing files has been difficult; when someone creates a file, it is associated with the primary group to which they belong. When a single person works on multiple projects, it is difficult to associate the right files with the right group. Using the UPG scheme, however, groups are automatically assigned to files created within a directory with the *setgid* bit set. The setgid bit makes managing group projects that share a common directory very simple because any files a user creates within the directory are owned by the group which owns the directory.

Let us say, for example, that a group of people need to work on files in the **`/usr/share/emacs/site-lisp/`** directory. Some people are trusted to modify the directory, but certainly not everyone is trusted. First create an `emacs` group, as in the following command:

```
/usr/sbin/groupadd emacs
```

To associate the contents of the directory with the `emacs` group, type:

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

Now, it is possible to add the proper users to the group with the **`gpasswd`** command:

```
/usr/bin/gpasswd -a <username> emacs
```

To allow users to create files within the directory, use the following command:

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

When a user creates a new file, it is assigned the group of the user's default private group. Next, set the setgid bit, which assigns everything created in the directory the same group permission as the directory itself (`emacs`). Use the following command:

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

At this point, because the default umask of each user is 002, all members of the `emacs` group can create and edit files in the **`/usr/share/emacs/site-lisp/`** directory without the administrator having to change file permissions every time users write new files.

22.6. Shadow Passwords

In multiuser environments it is very important to use *shadow passwords* (provided by the **`shadow-utils`** package). Doing so enhances the security of system authentication files. For this reason, the installation program enables shadow passwords by default.

The following lists the advantages pf shadow passwords have over the traditional way of storing passwords on UNIX-based systems:

- Improves system security by moving encrypted password hashes from the world-readable **`/etc/passwd`** file to **`/etc/shadow`**, which is readable only by the root user.
- Stores information about password aging.
- Allows the use the **`/etc/login.defs`** file to enforce security policies.

Most utilities provided by the **`shadow-utils`** package work properly whether or not shadow passwords are enabled. However, since password aging information is stored exclusively in the **`/etc/shadow`** file, any commands which create or modify password aging information do not work.

The following is a list of commands which do not work without first enabling shadow passwords:

- **`chage`**
- **`gpasswd`**
- **`/usr/sbin/usermod`** `-e` or `-f` options
- **`/usr/sbin/useradd`** `-e` or `-f` options

22.7. Additional Resources

For more information about users and groups, and tools to manage them, refer to the following resources.

22.7.1. Installed Documentation

- Related man pages — There are a number of man pages for the various applications and configuration files involved with managing users and groups. Some of the more important man pages have been listed here:

  User and Group Administrative Applications

  - **`man chage`** — A command to modify password aging policies and account expiration.
  - **`man gpasswd`** — A command to administer the **`/etc/group`** file.
  - **`man groupadd`** — A command to add groups.
  - **`man grpck`** — A command to verify the **`/etc/group`** file.
  - **`man groupdel`** — A command to remove groups.
  - **`man groupmod`** — A command to modify group membership.
  - **`man pwck`** — A command to verify the **`/etc/passwd`** and **`/etc/shadow`** files.
  - **`man pwconv`** — A tool to convert standard passwords to shadow passwords.
  - **`man pwunconv`** — A tool to convert shadow passwords to standard passwords.
  - **`man useradd`** — A command to add users.
  - **`man userdel`** — A command to remove users.
  - **`man usermod`** — A command to modify users.

  Configuration Files

  - **`man 5 group`** — The file containing group information for the system.
  - **`man 5 passwd`** — The file containing user information for the system.
  - **`man 5 shadow`** — The file containing passwords and account expiration information for the system.

Printer Configuration

Printer Configuration Tool allows users to configure a printer. This tool helps maintain the printer configuration file, print spool directories, print filters, and printer classes.

Fedora 12 uses the Common Unix Printing System (CUPS). If a system was upgraded from a previous Fedora version that used CUPS, the upgrade process preserves the configured queues.

Using **Printer Configuration Tool** requires root privileges. To start the application, select System (on the panel) > **Administration** > **Printing**, or type the command **`system-config-printer`** at a shell prompt.

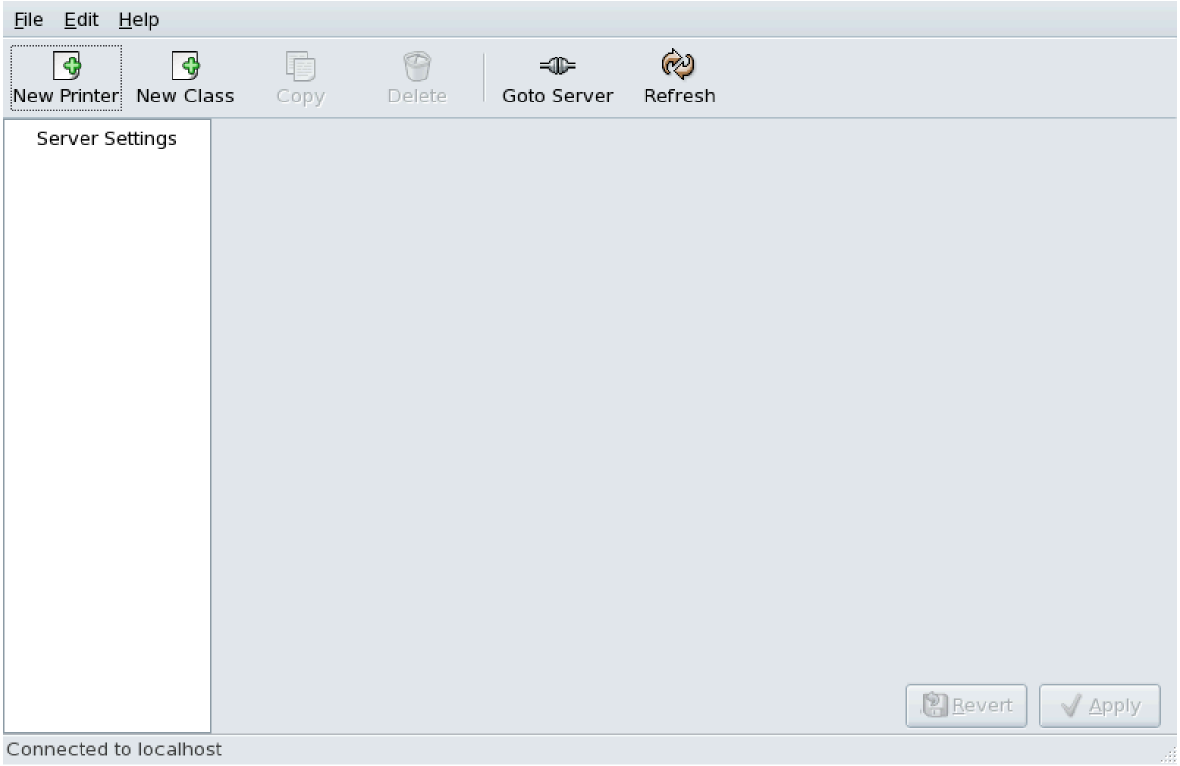


Figure 23.1. **Printer Configuration Tool**

The following types of print queues can be configured:

- **AppSocket/HP JetDirect** — a printer connected directly to the network through HP JetDirect or Appsocket interface instead of a computer.
- **Internet Printing Protocol (IPP)** — a printer that can be accessed over a TCP/IP network via the Internet Printing Protocol (for example, a printer attached to another Fedora system running CUPS on the network).
- **LPD/LPR Host or Printer** — a printer attached to a different UNIX system that can be accessed over a TCP/IP network (for example, a printer attached to another Fedora system running LPD on the network).
- **Networked Windows (SMB)** — a printer attached to a different system which is sharing a printer over an SMB network (for example, a printer attached to a Microsoft Windows™ machine).

- **Networked JetDirect** — a printer connected directly to the network through HP JetDirect instead of a computer.

Important

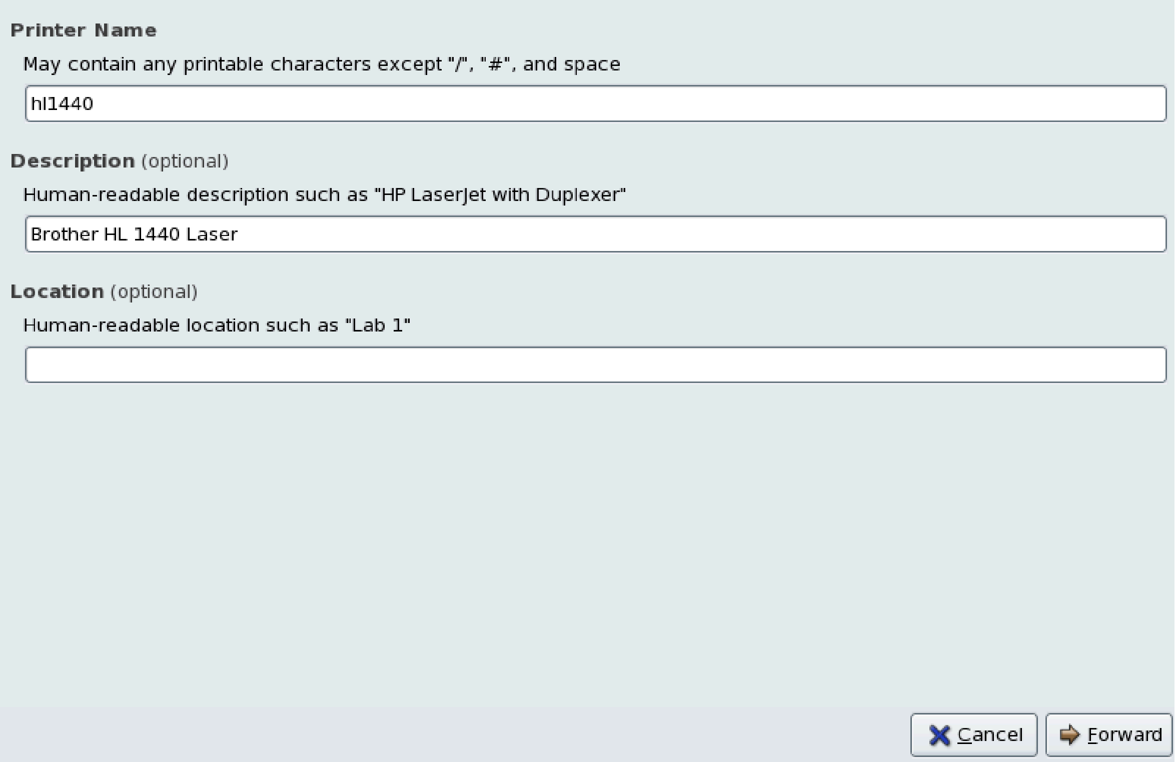
If you add a new print queue or modify an existing one, you must apply the changes for them to take effect.

Clicking the **Apply** button prompts the printer daemon to restart with the changes you have configured.

Clicking the **Revert** button discards unapplied changes.

23.1. Adding a Local Printer

To add a local printer, such as one attached through a parallel port or USB port on your computer, click the **New Printer** button in the main **Printer Configuration Tool** window to display the window in *Figure 23.2, "**Adding a Printer**"*.



Printer Name

May contain any printable characters except "/", "#", and space

hl1440

Description (optional)

Human-readable description such as "HP LaserJet with Duplexer"

Brother HL 1440 Laser

Location (optional)

Human-readable location such as "Lab 1"

Figure 23.2. **Adding a Printer**

Click **Forward** to proceed.

Enter a unique name for the printer in the **Printer Name** field. The printer name can contain letters, numbers, dashes (-), and underscores (_); it *must not* contain any spaces.

You can also use the **Description** and **Location** fields to further distinguish this printer from others that may be configured on your system. Both of these fields are optional, and may contain spaces.

Click **Forward** to open the **New Printer** dialogue (refer to *Figure 23.3, “Adding a Local Printer”*). If the printer has been automatically detected, the printer model appears in **Select Connection**. Select the printer model and click **Forward** to continue.

If the device does not automatically appear, select the device to which the printer is connected (such as **LPT #1** or **Serial Port #1**) in **Select Connection**.

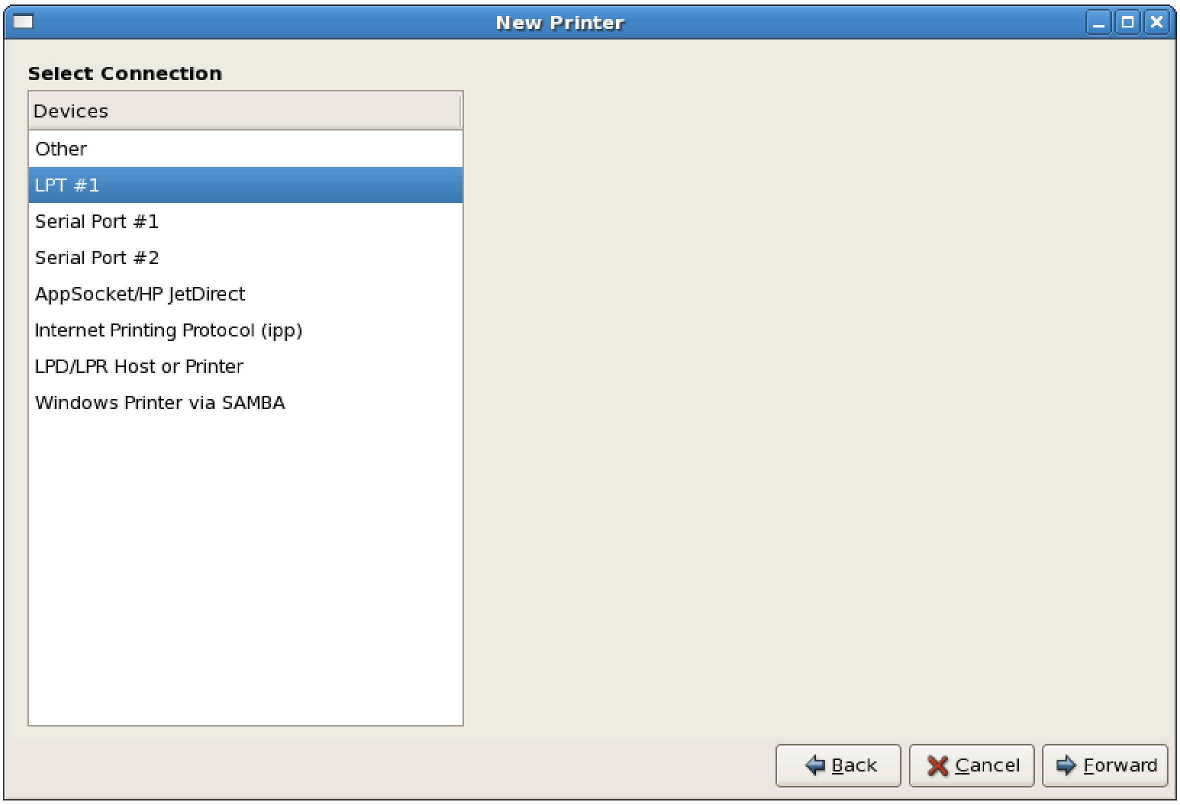


Figure 23.3. Adding a Local Printer

Next, select the printer type. Refer to *Section 23.5, “Selecting the Printer Model and Finishing”* for details.

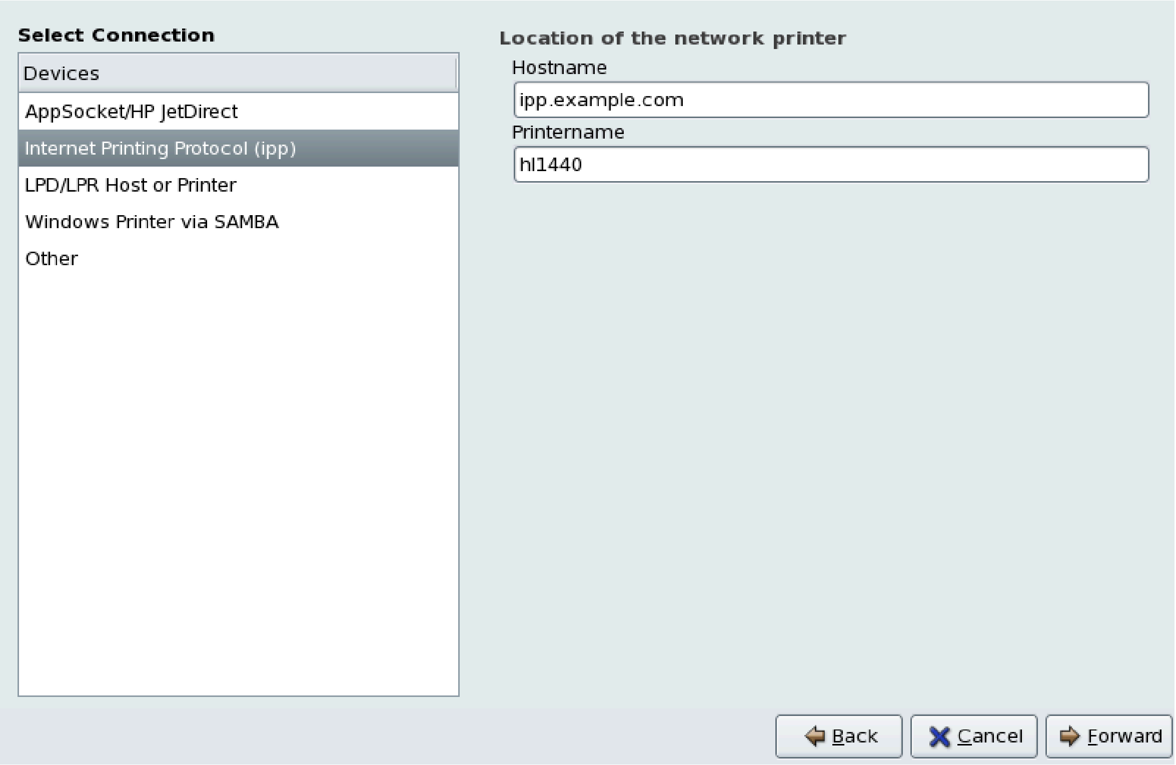
23.2. Adding an IPP Printer

An IPP printer is a printer attached to a different system on the same TCP/IP network. The system this printer is attached to may either be running CUPS or simply configured to use IPP.

If a firewall is enabled on the printer server, then the firewall should be configured to allow send / receive connections on the incoming UDP port 631. If a firewall is enabled on the client (the system sending the print request) then the firewall should be configured to allow accept and create connections through port 631.

You can add a networked IPP printer by clicking the **New Printer** button in the main **Printer Configuration Tool** window to display the window in *Figure 23.2, “**Adding a Printer**”*. Enter the **Printer Name** (printer names cannot contain spaces and may contain letters, numbers, dashes (-), and underscores (_)), **Description**, and **Location** to distinguish this printer from others that you may configure on your system. Click **Forward** to proceed.

In the window shown in *Figure 23.4, “Adding an IPP Printer”*, enter the hostname of the IPP printer in the **Hostname** field as well as a unique name for the printer in the **Printername** field.



Internet Printing Protocol (ipp)

Location of the network printer

Hostname: ipp.example.com

Printername: hl1440

Figure 23.4. Adding an IPP Printer

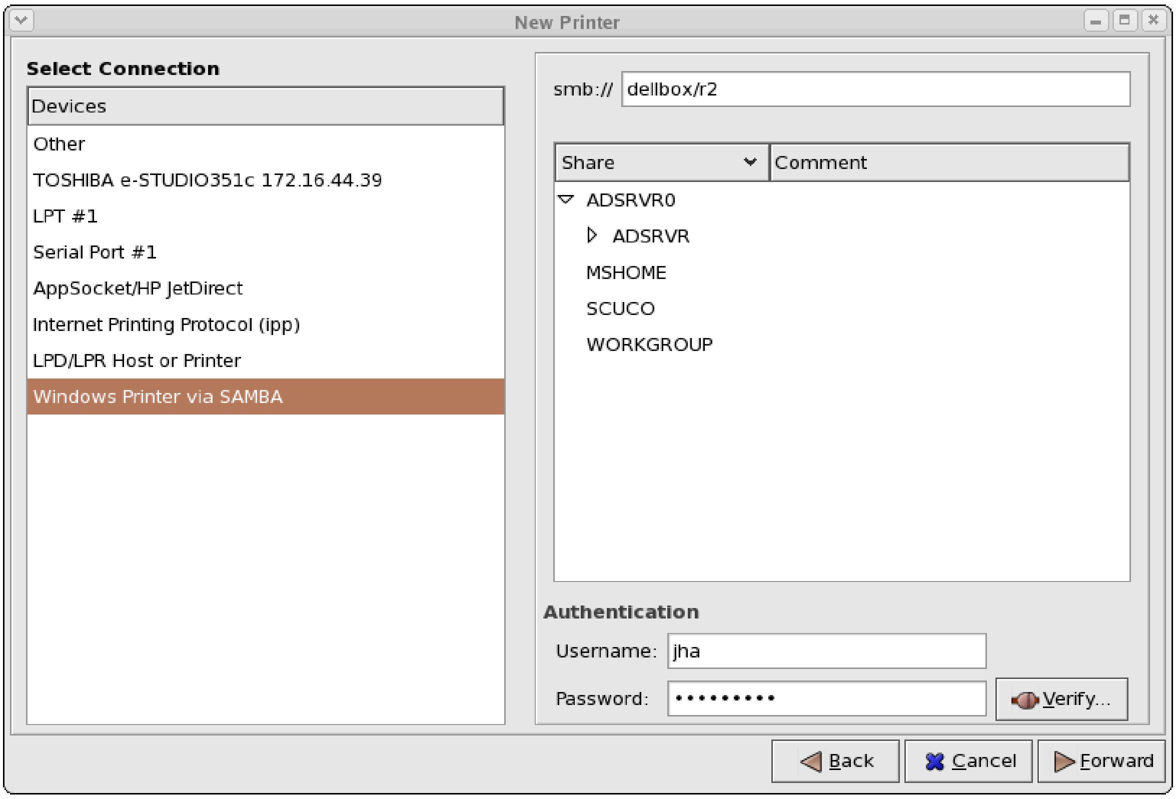
Click **Forward** to continue.

Next, select the printer type. Refer to *Section 23.5, “Selecting the Printer Model and Finishing”* for details.

23.3. Adding a Samba (SMB) Printer

You can add a Samba (SMB) based printer share by clicking the **New Printer** button in the main **Printer Configuration Tool** window to display the window in *Figure 23.2, “**Adding a Printer**”*. Enter a unique name for the printer in the **Printer Name** field. The printer name can contain letters, numbers, dashes (-), and underscores (_); it *must not* contain any spaces.

You can also use the **Description** and **Location** fields to further distinguish this printer from others that may be configured on your system. Both of these fields are optional, and may contain spaces.



Select Connection

- Devices
- Other
- TOSHIBA e-STUDIO351c 172.16.44.39
- LPT #1
- Serial Port #1
- AppSocket/HP JetDirect
- Internet Printing Protocol (ipp)
- LPD/LPR Host or Printer
- Windows Printer via SAMBA

smb:// dellbox/r2

Share	Comment
▽ ADSRVR0	
▷ ADSRVR	
MSHOME	
SCUCO	
WORKGROUP	

Authentication

Username: jha

Password: ••••••••••

Figure 23.5. Adding a SMB Printer

As shown in *Figure 23.5, “Adding a SMB Printer”*, available SMB shares are automatically detected and listed in the **Share** column. Click the arrow (▷) beside a Workgroup to expand it. From the expanded list, select a printer.

If the printer you are looking for does not appear in the list, enter the SMB address in the **smb://** field. Use the format *`computer name/printer share`*. In *Figure 23.5, “Adding a SMB Printer”*, the *`computer name`* is **`dellbox`**, while the *`printer share`* is **`r2`**.

In the **Username** field, enter the username to access the printer. This user must exist on the SMB system, and the user must have permission to access the printer. The default user name is typically **`guest`** for Windows servers, or **nobody** for Samba servers.

Enter the **Password** (if required) for the user specified in the **Username** field.

You can then test the connection by clicking **Verify**. Upon successful verification, a dialog box appears confirming printer share accessibility.

Next, select the printer type. Refer to *Section 23.5, “Selecting the Printer Model and Finishing”* for details.

Warning

Samba printer usernames and passwords are stored in the printer server as unencrypted files readable by root and lpd. Thus, other users that have root access to the printer server can view the username and password you use to access the Samba printer.

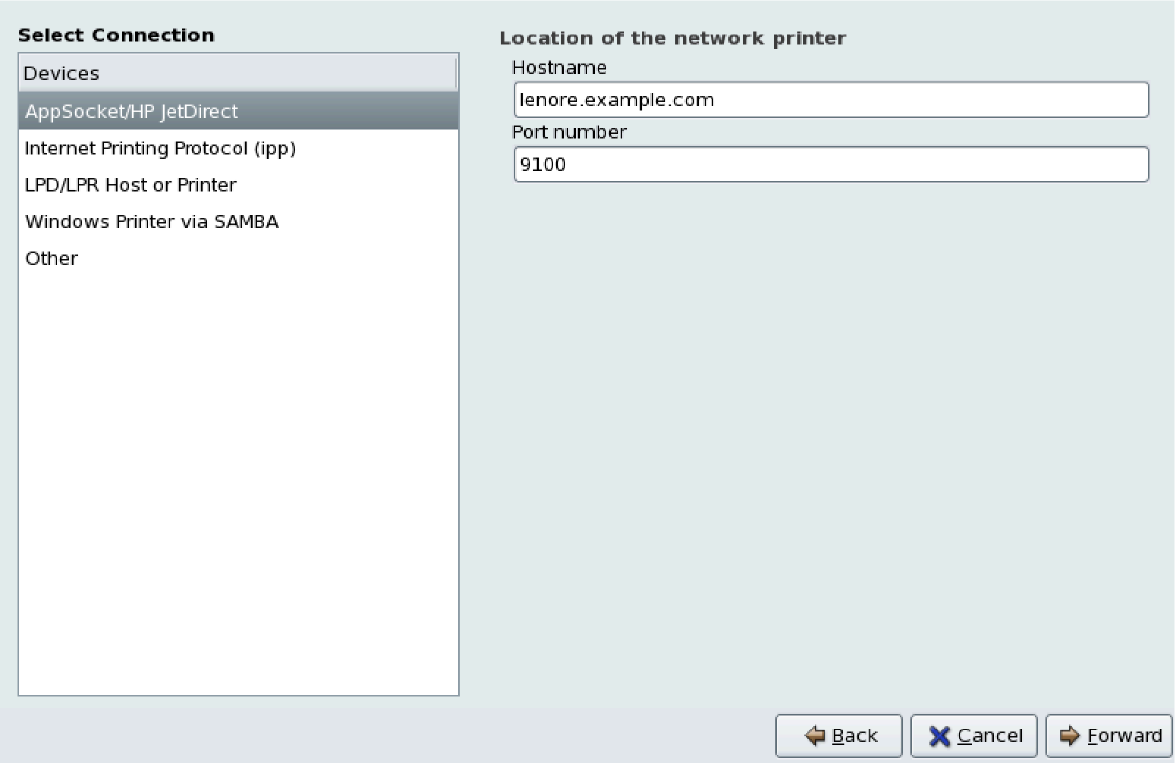
As such, when you choose a username and password to access a Samba printer, it is advisable that you choose a password that is different from what you use to access your local Fedora system.

If there are files shared on the Samba print server, it is recommended that they also use a password different from what is used by the print queue.

23.4. Adding a JetDirect Printer

To add a JetDirect or AppSocket connected printer share, click the **New Printer** button in the main **Printer Configuration Tool** window to display the window in *Figure 23.2, “**Adding a Printer**”*. Enter a unique name for the printer in the **Printer Name** field. The printer name can contain letters, numbers, dashes (-), and underscores (_); it *must not* contain any spaces.

You can also use the **Description** and **Location** fields to further distinguish this printer from others that may be configured on your system. Both of these fields are optional, and may contain spaces.



Select Connection

Devices

AppSocket/HP JetDirect

Internet Printing Protocol (ipp)

LPD/LPR Host or Printer

Windows Printer via SAMBA

Other

Location of the network printer

Hostname: lenore.example.com

Port number: 9100

Figure 23.6. Adding a JetDirect Printer

Click **Forward** to continue.

Text fields for the following options appear:

- **Hostname** — The hostname or IP address of the JetDirect printer.
- **Port Number** — The port on the JetDirect printer that is listening for print jobs. The default port is 9100.

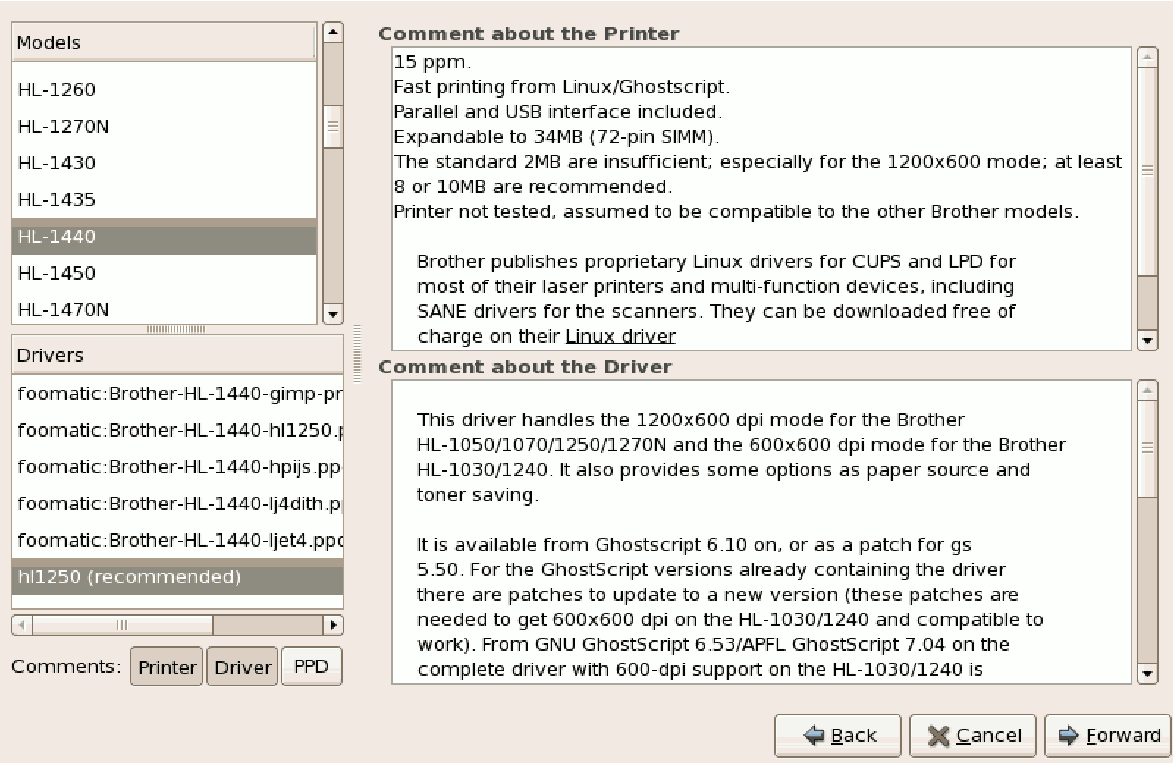
Next, select the printer type. Refer to *Section 23.5, “Selecting the Printer Model and Finishing”* for details.

23.5. Selecting the Printer Model and Finishing

Once you have properly selected a printer queue type, you can choose either option:

- Select a Printer from database - If you select this option, choose the make of your printer from the list of **Makes**. If your printer make is not listed, choose **Generic**.
- Provide PPD file - A PostScript Printer Description (PPD) file may also be provided with your printer. This file is normally provided by the manufacturer. If you are provided with a PPD file, you can choose this option and use the browser bar below the option description to select the PPD file.

Refer to *Figure 23.7, "Selecting a Printer Model"*.



Models

HL-1260
HL-1270N
HL-1430
HL-1435
HL-1440
HL-1450
HL-1470N

Drivers

foomatic:Brother-HL-1440-gimp-pr
foomatic:Brother-HL-1440-hl1250.
foomatic:Brother-HL-1440-hpijs.pp
foomatic:Brother-HL-1440-lj4dith.p
foomatic:Brother-HL-1440-ljet4.ppc
hl1250 (recommended)

Comment about the Printer

15 ppm.
Fast printing from Linux/Ghostscript.
Parallel and USB interface included.
Expandable to 34MB (72-pin SIMM).
The standard 2MB are insufficient; especially for the 1200x600 mode; at least 8 or 10MB are recommended.
Printer not tested, assumed to be compatible to the other Brother models.

Brother publishes proprietary Linux drivers for CUPS and LPD for most of their laser printers and multi-function devices, including SANE drivers for the scanners. They can be downloaded free of charge on their Linux driver

Comment about the Driver

This driver handles the 1200x600 dpi mode for the Brother HL-1050/1070/1250/1270N and the 600x600 dpi mode for the Brother HL-1030/1240. It also provides some options as paper source and toner saving.

It is available from Ghostscript 6.10 on, or as a patch for gs 5.50. For the GhostScript versions already containing the driver there are patches to update to a new version (these patches are needed to get 600x600 dpi on the HL-1030/1240 and compatible to work). From GNU GhostScript 6.53/APFL GhostScript 7.04 on the complete driver with 600-dpi support on the HL-1030/1240 is

Figure 23.7. Selecting a Printer Model

After choosing an option, click **Forward** to continue. *Figure 23.7, "Selecting a Printer Model"* appears. You now have to choose the corresponding model and driver for the printer.

The recommended printed driver is automatically selected based on the printer model you chose. The print driver processes the data that you want to print into a format the printer can understand. Since a local printer is attached directly to your computer, you need a printer driver to process the data that is sent to the printer.

If you have a PPD file for the device (usually provided by the manufacturer), you can select it by choosing **Provide PPD file**. You can then browse the filesystem for the PPD file by clicking **Browse**.

23.5.1. Confirming Printer Configuration

The last step is to confirm your printer configuration. Click **Apply** to add the print queue if the settings are correct. Click **Back** to modify the printer configuration.

After applying the changes, print a test page to ensure the configuration is correct. Refer to *Section 23.6, “Printing a Test Page”* for details.

23.6. Printing a Test Page

After you have configured your printer, you should print a test page to make sure the printer is functioning properly. To print a test page, select the printer that you want to try out from the printer list, then click **Print Test Page** from the printer's **Settings** tab.

If you change the print driver or modify the driver options, you should print a test page to test the different configuration.

23.7. Modifying Existing Printers

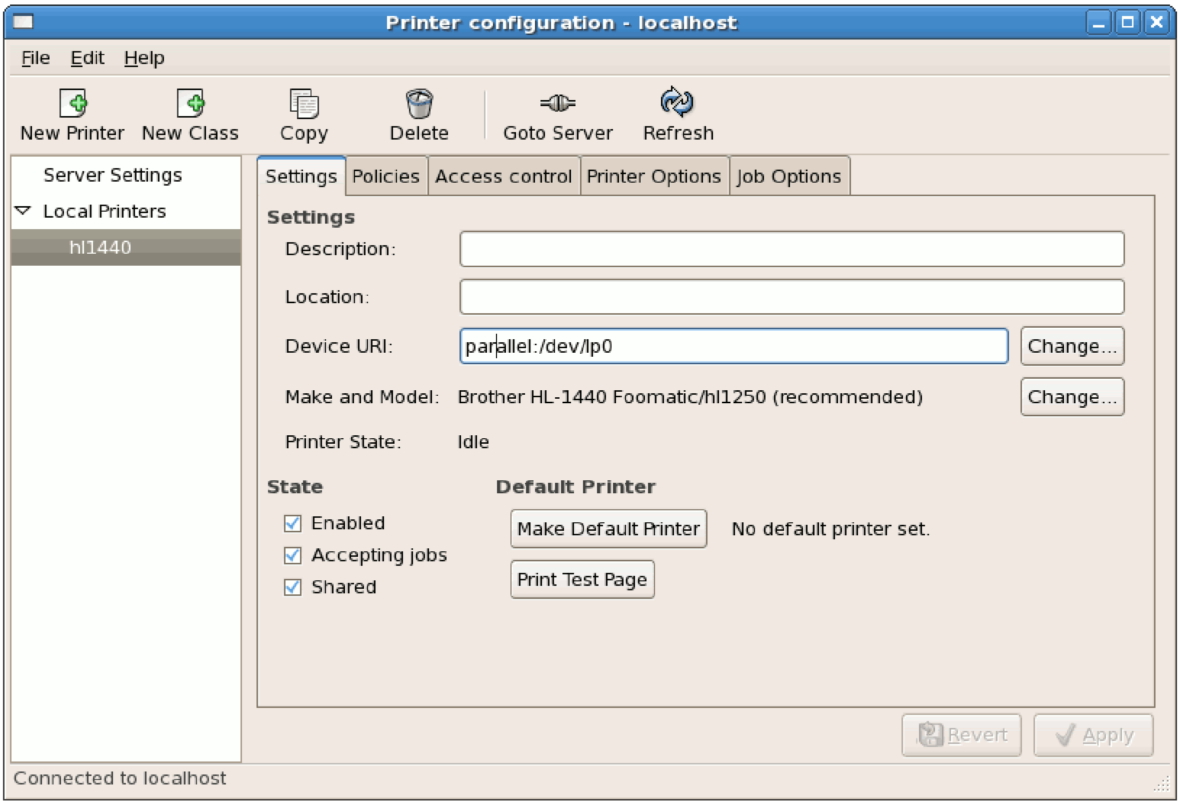
To delete an existing printer, select the printer and click the **Delete** button on the toolbar. The printer is removed from the printer list once you confirm deletion of the printer configuration.

To set the default printer, select the printer from the printer list and click the **Make Default Printer** button in the **Settings** tab.

23.7.1. The Settings Tab

To change printer driver configuration, click the corresponding name in the **Printer** list and click the **Settings** tab.

You can modify printer settings such as make and model, make a printer the default, print a test page, change the device location (URI), and more.



Description:

Location:

Device URI: parallel:/dev/lp0

Make and Model: Brother HL-1440 Foomatic/hl1250 (recommended)

Printer State: Idle

☑ Enabled
☑ Accepting jobs
☑ Shared

No default printer set.

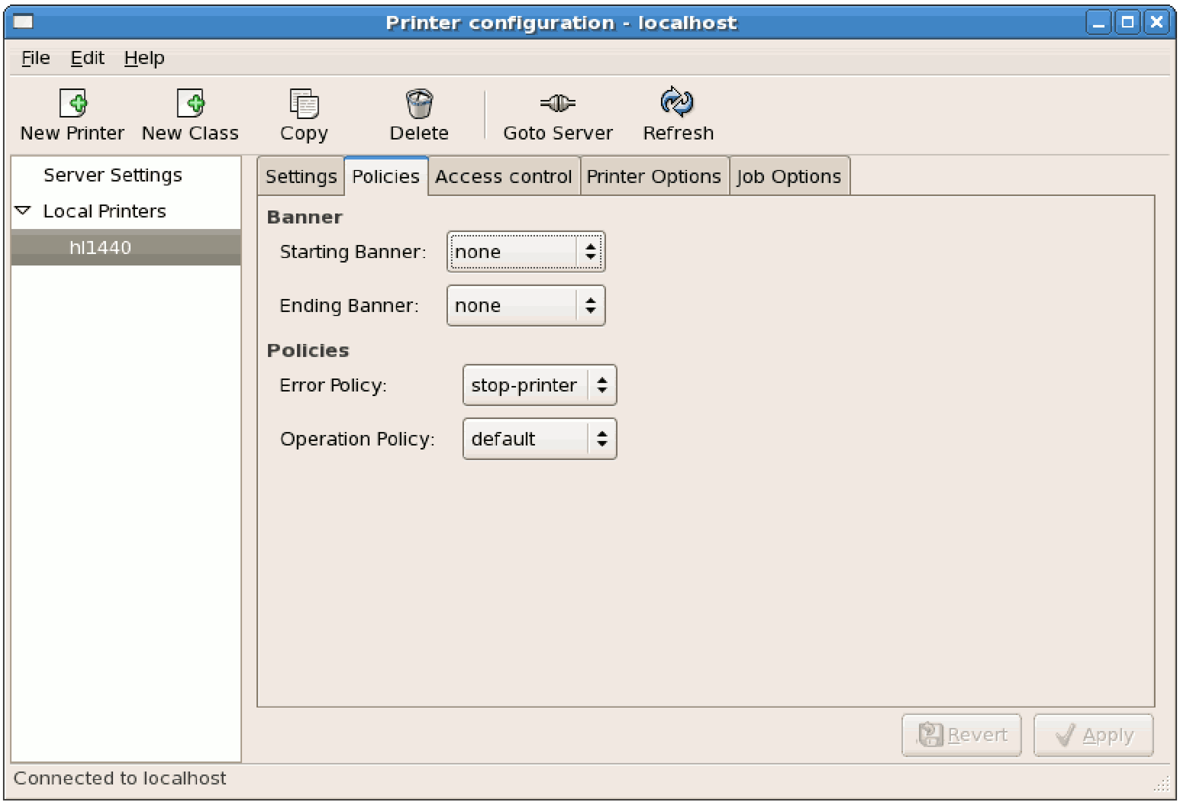
Connected to localhost

Figure 23.8. Settings Tab

23.7.2. The Policies Tab

To change settings in print output, click the **Policies** tab.

For example, to create a *banner page* (a page that describes aspects of the print job such as the originating printer, the username from the which the job originated, and the security status of the document being printed) click the **Starting Banner** or **Ending Banner** drop-menu and choose the option that best describes the nature of the print jobs (such as **topsecret**, **classified**, or **confidential**).



Banner

Starting Banner: none

Ending Banner: none

Policies

Error Policy: stop-printer

Operation Policy: default

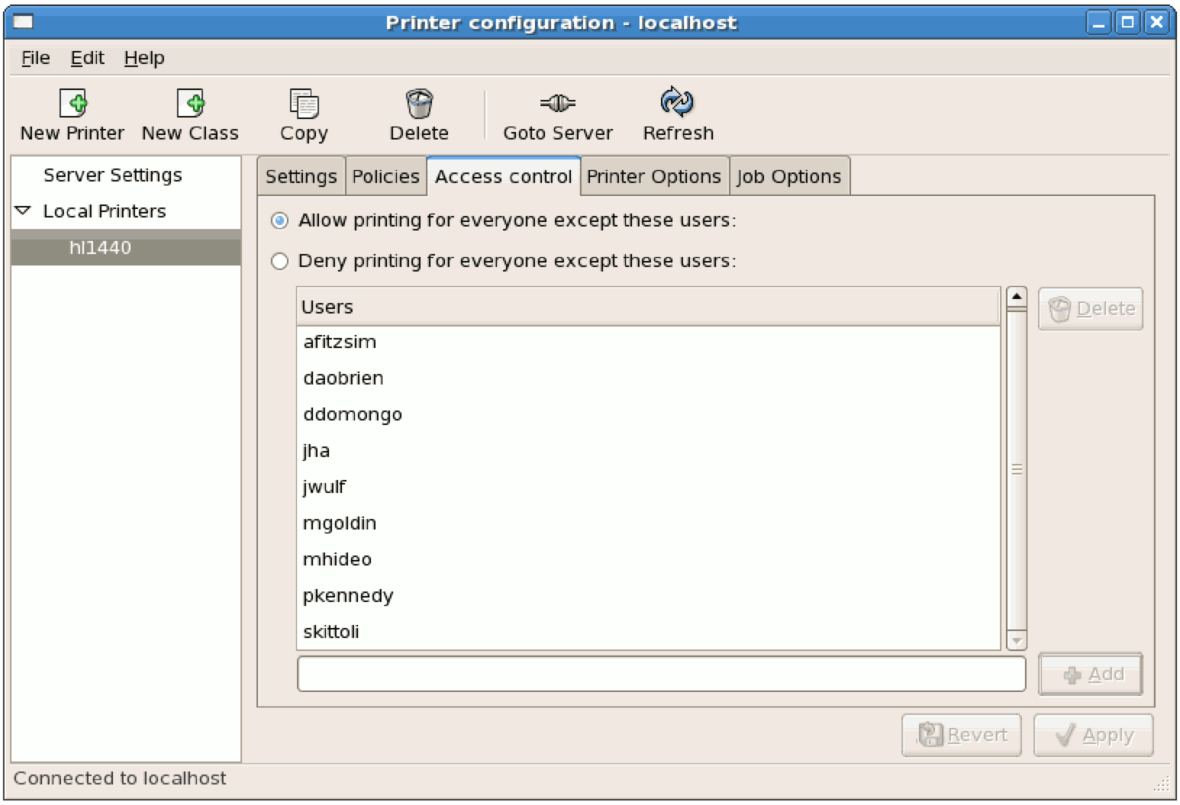
Figure 23.9. Policies Tab

You can also configure the **Error Policy** of the printer, by choosing an option from the drop-down menu. You can choose to abort the print job, retry, or stop it.

23.7.3. The Access Control Tab

You can change user-level access to the configured printer by clicking the **Access Control** tab.

Add users using the text box and click the **Add** button beside it. You can then choose to only allow use of the printer to that subset of users or deny use to those users.



● Allow printing for everyone except these users:

○ Deny printing for everyone except these users:

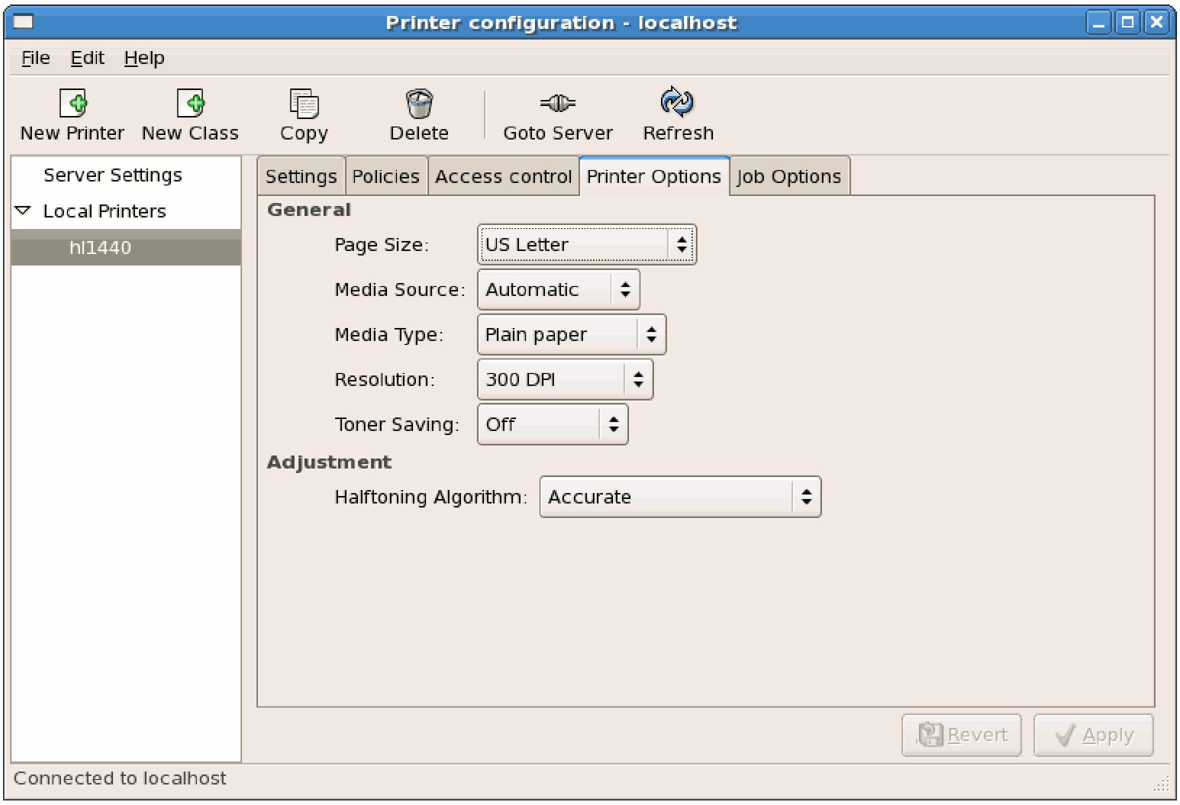
Users
afitzsim
daobrien
ddomongo
jha
jwulf
mgoldin
mhideo
pkennedy
skittoli

Connected to localhost

Figure 23.10. Access Control Tab

23.7.4. The Printer and Job OptionsTab

The **Printer Options** tab contains various configuration options for the printer media and output.



General

Page Size: US Letter

Media Source: Automatic

Media Type: Plain paper

Resolution: 300 DPI

Toner Saving: Off

Adjustment

Halftoning Algorithm: Accurate

Figure 23.11. Printer Options Tab

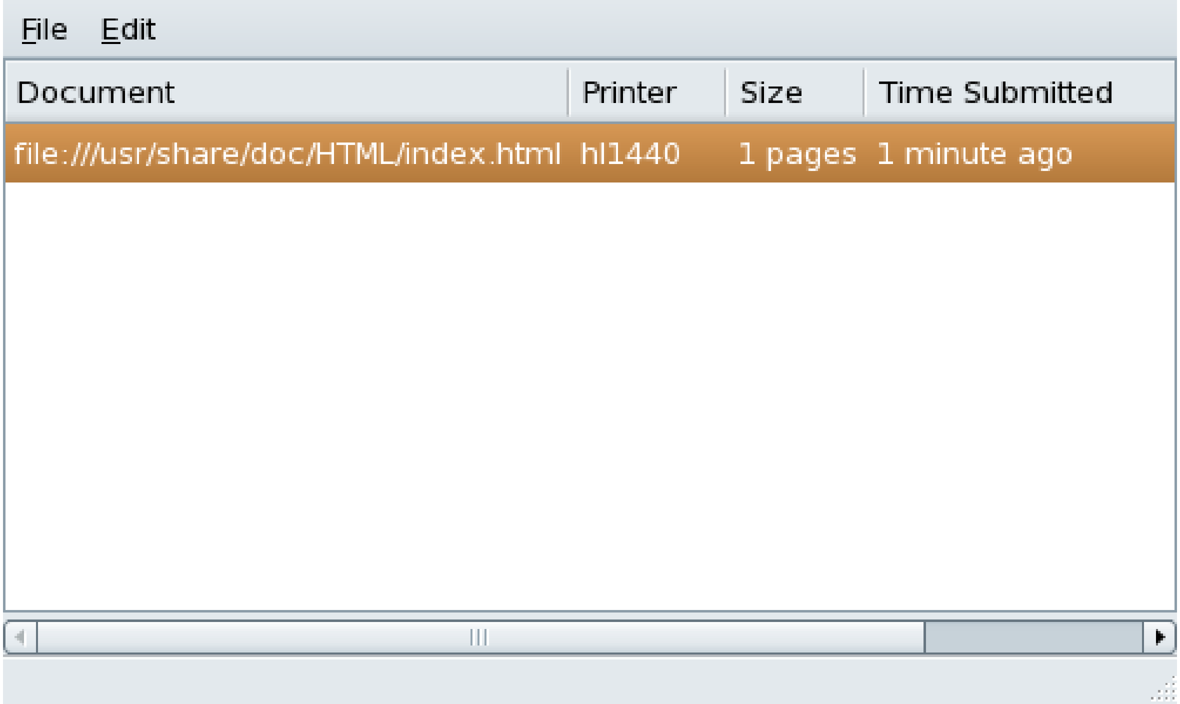
- **Page Size** — Allows the paper size to be selected. The options include US Letter, US Legal, A3, and A4
- **Media Source** — set to **Automatic** by default. Change this option to use paper from a different tray.
- **Media Type** — Allows you to change paper type. Options include: Plain, thick, bond, and transparency.
- **Resolution** — Configure the quality and detail of the printout. Default is 300 dots per inch (dpi).
- **Toner Saving** — Choose whether the printer uses less toner to conserve resources.

You can also configure printer job options using the **Job Options** tab. Use the drop-menu and choose the job options you wish to use, such as **Landscape** modes (horizontal or vertical printout), **copies**, or **scaling** (increase or decrease the size of the printable area, which can be used to fit an oversize print area onto a smaller physical sheet of print medium).

23.8. Managing Print Jobs

When you send a print job to the printer daemon, such as printing a text file from **Emacs** or printing an image from **The GIMP**, the print job is added to the print spool queue. The print spool queue is a list of print jobs that have been sent to the printer and information about each print request, such as the status of the request, the the job number, and more.

During the printing process, the Printer Status icon appears in the **Notification Area** on the panel. To check the status of a print job, double click the Printer Status, which displays a window similar to *Figure 23.12, “GNOME Print Status”*.



Document	Printer	Size	Time Submitted
file:///usr/share/doc/HTML/index.html	hl1440	1 pages	1 minute ago

Figure 23.12. GNOME Print Status

To cancel a specific print job listed in the **GNOME Print Status**, select it from the list and select **Edit** > **Cancel Documents** from the pulldown menu.

To view the list of print jobs in the print spool from a shell prompt, type the command **`lpq`**. The last few lines look similar to the following:

```
Rank   Owner/ID              Class  Job Files       Size Time
active user@localhost+902    A      902 sample.txt  2050 01:20:46
```

Example 23.1. Example of **`lpq`** output

If you want to cancel a print job, find the job number of the request with the command **`lpq`** and then use the command **`lprm`** ***`job number`***. For example, **`lprm 902`** would cancel the print job in *Example 23.1, “Example of **`lpq`** output”*. You must have proper permissions to cancel a print job. You can not cancel print jobs that were started by other users unless you are logged in as root on the machine to which the printer is attached.

You can also print a file directly from a shell prompt. For example, the command **`lpr sample.txt`** prints the text file **`sample.txt`**. The print filter determines what type of file it is and converts it into a format the printer can understand.

23.9. Additional Resources

To learn more about printing on Fedora, refer to the following resources.

23.9.1. Installed Documentation

- **`map lpr`** — The manual page for the **`lpr`** command that allows you to print files from the command line.

- **`man lprm`** — The manual page for the command line utility to remove print jobs from the print queue.
- **`man mpage`** — The manual page for the command line utility to print multiple pages on one sheet of paper.
- **`man cupsd`** — The manual page for the CUPS printer daemon.
- **`man cupsd.conf`** — The manual page for the CUPS printer daemon configuration file.
- **`man classes.conf`** — The manual page for the class configuration file for CUPS.

23.9.2. Useful Websites

- *http://www.linuxprinting.org* — *GNU/Linux Printing* contains a large amount of information about printing in Linux.
- *http://www.cups.org/* — Documentation, FAQs, and newsgroups about CUPS.

Automated Tasks

In Linux, tasks can be configured to run automatically within a specified period of time, on a specified date, or when the system load average is below a specified number. Fedora is pre-configured to run important system tasks to keep the system updated. For example, the slocate database used by the **`locate`** command is updated daily. A system administrator can use automated tasks to perform periodic backups, monitor the system, run custom scripts, and more.

Fedora comes with several automated tasks utilities: **`cron`**, **`at`**, and **`batch`**.

24.1. Cron

Cron is a daemon that can be used to schedule the execution of recurring tasks according to a combination of the time, day of the month, month, day of the week, and week.

Cron assumes that the system is on continuously. If the system is not on when a task is scheduled, it is not executed. To schedule one-time tasks, refer to *Section 24.2, “At and Batch”*.

To use the cron service, the **`vixie-cron`** RPM package must be installed and the **`crond`** service must be running. To determine if the package is installed, use the **`rpm -q vixie-cron`** command. To determine if the service is running, use the command **`/sbin/service crond status`**.

24.1.1. Configuring Cron Tasks

The main configuration file for cron, **`/etc/crontab`**, contains the following lines:

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root HOME=/
# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

The first four lines are variables used to configure the environment in which the cron tasks are run. The `SHELL` variable tells the system which shell environment to use (in this example the bash shell), while the `PATH` variable defines the path used to execute commands. The output of the cron tasks are emailed to the username defined with the `MAILTO` variable. If the `MAILTO` variable is defined as an empty string (`MAILTO=""`), email is not sent. The `HOME` variable can be used to set the home directory to use when executing commands or scripts.

Each line in the **`/etc/crontab`** file represents a task and has the following format:

```
minute   hour   day   month   dayofweek   command
```

- `minute` — any integer from 0 to 59
- `hour` — any integer from 0 to 23

- `day` — any integer from 1 to 31 (must be a valid day if a month is specified)
- `month` — any integer from 1 to 12 (or the short name of the month such as jan or feb)
- `dayofweek` — any integer from 0 to 7, where 0 or 7 represents Sunday (or the short name of the week such as sun or mon)
- `command` — the command to execute (the command can either be a command such as **`ls /proc >> /tmp/proc`** or the command to execute a custom script)

For any of the above values, an asterisk (*) can be used to specify all valid values. For example, an asterisk for the month value means execute the command every month within the constraints of the other values.

A hyphen (-) between integers specifies a range of integers. For example, **`1-4`** means the integers 1, 2, 3, and 4.

A list of values separated by commas (,) specifies a list. For example, **`3, 4, 6, 8`** indicates those four specific integers.

The forward slash (/) can be used to specify step values. The value of an integer can be skipped within a range by following the range with **`/<`*`integer`*`>`**. For example, **`0-59/2`** can be used to define every other minute in the minute field. Step values can also be used with an asterisk. For instance, the value **`*/3`** can be used in the month field to run the task every third month.

Any lines that begin with a hash mark (#) are comments and are not processed.

As shown in the **`/etc/crontab`** file, the **`run-parts`** script executes the scripts in the **`/etc/cron.hourly/`**, **`/etc/cron.daily/`**, **`/etc/cron.weekly/`**, and **`/etc/cron.monthly/`** directories on an hourly, daily, weekly, or monthly basis respectively. The files in these directories should be shell scripts.

If a cron task is required to be executed on a schedule other than hourly, daily, weekly, or monthly, it can be added to the **`/etc/cron.d/`** directory. All files in this directory use the same syntax as **`/etc/crontab`**. Refer to *Example 24.1, “Crontab Examples”* for examples.

```
# record the memory usage of the system every monday
# at 3:30AM in the file /tmp/meminfo
30 3 * * mon cat /proc/meminfo >> /tmp/meminfo
# run custom script the first day of every month at 4:10AM
10 4 1 * * /root/scripts/backup.sh
```

Example 24.1. Crontab Examples

Users other than root can configure cron tasks by using the **`crontab`** utility. All user-defined crontabs are stored in the **`/var/spool/cron/`** directory and are executed using the usernames of the users that created them. To create a crontab as a user, login as that user and type the command **`crontab -e`** to edit the user's crontab using the editor specified by the `VISUAL` or `EDITOR` environment variable. The file uses the same format as **`/etc/crontab`**. When the changes to the crontab are saved, the crontab is stored according to username and written to the file **`/var/spool/cron/`*`username`***.

The cron daemon checks the **/etc/crontab** file, the **/etc/cron.d/** directory, and the **/var/spool/cron/** directory every minute for any changes. If any changes are found, they are loaded into memory. Thus, the daemon does not need to be restarted if a crontab file is changed.

24.1.2. Controlling Access to Cron

The **/etc/cron.allow** and **/etc/cron.deny** files are used to restrict access to cron. The format of both access control files is one username on each line. Whitespace is not permitted in either file. The cron daemon (**crond**) does not have to be restarted if the access control files are modified. The access control files are read each time a user tries to add or delete a cron task.

The root user can always use cron, regardless of the usernames listed in the access control files.

If the file **cron.allow** exists, only users listed in it are allowed to use cron, and the **cron.deny** file is ignored.

If **cron.allow** does not exist, users listed in **cron.deny** are not allowed to use cron.

24.1.3. Starting and Stopping the Service

To start the cron service, use the command **/sbin/service crond start**. To stop the service, use the command **/sbin/service crond stop**. It is recommended that you start the service at boot time. Refer to *Chapter 6, Controlling Access to Services* for details on starting the cron service automatically at boot time.

24.2. At and Batch

While cron is used to schedule recurring tasks, the **at** command is used to schedule a one-time task at a specific time and the **batch** command is used to schedule a one-time task to be executed when the systems load average drops below 0.8.

To use **at** or **batch**, the **at** RPM package must be installed, and the **atd** service must be running. To determine if the package is installed, use the **rpm -q at** command. To determine if the service is running, use the command **/sbin/service atd status**.

24.2.1. Configuring At Jobs

To schedule a one-time job at a specific time, type the command **at *time***, where ***time*** is the time to execute the command.

The argument *time* can be one of the following:

- HH:MM format — For example, 04:00 specifies 4:00 a.m. If the time is already past, it is executed at the specified time the next day.

- midnight — Specifies 12:00 a.m.

- noon — Specifies 12:00 p.m.

- teatime — Specifies 4:00 p.m.

- month-name day year format — For example, January 15 2002 specifies the 15th day of January in the year 2002. The year is optional.

- MMDDYY, MM/DD/YY, or MM.DD.YY formats — For example, 011502 for the 15th day of January in the year 2002.
- now + time — time is in minutes, hours, days, or weeks. For example, now + 5 days specifies that the command should be executed at the same time five days from now.

The time must be specified first, followed by the optional date. For more information about the time format, read the **/usr/share/doc/at-<*version*>/timespec** text file.

After typing the **at** command with the time argument, the at> prompt is displayed. Type the command to execute, press **Enter**, and type **Ctrl**+**D** . Multiple commands can be specified by typing each command followed by the **Enter** key. After typing all the commands, press **Enter** to go to a blank line and type **Ctrl**+**D** . Alternatively, a shell script can be entered at the prompt, pressing **Enter** after each line in the script, and typing **Ctrl**+**D** on a blank line to exit. If a script is entered, the shell used is the shell set in the user's SHELL environment, the user's login shell, or **/bin/sh** (whichever is found first).

If the set of commands or script tries to display information to standard out, the output is emailed to the user.

Use the command **atq** to view pending jobs. Refer to *Section 24.2.3, “Viewing Pending Jobs”* for more information.

Usage of the **at** command can be restricted. For more information, refer to *Section 24.2.5, “Controlling Access to At and Batch”* for details.

24.2.2. Configuring Batch Jobs

To execute a one-time task when the load average is below 0.8, use the **batch** command.

After typing the **batch** command, the at> prompt is displayed. Type the command to execute, press **Enter**, and type **Ctrl**+**D** . Multiple commands can be specified by typing each command followed by the **Enter** key. After typing all the commands, press **Enter** to go to a blank line and type **Ctrl**+**D** . Alternatively, a shell script can be entered at the prompt, pressing **Enter** after each line in the script, and typing **Ctrl**+**D** on a blank line to exit. If a script is entered, the shell used is the shell set in the user's SHELL environment, the user's login shell, or **/bin/sh** (whichever is found first). As soon as the load average is below 0.8, the set of commands or script is executed.

If the set of commands or script tries to display information to standard out, the output is emailed to the user.

Use the command **atq** to view pending jobs. Refer to *Section 24.2.3, “Viewing Pending Jobs”* for more information.

Usage of the **batch** command can be restricted. For more information, refer to *Section 24.2.5, “Controlling Access to At and Batch”* for details.

24.2.3. Viewing Pending Jobs

To view pending **at** and **batch** jobs, use the **atq** command. The **atq** command displays a list of pending jobs, with each job on a line. Each line follows the job number, date, hour, job class, and username format. Users can only view their own jobs. If the root user executes the **atq** command, all jobs for all users are displayed.

24.2.4. Additional Command Line Options

Additional command line options for **at** and **batch** include:

Option	Description
`-f`	Read the commands or shell script from a file instead of specifying them at the prompt.
`-m`	Send email to the user when the job has been completed.
`-v`	Display the time that the job is executed.

Table 24.1. **at** and **batch** Command Line Options

24.2.5. Controlling Access to At and Batch

The **/etc/at.allow** and **/etc/at.deny** files can be used to restrict access to the **at** and **batch** commands. The format of both access control files is one username on each line. Whitespace is not permitted in either file. The **at** daemon (**atd**) does not have to be restarted if the access control files are modified. The access control files are read each time a user tries to execute the **at** or **batch** commands.

The root user can always execute **at** and **batch** commands, regardless of the access control files.

If the file **at.allow** exists, only users listed in it are allowed to use **at** or **batch**, and the **at.deny** file is ignored.

If **at.allow** does not exist, users listed in **at.deny** are not allowed to use **at** or **batch**.

24.2.6. Starting and Stopping the Service

To start the **at** service, use the command **/sbin/service atd start**. To stop the service, use the command **/sbin/service atd stop**. It is recommended that you start the service at boot time. Refer to *Chapter 6, Controlling Access to Services* for details on starting the cron service automatically at boot time.

24.3. Additional Resources

To learn more about configuring automated tasks, refer to the following resources.

24.3.1. Installed Documentation

- **cron** man page — overview of cron.
- **crontab** man pages in sections 1 and 5 — The man page in section 1 contains an overview of the **crontab** file. The man page in section 5 contains the format for the file and some example entries.
- **/usr/share/doc/at-<*version*>/timespec** contains more detailed information about the times that can be specified for cron jobs.
- **at** man page — description of **at** and **batch** and their command line options.

Log Files

Log files are files that contain messages about the system, including the kernel, services, and applications running on it. There are different log files for different information. For example, there is a default system log file, a log file just for security messages, and a log file for cron tasks.

Log files can be very useful when trying to troubleshoot a problem with the system such as trying to load a kernel driver or when looking for unauthorized log in attempts to the system. This chapter discusses where to find log files, how to view log files, and what to look for in log files.

Some log files are controlled by a daemon called **`syslogd`**. A list of log messages maintained by **`syslogd`** can be found in the **`/etc/syslog.conf`** configuration file.

25.1. Locating Log Files

Most log files are located in the **`/var/log/`** directory. Some applications such as **`httpd`** and **`samba`** have a directory within **`/var/log/`** for their log files.

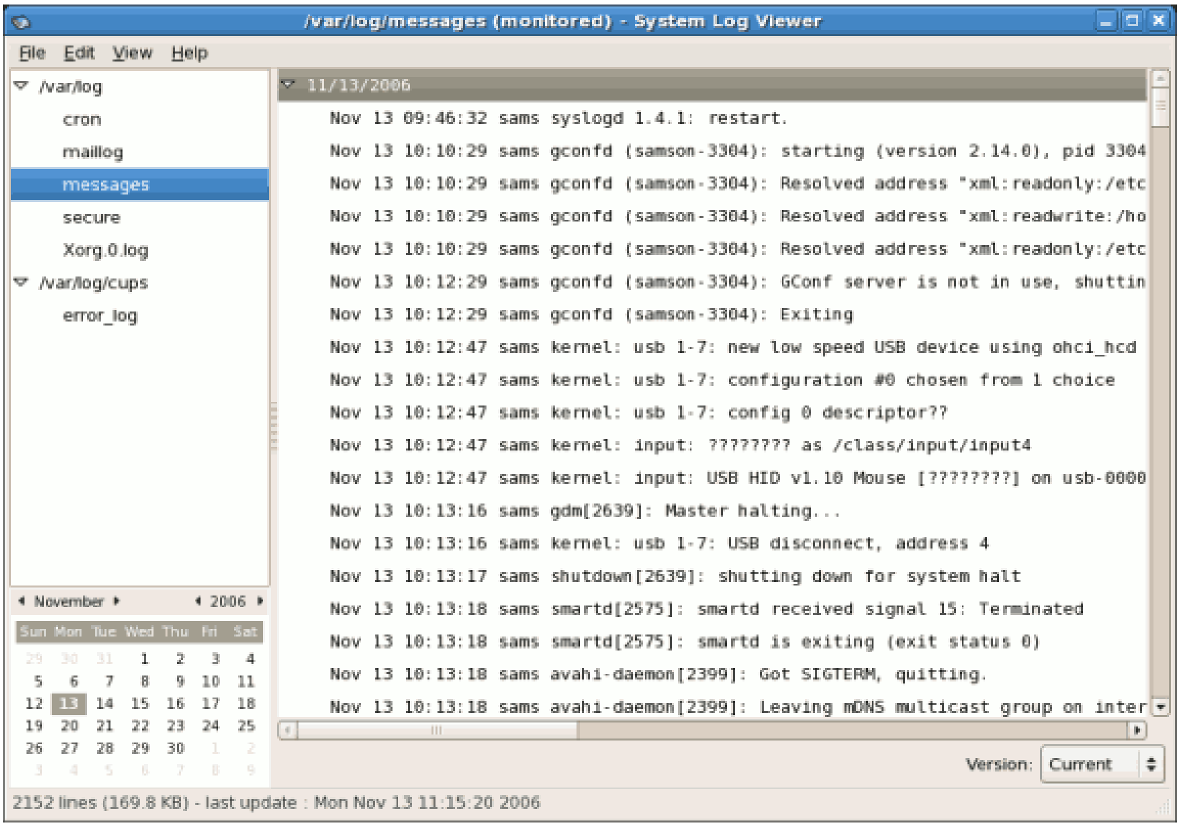
You may notice multiple files in the log file directory with numbers after them. These are created when the log files are rotated. Log files are rotated so their file sizes do not become too large. The **`logrotate`** package contains a cron task that automatically rotates log files according to the **`/etc/logrotate.conf`** configuration file and the configuration files in the **`/etc/logrotate.d/`** directory. By default, it is configured to rotate every week and keep four weeks worth of previous log files.

25.2. Viewing Log Files

Most log files are in plain text format. You can view them with any text editor such as **`Vi`** or **Emacs**. Some log files are readable by all users on the system; however, root privileges are required to read most log files.

To view system log files in an interactive, real-time application, use the **System Log Viewer**. To start the application, go to **Applications** (the main menu on the panel) > **System** > **System Logs**, or type the command **`gnome-system-log`** at a shell prompt.

The application only displays log files that exist; thus, the list might differ from the one shown in *Figure 25.1, "**System Log Viewer**"*.

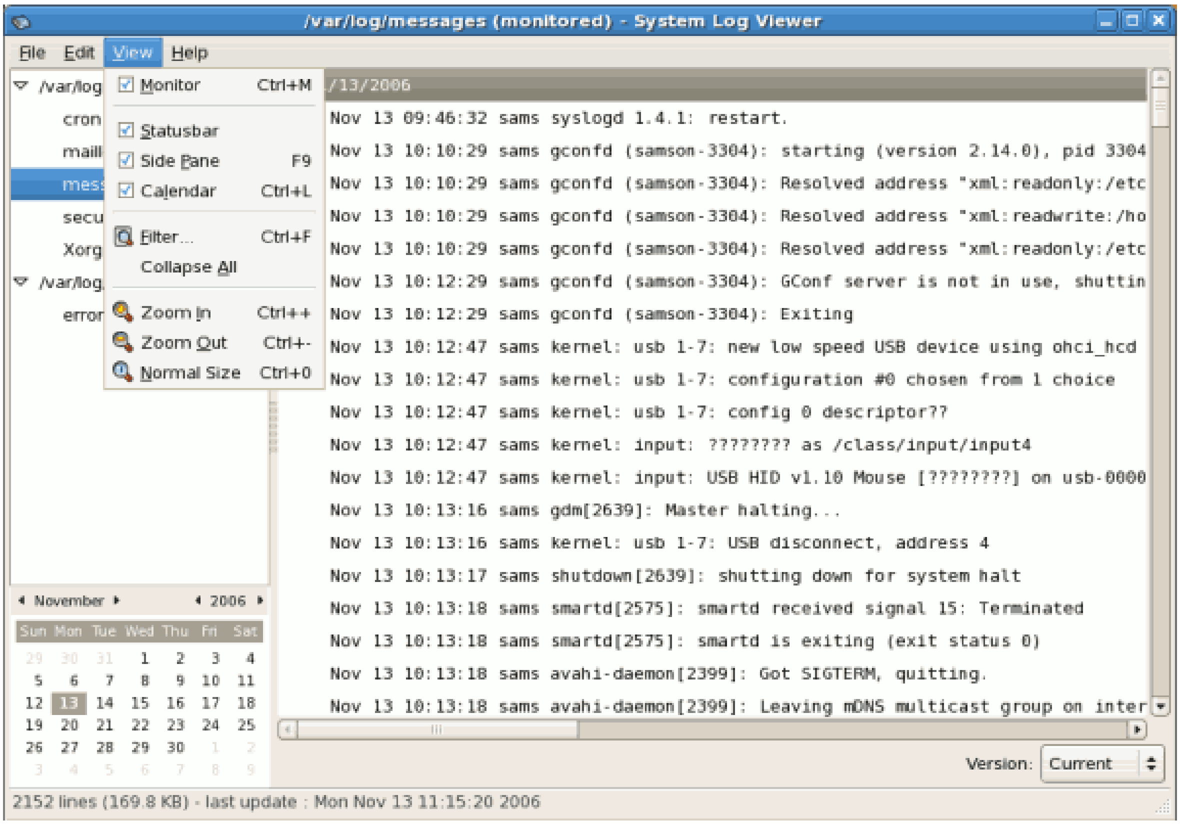


```
Nov 13 09:46:32 sams syslogd 1.4.1: restart.
Nov 13 10:10:29 sams gconfd (samson-3304): starting (version 2.14.0), pid 3304
Nov 13 10:10:29 sams gconfd (samson-3304): Resolved address "xml:readonly:/etc
Nov 13 10:10:29 sams gconfd (samson-3304): Resolved address "xml:readwrite:/ho
Nov 13 10:10:29 sams gconfd (samson-3304): Resolved address "xml:readonly:/etc
Nov 13 10:12:29 sams gconfd (samson-3304): GConf server is not in use, shuttin
Nov 13 10:12:29 sams gconfd (samson-3304): Exiting
Nov 13 10:12:47 sams kernel: usb 1-7: new low speed USB device using ohci_hcd
Nov 13 10:12:47 sams kernel: usb 1-7: configuration #0 chosen from 1 choice
Nov 13 10:12:47 sams kernel: usb 1-7: config 0 descriptor??
Nov 13 10:12:47 sams kernel: input: ???????? as /class/input/input4
Nov 13 10:12:47 sams kernel: input: USB HID v1.10 Mouse [????????] on usb-0000
Nov 13 10:13:16 sams gdm[2639]: Master halting...
Nov 13 10:13:16 sams kernel: usb 1-7: USB disconnect, address 4
Nov 13 10:13:17 sams shutdown[2639]: shutting down for system halt
Nov 13 10:13:18 sams smartd[2575]: smartd received signal 15: Terminated
Nov 13 10:13:18 sams smartd[2575]: smartd is exiting (exit status 0)
Nov 13 10:13:18 sams avahi-daemon[2399]: Got SIGTERM, quitting.
Nov 13 10:13:18 sams avahi-daemon[2399]: Leaving mDNS multicast group on inter
```

2152 lines (169.8 KB) - last update : Mon Nov 13 11:15:20 2006

Figure 25.1. **System Log Viewer**

To filter the contents of the selected log file, click on **View** from the menu and select **Filter** as illustrated below.



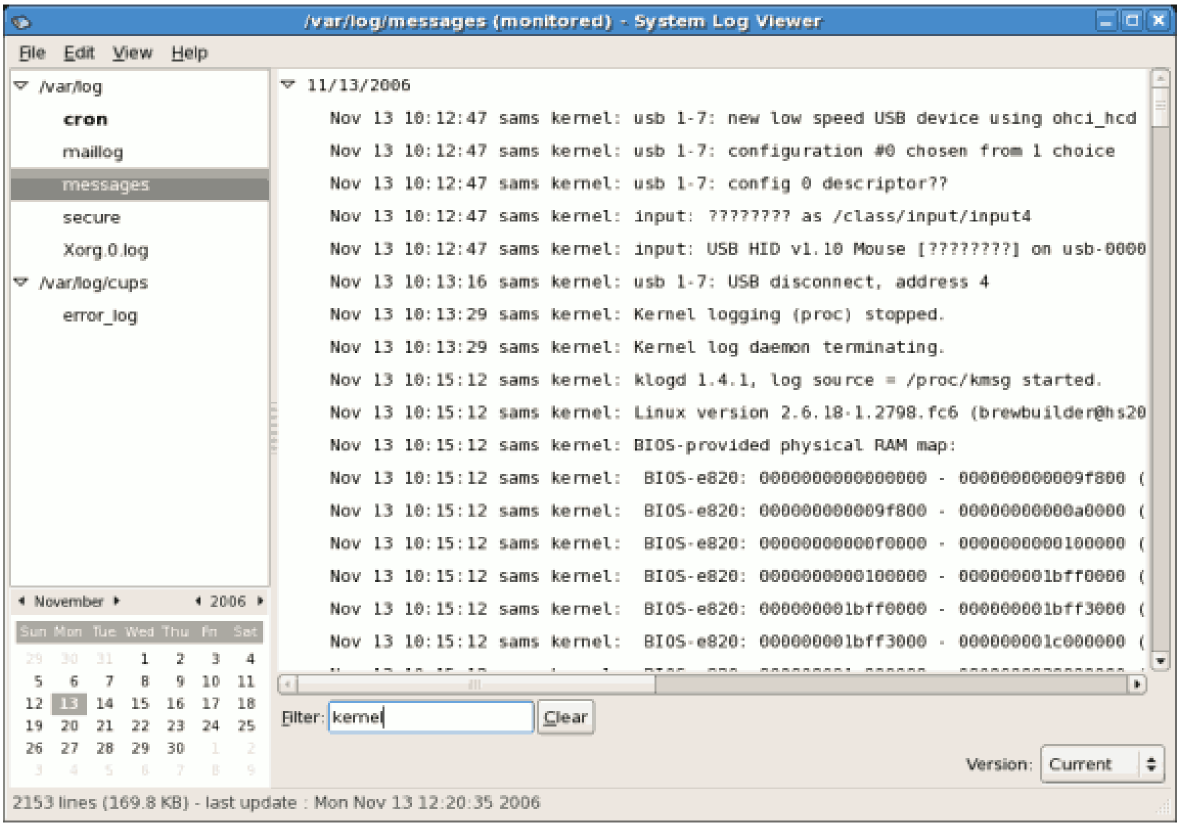
```
Nov 13 09:46:32 sams syslogd 1.4.1: restart.
Nov 13 10:10:29 sams gconfd (samson-3304): starting (version 2.14.0), pid 3304
Nov 13 10:10:29 sams gconfd (samson-3304): Resolved address "xml:readonly:/etc
Nov 13 10:10:29 sams gconfd (samson-3304): Resolved address "xml:readwrite:/ho
Nov 13 10:10:29 sams gconfd (samson-3304): Resolved address "xml:readonly:/etc
Nov 13 10:12:29 sams gconfd (samson-3304): GConf server is not in use, shuttin
Nov 13 10:12:29 sams gconfd (samson-3304): Exiting
Nov 13 10:12:47 sams kernel: usb 1-7: new low speed USB device using ohci_hcd
Nov 13 10:12:47 sams kernel: usb 1-7: configuration #0 chosen from 1 choice
Nov 13 10:12:47 sams kernel: usb 1-7: config 0 descriptor??
Nov 13 10:12:47 sams kernel: input: ???????? as /class/input/input4
Nov 13 10:12:47 sams kernel: input: USB HID v1.10 Mouse [????????] on usb-0000
Nov 13 10:13:16 sams gdm[2639]: Master halting...
Nov 13 10:13:16 sams kernel: usb 1-7: USB disconnect, address 4
Nov 13 10:13:17 sams shutdown[2639]: shutting down for system halt
Nov 13 10:13:18 sams smartd[2575]: smartd received signal 15: Terminated
Nov 13 10:13:18 sams smartd[2575]: smartd is exiting (exit status 0)
Nov 13 10:13:18 sams avahi-daemon[2399]: Got SIGTERM, quitting.
Nov 13 10:13:18 sams avahi-daemon[2399]: Leaving mDNS multicast group on inter
```

Version: Current

2152 lines (169.8 KB) - last update : Mon Nov 13 11:15:20 2006

Figure 25.2. **System Log Viewer - View Menu**

Selecting the **Filter** menu item will display the **Filter** text field where you can type the keywords you wish to use for your filter. To clear your filter click on the **Clear** button.The figure below illustrates a sample filter.



```
11/13/2006
    Nov 13 10:12:47 sams kernel: usb 1-7: new low speed USB device using ohci_hcd
    Nov 13 10:12:47 sams kernel: usb 1-7: configuration #0 chosen from 1 choice
    Nov 13 10:12:47 sams kernel: usb 1-7: config 0 descriptor??
    Nov 13 10:12:47 sams kernel: input: ???????? as /class/input/input4
    Nov 13 10:12:47 sams kernel: input: USB HID v1.10 Mouse [????????] on usb-0000
    Nov 13 10:13:16 sams kernel: usb 1-7: USB disconnect, address 4
    Nov 13 10:13:29 sams kernel: Kernel logging (proc) stopped.
    Nov 13 10:13:29 sams kernel: Kernel log daemon terminating.
    Nov 13 10:15:12 sams kernel: klogd 1.4.1, log source = /proc/kmsg started.
    Nov 13 10:15:12 sams kernel: Linux version 2.6.18-1.2798.fc6 (brewbuilder@hs20
    Nov 13 10:15:12 sams kernel: BIOS-provided physical RAM map:
    Nov 13 10:15:12 sams kernel:  BIOS-e820: 0000000000000000 - 000000000009f800 (
    Nov 13 10:15:12 sams kernel:  BIOS-e820: 000000000009f800 - 00000000000a0000 (
    Nov 13 10:15:12 sams kernel:  BIOS-e820: 00000000000f0000 - 0000000000100000 (
    Nov 13 10:15:12 sams kernel:  BIOS-e820: 0000000000100000 - 000000001bff0000 (
    Nov 13 10:15:12 sams kernel:  BIOS-e820: 000000001bff0000 - 000000001bff3000 (
    Nov 13 10:15:12 sams kernel:  BIOS-e820: 000000001bff3000 - 000000001c000000 (
```

Filter: kernel

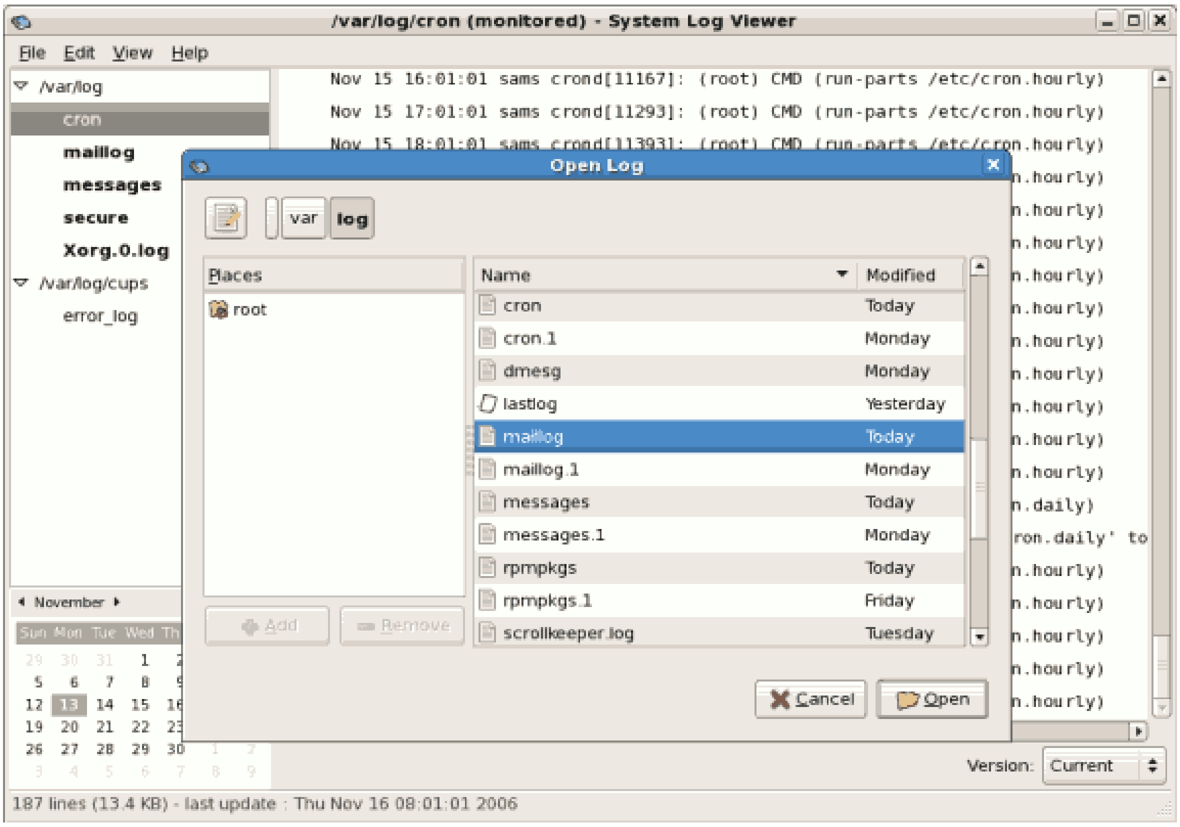
Version: Current

2153 lines (169.8 KB) - last update : Mon Nov 13 12:20:35 2006

Figure 25.3. **System Log Viewer - Filter**

25.3. Adding a Log File

To add a log file you wish to view in the list, select **File** > **Open**. This will display the **Open Log** window where you can select the directory and filename of the log file you wish to view.The figure below illustrates the **Open Log** window.



```
Nov 15 16:01:01 sams crond[11167]: (root) CMD (run-parts /etc/cron.hourly)
Nov 15 17:01:01 sams crond[11293]: (root) CMD (run-parts /etc/cron.hourly)
Nov 15 18:01:01 sams crond[11393]: (root) CMD (run-parts /etc/cron.hourly)
```

Name	Modified
cron	Today
cron.1	Monday
dmesg	Monday
lastlog	Yesterday
maillog	Today
maillog.1	Monday
messages	Today
messages.1	Monday
rpmpkgs	Today
rpmpkgs.1	Friday
scrollkeeper.log	Tuesday

Version: Current

187 lines (13.4 KB) - last update : Thu Nov 16 08:01:01 2006

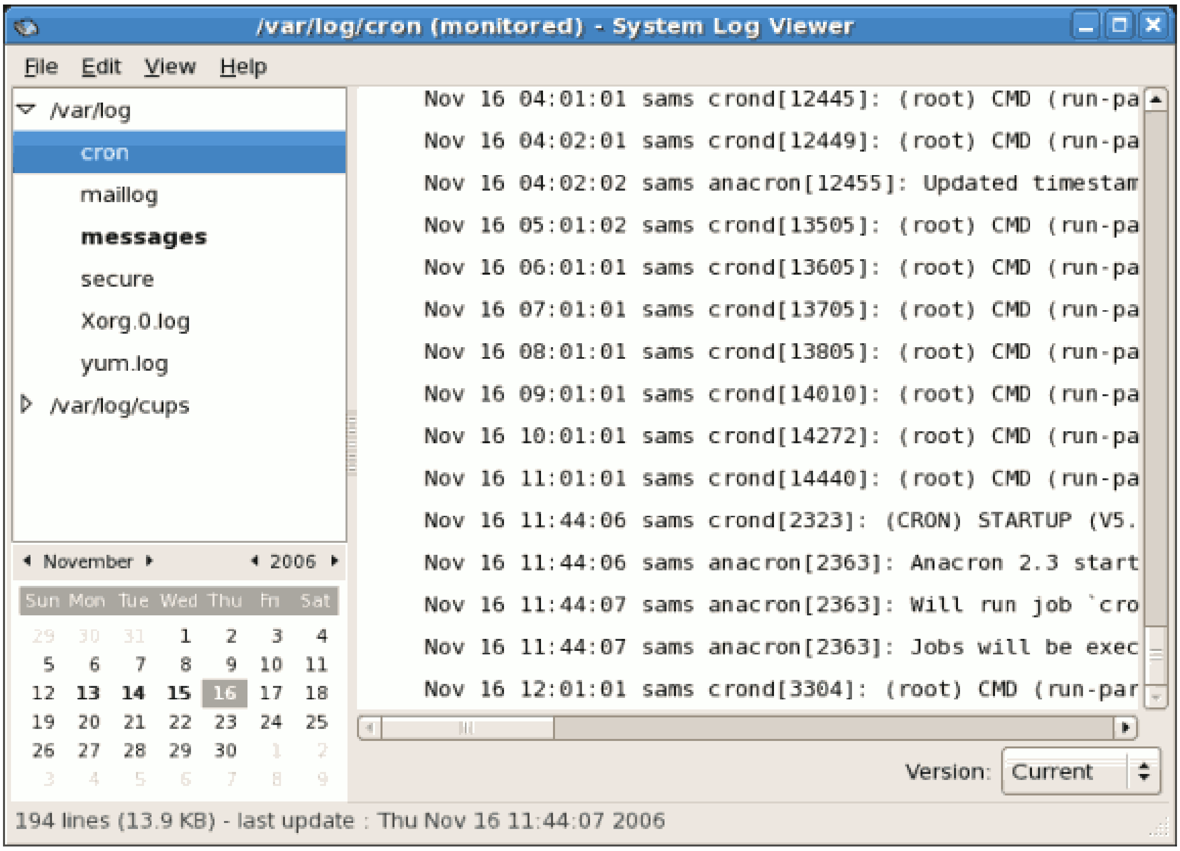
Figure 25.4. Adding a Log File

Click on the **Open** button to open the file. The file is immediately added to the viewing list where you can select it and view the contents.

Please also note that the System Log Viewer also allows you to open zipped logs whose filenames end in ".gz".

25.4. Monitoring Log Files

System Log Viewer monitors all opened logs by default. If a new line is added to a monitored log file, the log name appears in bold in the log list. If the log file is selected or displayed, the new lines appear in bold at the bottom of the log file and after five seconds are displayed in normal format. This is illustrated in the figures below. The figure below illustrates a new alert in the **messages** log file. The log file is listed in bold text.

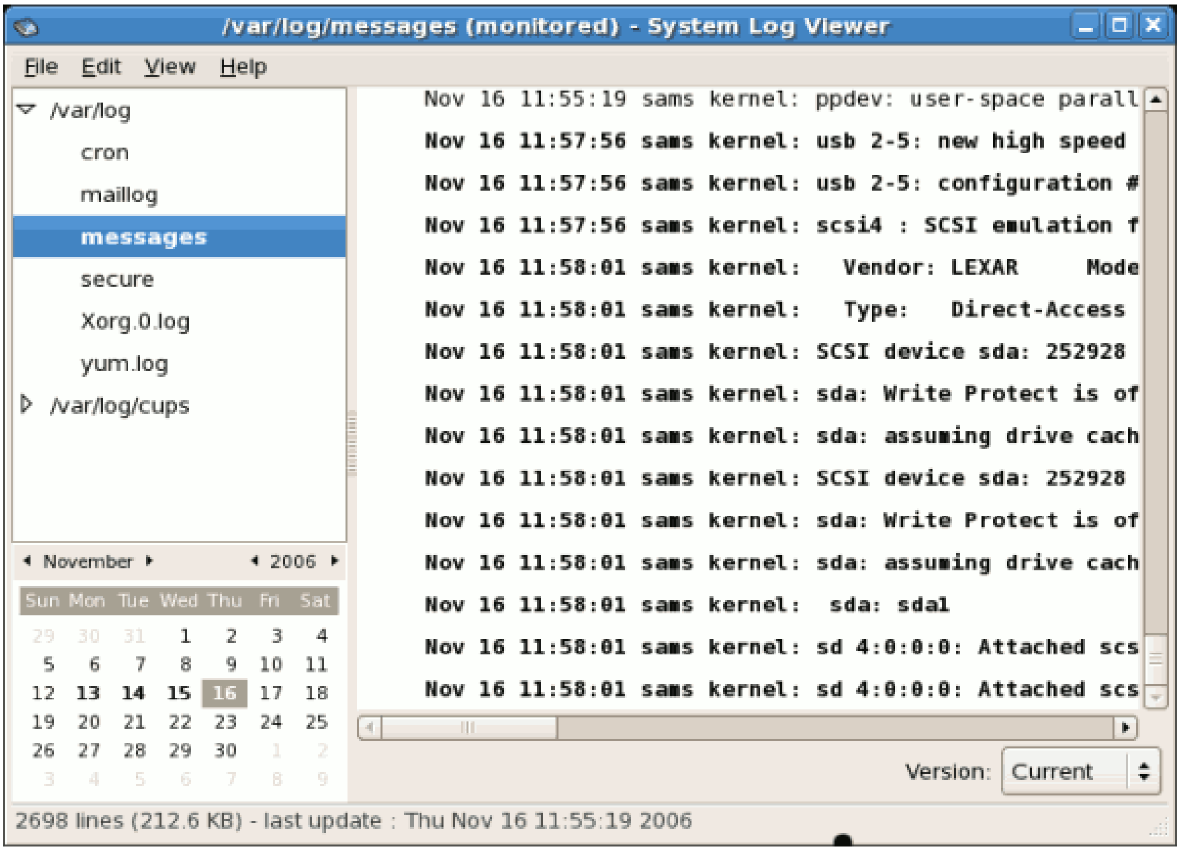


```
Nov 16 04:01:01 sams crond[12445]: (root) CMD (run-pa
Nov 16 04:02:01 sams crond[12449]: (root) CMD (run-pa
Nov 16 04:02:02 sams anacron[12455]: Updated timestam
Nov 16 05:01:02 sams crond[13505]: (root) CMD (run-pa
Nov 16 06:01:01 sams crond[13605]: (root) CMD (run-pa
Nov 16 07:01:01 sams crond[13705]: (root) CMD (run-pa
Nov 16 08:01:01 sams crond[13805]: (root) CMD (run-pa
Nov 16 09:01:01 sams crond[14010]: (root) CMD (run-pa
Nov 16 10:01:01 sams crond[14272]: (root) CMD (run-pa
Nov 16 11:01:01 sams crond[14440]: (root) CMD (run-pa
Nov 16 11:44:06 sams crond[2323]: (CRON) STARTUP (V5.
Nov 16 11:44:06 sams anacron[2363]: Anacron 2.3 start
Nov 16 11:44:07 sams anacron[2363]: Will run job `cro
Nov 16 11:44:07 sams anacron[2363]: Jobs will be exec
Nov 16 12:01:01 sams crond[3304]: (root) CMD (run-par
```

194 lines (13.9 KB) - last update : Thu Nov 16 11:44:07 2006

Figure 25.5. Log File Alert

Clicking on the **messages** log file displays the logs in the file with the new lines in bold as illustrated below.

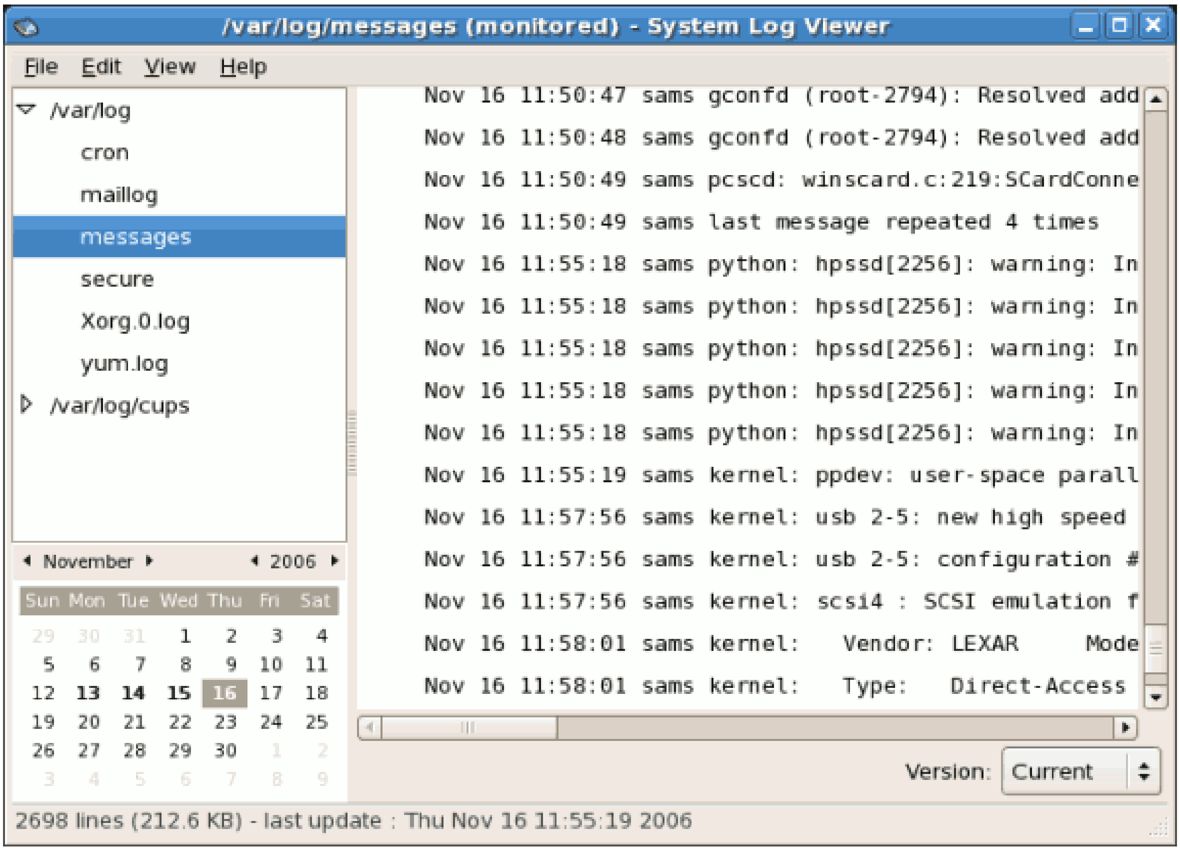


```
Nov 16 11:55:19 sams kernel: ppdev: user-space parall
Nov 16 11:57:56 sams kernel: usb 2-5: new high speed
Nov 16 11:57:56 sams kernel: usb 2-5: configuration #
Nov 16 11:57:56 sams kernel: scsi4 : SCSI emulation f
Nov 16 11:58:01 sams kernel:   Vendor: LEXAR     Mode
Nov 16 11:58:01 sams kernel:   Type:   Direct-Access
Nov 16 11:58:01 sams kernel: SCSI device sda: 252928
Nov 16 11:58:01 sams kernel: sda: Write Protect is of
Nov 16 11:58:01 sams kernel: sda: assuming drive cach
Nov 16 11:58:01 sams kernel: SCSI device sda: 252928
Nov 16 11:58:01 sams kernel: sda: Write Protect is of
Nov 16 11:58:01 sams kernel: sda: assuming drive cach
Nov 16 11:58:01 sams kernel:  sda: sda1
Nov 16 11:58:01 sams kernel: sd 4:0:0:0: Attached scs
Nov 16 11:58:01 sams kernel: sd 4:0:0:0: Attached scs
```

2698 lines (212.6 KB) - last update : Thu Nov 16 11:55:19 2006

Figure 25.6. Log file contents

The new lines are displayed in bold for five seconds after which they are displayed in normal font.



```
Nov 16 11:50:47 sams gconfd (root-2794): Resolved add
Nov 16 11:50:48 sams gconfd (root-2794): Resolved add
Nov 16 11:50:49 sams pcscd: winscard.c:219:SCardConne
Nov 16 11:50:49 sams last message repeated 4 times
Nov 16 11:55:18 sams python: hpssd[2256]: warning: In
Nov 16 11:55:18 sams python: hpssd[2256]: warning: In
Nov 16 11:55:18 sams python: hpssd[2256]: warning: In
Nov 16 11:55:18 sams python: hpssd[2256]: warning: In
Nov 16 11:55:18 sams python: hpssd[2256]: warning: In
Nov 16 11:55:19 sams kernel: ppdev: user-space parall
Nov 16 11:57:56 sams kernel: usb 2-5: new high speed
Nov 16 11:57:56 sams kernel: usb 2-5: configuration #
Nov 16 11:57:56 sams kernel: scsi4 : SCSI emulation f
Nov 16 11:58:01 sams kernel:   Vendor: LEXAR      Mode
Nov 16 11:58:01 sams kernel:   Type:    Direct-Access
```

2698 lines (212.6 KB) - last update : Thu Nov 16 11:55:19 2006

Figure 25.7. Log file contents after five seconds

Part IV. System Monitoring

System administrators also monitor system performance. Fedora contains tools to assist administrators with these tasks.

Chapter 26.

Gathering System Information

Before you learn how to configure your system, you should learn how to gather essential system information. For example, you should know how to find the amount of free memory, the amount of available hard drive space, how your hard drive is partitioned, and what processes are running. This chapter discusses how to retrieve this type of information from your Fedora system using simple commands and a few simple programs.

26.1. System Processes

The **ps ax** command displays a list of current system processes, including processes owned by other users. To display the owner alongside each process, use the **ps aux** command. This list is a static list; in other words, it is a snapshot of what was running when you invoked the command. If you want a constantly updated list of running processes, use **top** as described below.

The **ps** output can be long. To prevent it from scrolling off the screen, you can pipe it through less:

```
ps aux | less
```

You can use the **ps** command in combination with the **grep** command to see if a process is running. For example, to determine if **Emacs** is running, use the following command:

```
ps ax | grep emacs
```

The **top** command displays currently running processes and important information about them including their memory and CPU usage. The list is both real-time and interactive. An example of output from the **top** command is provided as follows:

```
 top - 15:02:46 up 35 min, 4 users, load average: 0.17, 0.65, 1.00 Tasks:
 110 total, 1 running, 107 sleeping, 0 stopped, 2 zombie Cpu(s): 41.1% us,
 2.0% sy, 0.0% ni, 56.6% id, 0.0% wa, 0.3% hi, 0.0% si Mem: 775024k total,
 772028k used, 2996k free, 68468k buffers Swap: 1048568k total, 176k used,
 1048392k free, 441172k cached PID USER PR NI VIRT RES SHR S %CPU %MEM TIME
+ COMMAND 4624 root 15 0 40192 18m 7228 S 28.4 2.4 1:23.21 X 4926 mhideo
 15 0 55564 33m 9784 S 13.5 4.4 0:25.96 gnome-terminal 6475 mhideo 16 0
 3612 968 760 R 0.7 0.1 0:00.11 top 4920 mhideo 15 0 20872 10m 7808 S 0.3
 1.4 0:01.61 wnck-applet 1 root 16 0 1732 548 472 S 0.0 0.1 0:00.23 init
 2 root 34 19 0 0 0 S 0.0 0.0 0:00.00 ksoftirqd/0 3 root 5 -10 0 0 0 S 0.0
 0.0 0:00.03 events/0 4 root 6 -10 0 0 0 S 0.0 0.0 0:00.02 khelper 5 root 5
 -10 0 0 0 S 0.0 0.0 0:00.00 kacpid 29 root 5 -10 0 0 0 S 0.0 0.0 0:00.00
 kblockd/0 47 root 16 0 0 0 0 S 0.0 0.0 0:01.74 pdflush 50 root 11 -10 0
 0 0 S 0.0 0.0 0:00.00 aio/0 30 root 15 0 0 0 0 S 0.0 0.0 0:00.05 khubd 49
 root 16 0 0 0 0 S 0.0 0.0 0:01.44 kswapd0
```

To exit **top**, press the **q** key.

*Table 26.1, “Interactive **top** commands”* contains useful interactive commands that you can use with **top**. For more information, refer to the **top**(1) manual page.

Command	Description
`Space`	Immediately refresh the display
`h`	Display a help screen
`k`	Kill a process. You are prompted for the process ID and the signal to send to it.
`n`	Change the number of processes displayed. You are prompted to enter the number.
`u`	Sort by user.
`M`	Sort by memory usage.
`P`	Sort by CPU usage.

Table 26.1. Interactive `top` commands

If you prefer a graphical interface for `top`, you can use the **GNOME System Monitor**. To start it from the desktop, select **System** > **Administration** > **System Monitor** or type `gnome-system-monitor` at a shell prompt (such as an XTerm). Select the **Process Listing** tab.

The **GNOME System Monitor** allows you to search for a process in the list of running processes. Using the Gnome System Monitor, you can also view all processes, your processes, or active processes.

The **Edit** menu item allows you to:

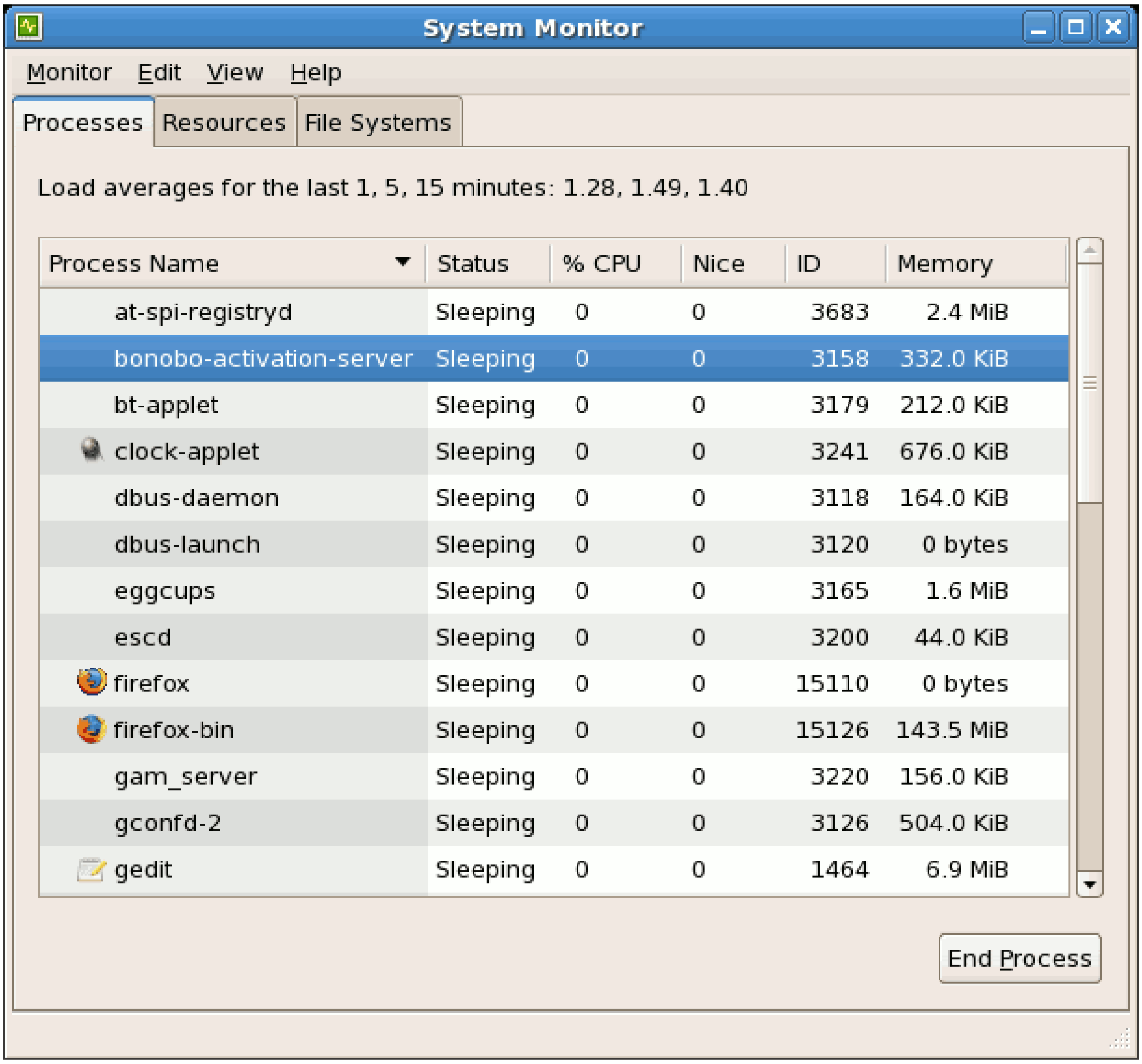
- Stop a process.
- Continue or start a process.
- End a processes.
- Kill a process.
- Change the priority of a selected process.
- Edit the System Monitor preferences. These include changing the interval seconds to refresh the list and selecting process fields to display in the System Monitor window.

The **View** menu item allows you to:

- View only active processes.
- View all processes.
- View my processes.
- View process dependencies.
- Hide a process.
- View hidden processes.
- View memory maps.
- View the files opened by the selected process.

To stop a process, select it and click **End Process**. Alternatively you can also stop a process by selecting it, clicking **Edit** on your menu and selecting **Stop Process**.

To sort the information by a specific column, click on the name of the column. This sorts the information by the selected column in ascending order. Click on the name of the column again to toggle the sort between ascending and descending order.



Load averages for the last 1, 5, 15 minutes: 1.28, 1.49, 1.40

Process Name	Status	% CPU	Nice	ID	Memory
at-spi-registryd	Sleeping	0	0	3683	2.4 MiB
bonobo-activation-server	Sleeping	0	0	3158	332.0 KiB
bt-applet	Sleeping	0	0	3179	212.0 KiB
clock-applet	Sleeping	0	0	3241	676.0 KiB
dbus-daemon	Sleeping	0	0	3118	164.0 KiB
dbus-launch	Sleeping	0	0	3120	0 bytes
eggcups	Sleeping	0	0	3165	1.6 MiB
escd	Sleeping	0	0	3200	44.0 KiB
firefox	Sleeping	0	0	15110	0 bytes
firefox-bin	Sleeping	0	0	15126	143.5 MiB
gam_server	Sleeping	0	0	3220	156.0 KiB
gconfd-2	Sleeping	0	0	3126	504.0 KiB
gedit	Sleeping	0	0	1464	6.9 MiB

Figure 26.1. **GNOME System Monitor**

26.2. Memory Usage

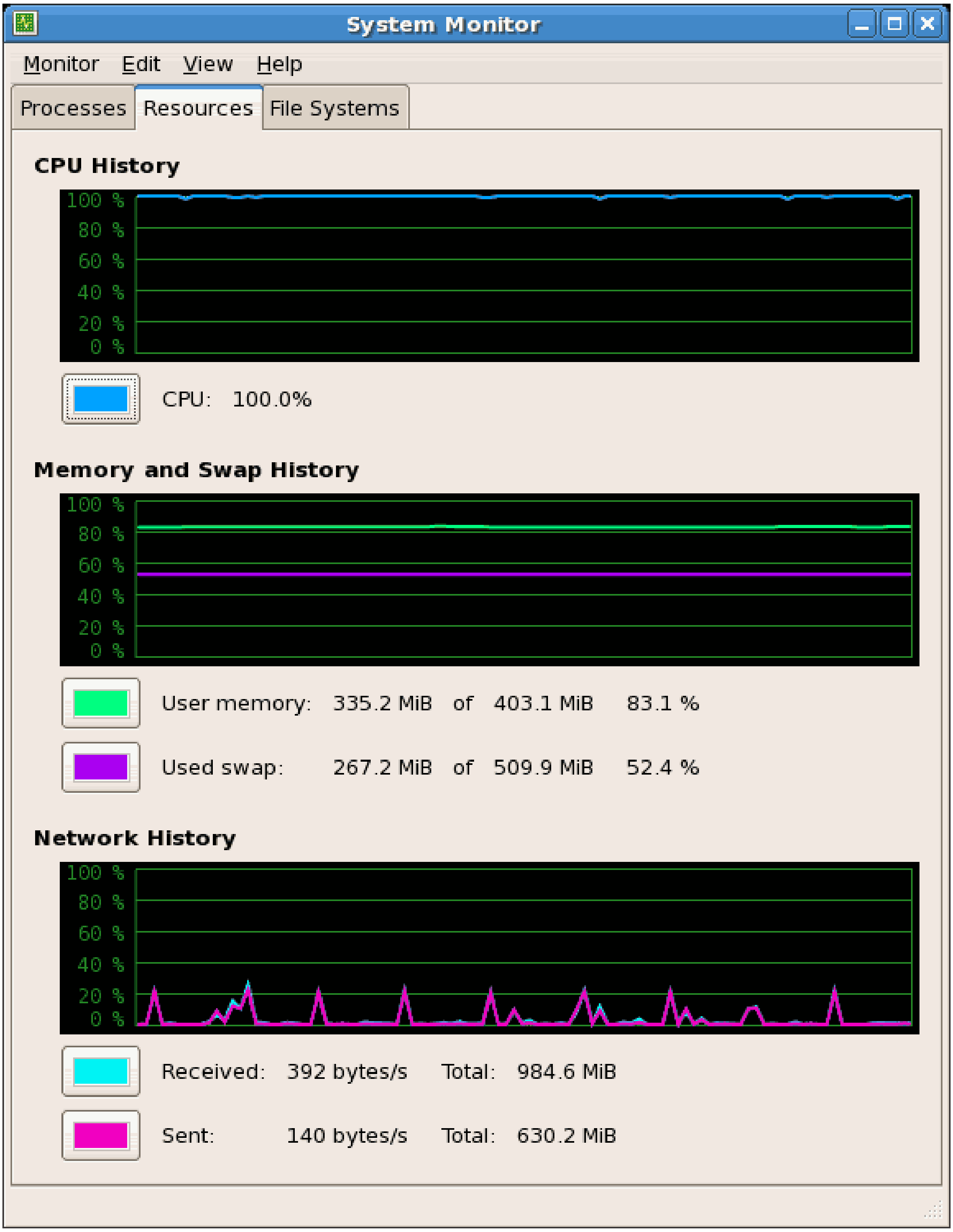
The **`free`** command displays the total amount of physical memory and swap space for the system as well as the amount of memory that is used, free, shared, in kernel buffers, and cached.

```
total used free shared buffers cached Mem: 645712 549720 95992 0 176248
224452 -/+ buffers/cache: 149020 496692 Swap: 1310712 0 1310712
```

The command **`free -m`** shows the same information in megabytes, which are easier to read.

```
total used free shared buffers cached Mem: 630 536 93 0 172 219 -/+
buffers/cache: 145 485 Swap: 1279 0 1279
```

If you prefer a graphical interface for `free`, you can use the **GNOME System Monitor**. To start it from the desktop, go to **System** > **Administration** > **System Monitor** or type **`gnome-system-monitor`** at a shell prompt (such as an XTerm). Click on the **Resources** tab.



CPU: 100.0%

User memory: 335.2 MiB of 403.1 MiB 83.1 %

Used swap: 267.2 MiB of 509.9 MiB 52.4 %

Received: 392 bytes/s Total: 984.6 MiB

Sent: 140 bytes/s Total: 630.2 MiB

Figure 26.2. **GNOME System Monitor - Resources tab**

26.3. File Systems

The **`df`** command reports the system's disk space usage. If you type the command **`df`** at a shell prompt, the output looks similar to the following:

```
 Filesystem 1K-blocks Used Available Use% Mounted on /dev/mapper/
VolGroup00-LogVol00 11675568 6272120 4810348 57% / /dev/sda1 100691 9281
 86211 10% /boot none 322856 0 322856 0% /dev/shm
```

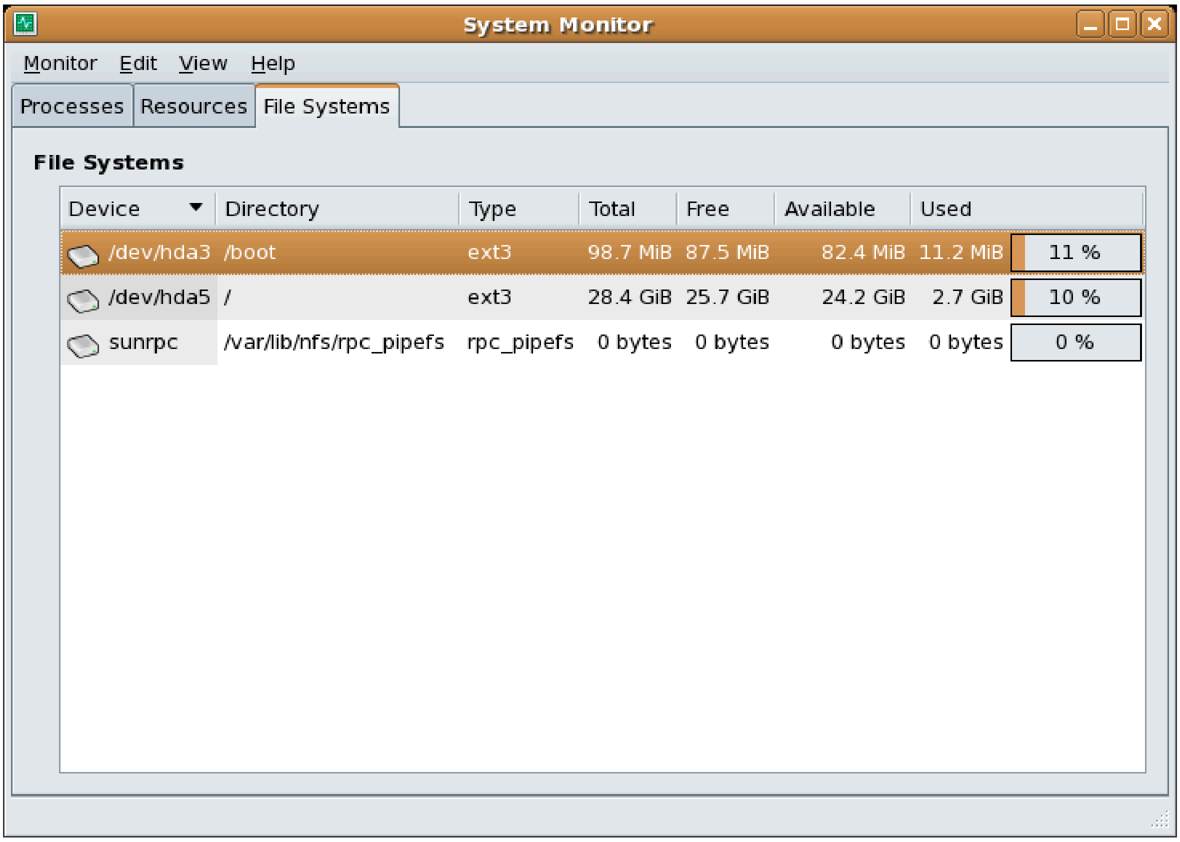
By default, this utility shows the partition size in 1 kilobyte blocks and the amount of used and available disk space in kilobytes. To view the information in megabytes and gigabytes, use the command **`df -h`**. The **`-h`** argument stands for human-readable format. The output looks similar to the following:

```
 Filesystem Size Used Avail Use% Mounted on /dev/mapper/VolGroup00-LogVol00
 12G 6.0G 4.6G 57% / /dev/sda1 99M 9.1M 85M 10% /boot none 316M 0 316M 0% /
dev/shm
```

In the list of mounted partitions, there is an entry for **`/dev/shm`**. This entry represents the system's virtual memory file system.

The **`du`** command displays the estimated amount of space being used by files in a directory. If you type **`du`** at a shell prompt, the disk usage for each of the subdirectories is displayed in a list. The grand total for the current directory and subdirectories are also shown as the last line in the list. If you do not want to see the totals for all the subdirectories, use the command **`du -hs`** to see only the grand total for the directory in human-readable format. Use the **`du --help`** command to see more options.

To view the system's partitions and disk space usage in a graphical format, use the **Gnome System Monitor** by clicking on **System** > **Administration** > **System Monitor** or type **`gnome-system-monitor`** at a shell prompt (such as an XTerm). Select the File Systems tab to view the system's partitions. The figure below illustrates the File Systems tab.



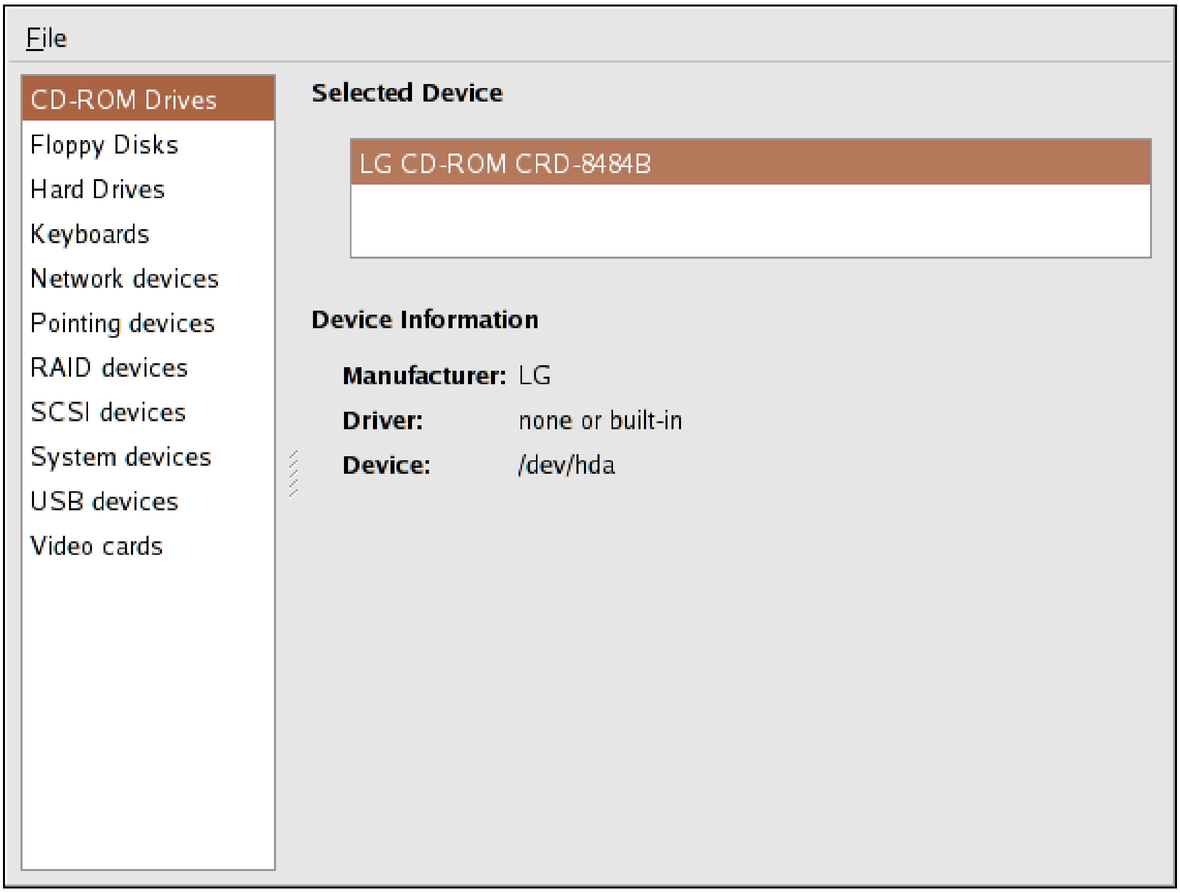
File Systems

Device	Directory	Type	Total	Free	Available	Used	
/dev/hda3	/boot	ext3	98.7 MiB	87.5 MiB	82.4 MiB	11.2 MiB	11 %
/dev/hda5	/	ext3	28.4 GiB	25.7 GiB	24.2 GiB	2.7 GiB	10 %
sunrpc	/var/lib/nfs/rpc_pipefs	rpc_pipefs	0 bytes	0 bytes	0 bytes	0 bytes	0 %

Figure 26.3. **GNOME System Monitor - File Systems**

26.4. Hardware

If you are having trouble configuring your hardware or just want to know what hardware is in your system, you can use the **Hardware Browser** application to display the hardware that can be probed. To start the program from the desktop, select **System** (the main menu on the panel) > **Administration** > **Hardware** or type **`hwbrowser`** at a shell prompt. As shown in *Figure 26.4, "**Hardware Browser**"*, it displays your CD-ROM devices, diskette drives, hard drives and their partitions, network devices, pointing devices, system devices, and video cards. Click on the category name in the left menu, and the information is displayed.



LG CD-ROM CRD-8484B

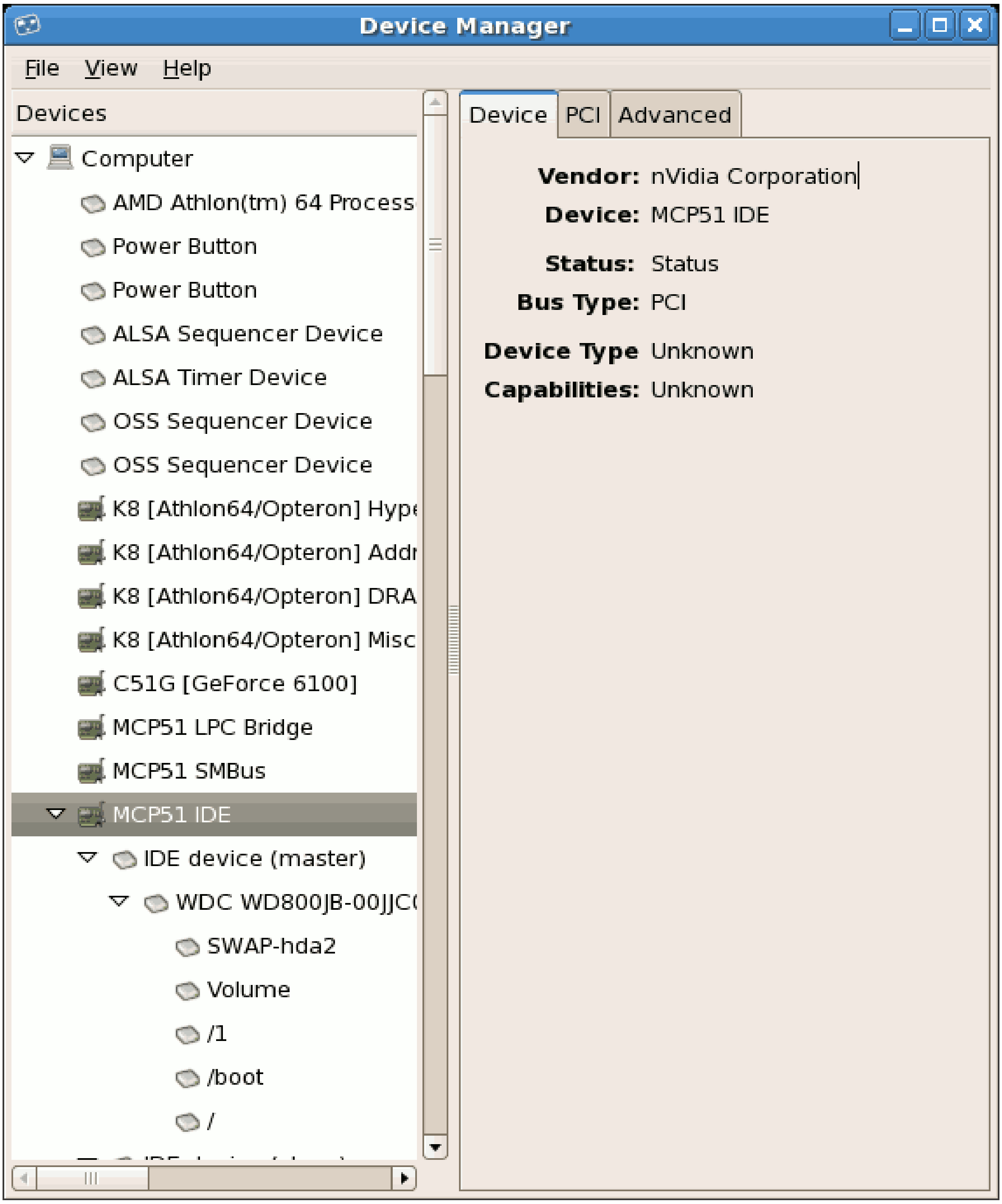
Manufacturer: LG

Driver: none or built-in

Device: /dev/hda

Figure 26.4. **Hardware Browser**

The **Device Manager** application can also be used to display your system hardware. This application can be started by selecting **System** (the main menu on the panel) > **Administration** > **Hardware** like the **Hardware Browser**. To start the application from a terminal, type **`hal-device-manager`**. Depending on your installation preferences, the graphical menu above may start this application or the **Hardware Browser** when clicked. The figure below illustrates the **Device Manager** window.



Vendor: nVidia Corporation
Device: MCP51 IDE
Status: Status
Bus Type: PCI
Device Type Unknown
Capabilities: Unknown

Figure 26.5. **Device Manager**

You can also use the **lspci** command to list all PCI devices. Use the command **lspci -v** for more verbose information or **lspci -vv** for very verbose output.

For example, **lspci** can be used to determine the manufacturer, model, and memory size of a system's video card:

```
00:00.0 Host bridge: ServerWorks CNB20LE Host Bridge (rev 06) 00:00.1 Host
bridge: ServerWorks CNB20LE Host Bridge (rev 06) 00:01.0 VGA compatible
controller: S3 Inc. Savage 4 (rev 04) 00:02.0 Ethernet controller:
```

```
Intel Corp. 82557/8/9 [Ethernet Pro 100] (rev 08) 00:0f.0 ISA bridge:
ServerWorks OSB4 South Bridge (rev 50) 00:0f.1 IDE interface: ServerWorks
OSB4 IDE Controller 00:0f.2 USB Controller: ServerWorks OSB4/CSB5 OHCI
USB Controller (rev 04) 01:03.0 SCSI storage controller: Adaptec AIC-7892P
U160/m (rev 02) 01:05.0 RAID bus controller: IBM ServeRAID Controller
```

The **`lspci`** is also useful to determine the network card in your system if you do not know the manufacturer or model number.

26.5. Additional Resources

To learn more about gathering system information, refer to the following resources.

26.5.1. Installed Documentation

- **`ps --help`** — Displays a list of options that can be used with **`ps`**.
- **`top`** manual page — Type **`man top`** to learn more about **`top`** and its many options.
- **`free`** manual page — type **`man free`** to learn more about **`free`** and its many options.
- **`df`** manual page — Type **`man df`** to learn more about the **`df`** command and its many options.
- **`du`** manual page — Type **`man du`** to learn more about the **`du`** command and its many options.
- **`lspci`** manual page — Type **`man lspci`** to learn more about the **`lspci`** command and its many options.
- **`/proc/`** directory — The contents of the **`/proc/`** directory can also be used to gather more detailed system information.

OProfile

OProfile is a low overhead, system-wide performance monitoring tool. It uses the performance monitoring hardware on the processor to retrieve information about the kernel and executables on the system, such as when memory is referenced, the number of L2 cache requests, and the number of hardware interrupts received. On a Fedora system, the **`oprofile`** RPM package must be installed to use this tool.

Many processors include dedicated performance monitoring hardware. This hardware makes it possible to detect when certain events happen (such as the requested data not being in cache). The hardware normally takes the form of one or more *counters* that are incremented each time an event takes place. When the counter value, essentially rolls over, an interrupt is generated, making it possible to control the amount of detail (and therefore, overhead) produced by performance monitoring.

OProfile uses this hardware (or a timer-based substitute in cases where performance monitoring hardware is not present) to collect *samples* of performance-related data each time a counter generates an interrupt. These samples are periodically written out to disk; later, the data contained in these samples can then be used to generate reports on system-level and application-level performance.

OProfile is a useful tool, but be aware of some limitations when using it:

- *Use of shared libraries* — Samples for code in shared libraries are not attributed to the particular application unless the `--separate=library` option is used.
- *Performance monitoring samples are inexact* — When a performance monitoring register triggers a sample, the interrupt handling is not precise like a divide by zero exception. Due to the out-of-order execution of instructions by the processor, the sample may be recorded on a nearby instruction.
- ***`opreport`*** *does not associate samples for inline functions' properly* — **`opreport`** uses a simple address range mechanism to determine which function an address is in. Inline function samples are not attributed to the inline function but rather to the function the inline function was inserted into.
- *OProfile accumulates data from multiple runs* — OProfile is a system-wide profiler and expects processes to start up and shut down multiple times. Thus, samples from multiple runs accumulate. Use the command **`opcontrol --reset`** to clear out the samples from previous runs.
- *Non-CPU-limited performance problems* — OProfile is oriented to finding problems with CPU-limited processes. OProfile does not identify processes that are asleep because they are waiting on locks or for some other event to occur (for example an I/O device to finish an operation).

27.1. Overview of Tools

Table 27.1, “OProfile Commands” provides a brief overview of the tools provided with the **`oprofile`** package.

Command	Description
`ophelp`	Displays available events for the system's processor along with a brief description of each.

Command	Description
`opimport`	Converts sample database files from a foreign binary format to the native format for the system. Only use this option when analyzing a sample database from a different architecture.
`opannotate`	Creates annotated source for an executable if the application was compiled with debugging symbols. Refer to *Section 27.5.4, “Using **`opannotate`**”* for details.
`opcontrol`	Configures what data is collected. Refer to *Section 27.2, “Configuring OProfile”* for details.
`opreport`	Retrieves profile data. Refer to *Section 27.5.1, “Using **`opreport`**”* for details.
`oprofiled`	Runs as a daemon to periodically write sample data to disk.

Table 27.1. OProfile Commands

27.2. Configuring OProfile

Before OProfile can be run, it must be configured. At a minimum, selecting to monitor the kernel (or selecting not to monitor the kernel) is required. The following sections describe how to use the **`opcontrol`** utility to configure OProfile. As the **`opcontrol`** commands are executed, the setup options are saved to the **`/root/.oprofile/daemonrc`** file.

27.2.1. Specifying the Kernel

First, configure whether OProfile should monitor the kernel. This is the only configuration option that is required before starting OProfile. All others are optional.

To monitor the kernel, execute the following command as root:

```
opcontrol --setup --vmlinux=/usr/lib/debug/lib/modules/`uname -r`/vmlinux
```

Note

The **`debuginfo`** package must be installed (which contains the uncompressed kernel) in order to monitor the kernel.

To configure OProfile not to monitor the kernel, execute the following command as root:

```
opcontrol --setup --no-vmlinux
```

This command also loads the `oprofile` kernel module, if it is not already loaded, and creates the **`/dev/oprofile/`** directory, if it does not already exist. Refer to *Section 27.6, “Understanding **`/dev/oprofile/`**”* for details about this directory.

Note

Even if OProfile is configured not to profile the kernel, the SMP kernel still must be running so that the **`oprofile`** module can be loaded from it.

Setting whether samples should be collected within the kernel only changes what data is collected, not how or where the collected data is stored. To generate different sample files for the kernel and application libraries, refer to *Section 27.2.3, “Separating Kernel and User-space Profiles”*.

27.2.2. Setting Events to Monitor

Most processors contain *counters*, which are used by OProfile to monitor specific events. As shown in *Table 27.2, “OProfile Processors and Counters”*, the number of counters available depends on the processor.

Processor	cpu_type	Number of Counters
Pentium Pro	i386/ppro	2
Pentium II	i386/pii	2
Pentium III	i386/piii	2
Pentium 4 (non-hyper-threaded)	i386/p4	8
Pentium 4 (hyper-threaded)	i386/p4-ht	4
Athlon	i386/athlon	4
AMD64	x86-64/hammer	4
Itanium	ia64/itanium	4
Itanium 2	ia64/itanium2	4
TIMER_INT	timer	1
IBM eServer iSeries and pSeries	timer	1
	ppc64/power4	8
	ppc64/power5	6
	ppc64/970	8
IBM eServer S/390 and S/390x	timer	1
IBM eServer zSeries	timer	1

Table 27.2. OProfile Processors and Counters

Use *Table 27.2, “OProfile Processors and Counters”* to verify that the correct processor type was detected and to determine the number of events that can be monitored simultaneously. `timer` is used as the processor type if the processor does not have supported performance monitoring hardware.

If `timer` is used, events cannot be set for any processor because the hardware does not have support for hardware performance counters. Instead, the timer interrupt is used for profiling.

If `timer` is not used as the processor type, the events monitored can be changed, and counter 0 for the processor is set to a time-based event by default. If more than one counter exists on the processor, the counters other than counter 0 are not set to an event by default. The default events monitored are shown in *Table 27.3, “Default Events”*.

Processor	Default Event for Counter	Description
Pentium Pro, Pentium II, Pentium III, Athlon, AMD64	CPU_CLK_UNHALTED	The processor's clock is not halted
Pentium 4 (HT and non-HT)	GLOBAL_POWER_EVENTS	The time during which the processor is not stopped
Itanium 2	CPU_CYCLES	CPU Cycles
TIMER_INT	(none)	Sample for each timer interrupt
ppc64/power4	CYCLES	Processor Cycles
ppc64/power5	CYCLES	Processor Cycles
ppc64/970	CYCLES	Processor Cycles

Table 27.3. Default Events

The number of events that can be monitored at one time is determined by the number of counters for the processor. However, it is not a one-to-one correlation; on some processors, certain events must be mapped to specific counters. To determine the number of counters available, execute the following command:

```
ls -d /dev/oprofile/[0-9]*
```

The events available vary depending on the processor type. To determine the events available for profiling, execute the following command as root (the list is specific to the system's processor type):

```
ophelp
```

The events for each counter can be configured via the command line or with a graphical interface. For more information on the graphical interface, refer to *Section 27.8, “Graphical Interface”*. If the counter cannot be set to a specific event, an error message is displayed.

To set the event for each configurable counter via the command line, use **opcontrol**:

```
opcontrol --event=<event-name>:<sample-rate>
```

Replace <*event-name*> with the exact name of the event from **ophelp**, and replace <*sample-rate*> with the number of events between samples.

27.2.2.1. Sampling Rate

By default, a time-based event set is selected. It creates a sample every 100,000 clock cycles per processor. If the timer interrupt is used, the timer is set to whatever the jiffy rate is and is not user-settable. If the `cpu_type` is not `timer`, each event can have a *sampling rate* set for it. The sampling rate is the number of events between each sample snapshot.

When setting the event for the counter, a sample rate can also be specified:

```
opcontrol --event=<event-name>:<sample-rate>
```

Replace *<sample-rate>* with the number of events to wait before sampling again. The smaller the count, the more frequent the samples. For events that do not happen frequently, a lower count may be needed to capture the event instances.

> **Caution**
>
> Be extremely careful when setting sampling rates. Sampling too frequently can overload the system, causing the system to appear as if it is frozen or causing the system to actually freeze.

27.2.2.2. Unit Masks

Some user performance monitoring events may also require unit masks to further define the event.

Unit masks for each event are listed with the **ophelp** command. The values for each unit mask are listed in hexadecimal format. To specify more than one unit mask, the hexadecimal values must be combined using a bitwise *or* operation.

```
opcontrol --event=<event-name>:<sample-rate>:<unit-mask>
```

27.2.3. Separating Kernel and User-space Profiles

By default, kernel mode and user mode information is gathered for each event. To configure OProfile to ignore events in kernel mode for a specific counter, execute the following command:

```
opcontrol --event=<event-name>:<sample-rate>:<unit-mask>:0
```

Execute the following command to start profiling kernel mode for the counter again:

```
opcontrol --event=<event-name>:<sample-rate>:<unit-mask>:1
```

To configure OProfile to ignore events in user mode for a specific counter, execute the following command:

```
opcontrol --event=<event-name>:<sample-rate>:<unit-mask>:<kernel>:0
```

Execute the following command to start profiling user mode for the counter again:

```
opcontrol --event=<event-name>:<sample-rate>:<unit-mask>:<kernel>:1
```

When the OProfile daemon writes the profile data to sample files, it can separate the kernel and library profile data into separate sample files. To configure how the daemon writes to sample files, execute the following command as root:

```
opcontrol --separate=<choice>
```

<*choice*> can be one of the following:

- `none` — do not separate the profiles (default)
- **`library`** — generate per-application profiles for libraries
- **`kernel`** — generate per-application profiles for the kernel and kernel modules
- **`all`** — generate per-application profiles for libraries and per-application profiles for the kernel and kernel modules

If `--separate=library` is used, the sample file name includes the name of the executable as well as the name of the library.

Note

These configuration changes will take effect when **`oprofile`** is restarted.

27.3. Starting and Stopping OProfile

To start monitoring the system with OProfile, execute the following command as root:

```
opcontrol --start
```

Output similar to the following is displayed:

```
Using log file /var/lib/oprofile/oprofiled.log Daemon started. Profiler
 running.
```

The settings in **`/root/.oprofile/daemonrc`** are used.

The OProfile daemon, **`oprofiled`**, is started; it periodically writes the sample data to the **`/var/lib/oprofile/samples/`** directory. The log file for the daemon is located at **`/var/lib/oprofile/oprofiled.log`**.

To stop the profiler, execute the following command as root:

```
opcontrol --shutdown
```

27.4. Saving Data

Sometimes it is useful to save samples at a specific time. For example, when profiling an executable, it may be useful to gather different samples based on different input data sets. If the number of events to be monitored exceeds the number of counters available for the processor, multiple runs of OProfile can be used to collect data, saving the sample data to different files each time.

To save the current set of sample files, execute the following command, replacing <*name*> with a unique descriptive name for the current session.

```
opcontrol --save=<name>
```

The directory **`/var/lib/oprofile/samples/name/`** is created and the current sample files are copied to it.

27.5. Analyzing the Data

Periodically, the OProfile daemon, **`oprofiled`**, collects the samples and writes them to the **`/var/lib/oprofile/samples/`** directory. Before reading the data, make sure all data has been written to this directory by executing the following command as root:

```
opcontrol --dump
```

Each sample file name is based on the name of the executable. For example, the samples for the default event on a Pentium III processor for **`/bin/bash`** becomes:

```
\{root\}/bin/bash/\{dep\}/\{root\}/bin/bash/CPU_CLK_UNHALTED.100000
```

The following tools are available to profile the sample data once it has been collected:

- **`opreport`**
- **`opannotate`**

Use these tools, along with the binaries profiled, to generate reports that can be further analyzed.

Warning

The executable being profiled must be used with these tools to analyze the data. If it must change after the data is collected, backup the executable used to create the samples as well as the sample files. Please note that the sample file and the binary have to agree. Making a backup isn't going to work if they do not match. **`oparchive`** can be used to address this problem.

Samples for each executable are written to a single sample file. Samples from each dynamically linked library are also written to a single sample file. While OProfile is running, if the executable being monitored changes and a sample file for the executable exists, the existing sample file is automatically deleted. Thus, if the existing sample file is needed, it must be backed up, along with the executable used to create it before replacing the executable with a new version. The oprofile analysis tools use the executable file that created the samples during analysis. If the executable changes the analysis tools will be unable to analyze the associated samples. Refer to *Section 27.4, “Saving Data”* for details on how to backup the sample file.

27.5.1. Using `opreport`

The **`opreport`** tool provides an overview of all the executables being profiled.

The following is part of a sample output:

```
Profiling through timer interrupt
TIMER:0|
samples|      %|
------------------
25926 97.5212 no-vmlinux
359  1.3504 pi
65  0.2445 Xorg
62  0.2332 libvte.so.4.4.0
56  0.2106 libc-2.3.4.so
34  0.1279 libglib-2.0.so.0.400.7
19  0.0715 libXft.so.2.1.2
17  0.0639 bash
8  0.0301 ld-2.3.4.so
8  0.0301 libgdk-x11-2.0.so.0.400.13
6  0.0226 libgobject-2.0.so.0.400.7
5  0.0188 oprofiled
4  0.0150 libpthread-2.3.4.so
4  0.0150 libgtk-x11-2.0.so.0.400.13
3  0.0113 libXrender.so.1.2.2
3  0.0113 du
1  0.0038 libcrypto.so.0.9.7a
1  0.0038 libpam.so.0.77
1  0.0038 libtermcap.so.2.0.8
1  0.0038 libX11.so.6.2
1  0.0038 libgthread-2.0.so.0.400.7
1  0.0038 libwnck-1.so.4.9.0
```

Each executable is listed on its own line. The first column is the number of samples recorded for the executable. The second column is the percentage of samples relative to the total number of samples. The third column is the name of the executable.

Refer to the **`opreport`** man page for a list of available command line options, such as the `-r` option used to sort the output from the executable with the smallest number of samples to the one with the largest number of samples.

27.5.2. Using `opreport` on a Single Executable

To retrieve more detailed profiled information about a specific executable, use **`opreport`**:

```
opreport <mode><executable>
```

<*executable*> must be the full path to the executable to be analyzed. <*mode*> must be one of the following:

-l

List sample data by symbols. For example, the following is part of the output from running the command **opreport -l /lib/tls/libc-<*version*>.so**:

```
samples % symbol name 12 21.4286 __gconv_transform_utf8_internal 5
8.9286 _int_malloc 4 7.1429 malloc 3 5.3571 __i686.get_pc_thunk.bx
3 5.3571 _dl_mcount_wrapper_check 3 5.3571 mbrtowc 3 5.3571 memcpy
2 3.5714 _int_realloc 2 3.5714 _nl_intern_locale_data 2 3.5714
free 2 3.5714 strcmp 1 1.7857 __ctype_get_mb_cur_max 1 1.7857
__unregister_atfork 1 1.7857 __write_nocancel 1 1.7857 _dl_addr 1
1.7857 _int_free 1 1.7857 _itoa_word 1 1.7857 calc_eclosure_iter 1
1.7857 fopen@@GLIBC_2.1 1 1.7857 getpid 1 1.7857 memmove 1 1.7857
msort_with_tmp 1 1.7857 strcpy 1 1.7857 strlen 1 1.7857 vfprintf 1
1.7857 write
```

The first column is the number of samples for the symbol, the second column is the percentage of samples for this symbol relative to the overall samples for the executable, and the third column is the symbol name.

To sort the output from the largest number of samples to the smallest (reverse order), use -r in conjunction with the -l option.

-i <*symbol-name*>

List sample data specific to a symbol name. For example, the following output is from the command **opreport -l -i __gconv_transform_utf8_internal /lib/tls/libc-<*version*>.so**:

```
samples % symbol name 12 100.000 __gconv_transform_utf8_internal
```

The first line is a summary for the symbol/executable combination.

The first column is the number of samples for the memory symbol. The second column is the percentage of samples for the memory address relative to the total number of samples for the symbol. The third column is the symbol name.

-d

List sample data by symbols with more detail than -l. For example, the following output is from the command **opreport -l -d __gconv_transform_utf8_internal /lib/tls/libc-<*version*>.so**:

```
vma samples % symbol name 00a98640 12 100.000
__gconv_transform_utf8_internal 00a98640 1 8.3333 00a9868c 2 16.6667
00a9869a 1 8.3333 00a986c1 1 8.3333 00a98720 1 8.3333 00a98749 1 8.3333
00a98753 1 8.3333 00a98789 1 8.3333 00a98864 1 8.3333 00a98869 1 8.3333
00a98b08 1 8.3333
```

The data is the same as the -l option except that for each symbol, each virtual memory address used is shown. For each virtual memory address, the number of samples and percentage of samples relative to the number of samples for the symbol is displayed.

`-x<`*`symbol-name`*`>`
: Exclude the comma-separated list of symbols from the output.

`session:<`*`name`*`>`
: Specify the full path to the session or a directory relative to the **`/var/lib/oprofile/samples/`** directory.

27.5.3. Getting more detailed output on the modules

OProfile collects data on a system-wide basis for kernel- and user-space code running on the machine. However, once a module is loaded into the kernel, the information about the origin of the kernel module is lost. The module could have come from the initrd file on boot up, the directory with the various kernel modules, or a locally created kernel module. As a result when OProfile records sample for a module, it just lists the samples for the modules for an executable in the root directory, but this is unlikely to be the place with the actual code for the module. You will need to take some steps to make sure that analysis tools get the executable.

For example on an AMD64 machine the sampling is set up to record "Data cache accesses" and "Data cache misses" and assuming you would like to see the data for the ext3 module:

```
$ opreport /ext3
CPU: AMD64 processors, speed 797.948 MHz (estimated)
Counted DATA_CACHE_ACCESSES events (Data cache accesses) with a unit mask
 of 0x00 (No unit mask) count 500000
Counted DATA_CACHE_MISSES events (Data cache misses) with a unit mask of
 0x00 (No unit mask) count 500000
DATA_CACHE_ACC...|DATA_CACHE_MIS...|
samples|      %|  samples|      %|
------------------------------------
148721 100.000      1493 100.000 ext3
```

To get a more detailed view of the actions of the module, you will need to either have the module unstripped (e.g. installed from a custom build) or have the debuginfo RPM installed for the kernel.

Find out which kernel is running, "uname -a", get the appropriate debuginfo rpm, and install on the machine.

Then make a symbolic link so oprofile finds the code for the module in the correct place:

```
# ln -s /lib/modules/`uname -r`/kernel/fs/ext3/ext3.ko /ext3
```

Then the detailed information can be obtained with:

```
# opreport image:/ext3 -l|more
warning: could not check that the binary file /ext3 has not been modified
 since the profile was taken. Results may be inaccurate.
CPU: AMD64 processors, speed 797.948 MHz (estimated)
Counted DATA_CACHE_ACCESSES events (Data cache accesses) with a unit mask
 of 0x00 (No unit mask) count 500000
```

```
Counted DATA_CACHE_MISSES events (Data cache misses) with a unit mask of
 0x00 (No unit mask) count 500000
samples  %        samples  %        symbol name
16728    11.2479  7          0.4689  ext3_group_sparse
16454    11.0637  4          0.2679  ext3_count_free_blocks
14583     9.8056  51         3.4159  ext3_fill_super
8281      5.5681  129        8.6403  ext3_ioctl
7810      5.2514  62         4.1527  ext3_write_info
7286      4.8991  67         4.4876  ext3_ordered_writepage
6509      4.3767  130        8.7073  ext3_new_inode
6378      4.2886  156       10.4488  ext3_new_block
5932      3.9887  87         5.8272  ext3_xattr_block_list
...
```

27.5.4. Using opannotate

The **opannotate** tool tries to match the samples for particular instructions to the corresponding lines in the source code. The resulting files generated should have the samples for the lines at the left. It also puts in a comment at the beginning of each function listing the total samples for the function.

For this utility to work, the executable must be compiled with GCC's `-g` option. By default, Fedora packages are not compiled with this option.

The general syntax for **opannotate** is as follows:

```
opannotate --search-dirs <src-dir> --source <executable>
```

The directory containing the source code and the executable to be analyzed must be specified. Refer to the **opannotate** man page for a list of additional command line options.

27.6. Understanding /dev/oprofile/

The **/dev/oprofile/** directory contains the file system for OProfile. Use the **cat** command to display the values of the virtual files in this file system. For example, the following command displays the type of processor OProfile detected:

```
cat /dev/oprofile/cpu_type
```

A directory exists in **/dev/oprofile/** for each counter. For example, if there are 2 counters, the directories **/dev/oprofile/0/** and **dev/oprofile/1/** exist.

Each directory for a counter contains the following files:

- **count** — The interval between samples.
- **enabled** — If 0, the counter is off and no samples are collected for it; if 1, the counter is on and samples are being collected for it.
- **event** — The event to monitor.

- **kernel** — If 0, samples are not collected for this counter event when the processor is in kernel-space; if 1, samples are collected even if the processor is in kernel-space.
- **unit_mask** — Defines which unit masks are enabled for the counter.
- **user** — If 0, samples are not collected for the counter event when the processor is in user-space; if 1, samples are collected even if the processor is in user-space.

The values of these files can be retrieved with the **cat** command. For example:

```
cat /dev/oprofile/0/count
```

27.7. Example Usage

While OProfile can be used by developers to analyze application performance, it can also be used by system administrators to perform system analysis. For example:

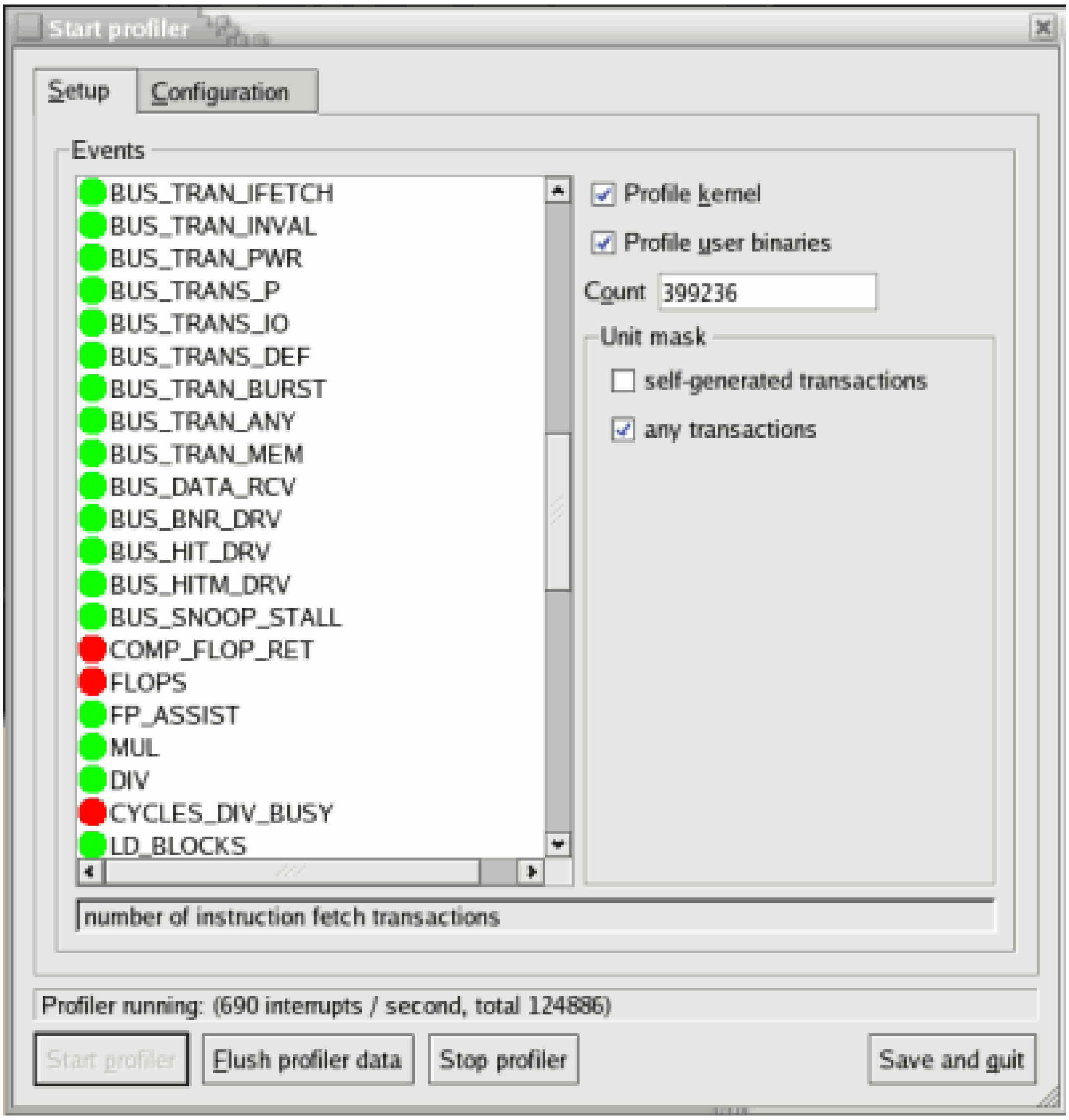
- *Determine which applications and services are used the most on a system* — **opreport** can be used to determine how much processor time an application or service uses. If the system is used for multiple services but is under performing, the services consuming the most processor time can be moved to dedicated systems.
- *Determine processor usage* — The CPU_CLK_UNHALTED event can be monitored to determine the processor load over a given period of time. This data can then be used to determine if additional processors or a faster processor might improve system performance.

27.8. Graphical Interface

Some OProfile preferences can be set with a graphical interface. To start it, execute the **oprof_start** command as root at a shell prompt. To use the graphical interface, you will need to have the **oprofile-gui** package installed.

After changing any of the options, save them by clicking the **Save and quit** button. The preferences are written to **/root/.oprofile/daemonrc**, and the application exits. Exiting the application does not stop OProfile from sampling.

On the **Setup** tab, to set events for the processor counters as discussed in *Section 27.2.2, “Setting Events to Monitor”*, select the counter from the pulldown menu and select the event from the list. A brief description of the event appears in the text box below the list. Only events available for the specific counter and the specific architecture are displayed. The interface also displays whether the profiler is running and some brief statistics about it.



🟢 BUS_TRAN_IFETCH
🟢 BUS_TRAN_INVAL
🟢 BUS_TRAN_PWR
🟢 BUS_TRANS_P
🟢 BUS_TRANS_IO
🟢 BUS_TRANS_DEF
🟢 BUS_TRAN_BURST
🟢 BUS_TRAN_ANY
🟢 BUS_TRAN_MEM
🟢 BUS_DATA_RCV
🟢 BUS_BNR_DRV
🟢 BUS_HIT_DRV
🟢 BUS_HITM_DRV
🟢 BUS_SNOOP_STALL
🔴 COMP_FLOP_RET
🔴 FLOPS
🟢 FP_ASSIST
🟢 MUL
🟢 DIV
🔴 CYCLES_DIV_BUSY
🟢 LD_BLOCKS

☑ Profile kernel
☑ Profile user binaries
Count 399236

Unit mask
☐ self-generated transactions
☑ any transactions

number of instruction fetch transactions

Profiler running: (690 interrupts / second, total 124886)

Figure 27.1. OProfile Setup

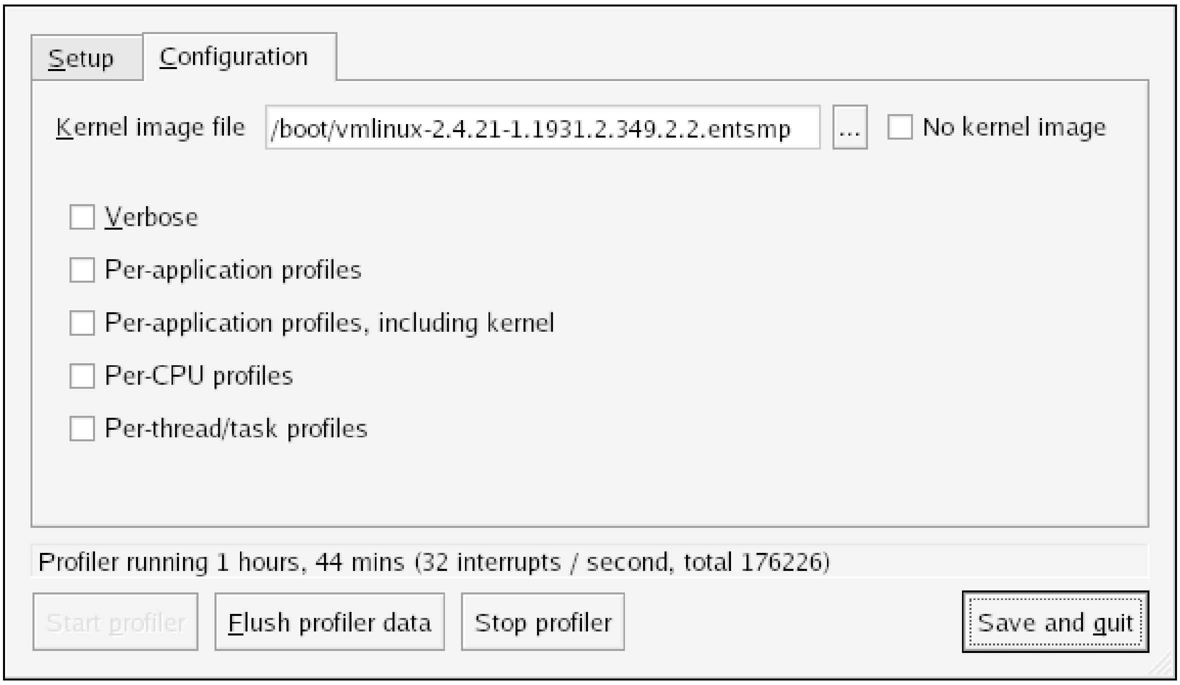
On the right side of the tab, select the **Profile kernel** option to count events in kernel mode for the currently selected event, as discussed in *Section 27.2.3, “Separating Kernel and User-space Profiles”*. If this option is unselected, no samples are collected for the kernel.

Select the **Profile user binaries** option to count events in user mode for the currently selected event, as discussed in *Section 27.2.3, “Separating Kernel and User-space Profiles”*. If this option is unselected, no samples are collected for user applications.

Use the **Count** text field to set the sampling rate for the currently selected event as discussed in *Section 27.2.2.1, “Sampling Rate”*.

If any unit masks are available for the currently selected event, as discussed in *Section 27.2.2.2, “Unit Masks”*, they are displayed in the **Unit Masks** area on the right side of the **Setup** tab. Select the checkbox beside the unit mask to enable it for the event.

On the **Configuration** tab, to profile the kernel, enter the name and location of the `vmlinux` file for the kernel to monitor in the **Kernel image file** text field. To configure OProfile not to monitor the kernel, select **No kernel image**.



Kernel image file /boot/vmlinux-2.4.21-1.1931.2.349.2.2.entsmp

□ No kernel image

□ Verbose

□ Per-application profiles

□ Per-application profiles, including kernel

□ Per-CPU profiles

□ Per-thread/task profiles

Profiler running 1 hours, 44 mins (32 interrupts / second, total 176226)

Figure 27.2. OProfile Configuration

If the **Verbose** option is selected, the **`oprofiled`** daemon log includes more information.

If **Per-application kernel samples files** is selected, OProfile generates per-application profiles for the kernel and kernel modules as discussed in *Section 27.2.3, “Separating Kernel and User-space Profiles”*. This is equivalent to the **`opcontrol --separate=kernel`** command. If **Per-application shared libs samples files** is selected, OProfile generates per-application profiles for libraries. This is equivalent to the **`opcontrol --separate=library`** command.

To force data to be written to samples files as discussed in *Section 27.5, “Analyzing the Data”*, click the **Flush profiler data** button. This is equivalent to the **`opcontrol --dump`** command.

To start OProfile from the graphical interface, click **Start profiler**. To stop the profiler, click **Stop profiler**. Exiting the application does not stop OProfile from sampling.

27.9. Additional Resources

This chapter only highlights OProfile and how to configure and use it. To learn more, refer to the following resources.

27.9.1. Installed Docs

- **`/usr/share/doc/oprofile-<`*`version`*`>/oprofile.html`** — *OProfile Manual*

- **`oprofile`** man page — Discusses **`opcontrol`**, **`opreport`**, **`opannotate`**, and **`ophelp`**

27.9.2. Useful Websites

- *http://oprofile.sourceforge.net/* — Contains the latest documentation, mailing lists, IRC channels, and more.

ABRT

28.1. Overview

ABRT is the Automatic Bug-Reporting Tool. ABRT consists of a daemon that runs silently in the background most of the time. It springs into action when an application crashes; it collects the relevant crash data such as a core file if there is one, the command line parameters to the application, and other contextual puzzle pieces of forensic utility. Finally, through its modular, plugin-oriented architecture, ABRT provides a number of ways to transmit the crash information to a location, such as by HTTP to Red Hat's Bugzilla.

28.2. Installing and Starting the Daemon, Applet and GUI

28.3. Configuring

C

28.4. Plugins and Sending Crash Reports

p

28.4.1. Reporting to Bugzilla

R

28.4.2. Emailing the Report

E

28.4.3. Transferring via SCP or FTP

T

28.4.4. Other plugins

O

Part V. Kernel and Driver Configuration

System administrators can learn about and customize their kernels. Fedora contains kernel tools to assist administrators with their customizations.

Manually Upgrading the Kernel

The Fedora kernel is custom built by the Fedora kernel team to ensure its integrity and compatibility with supported hardware. Before Red Hat releases a kernel, it must first pass a rigorous set of quality assurance tests.

Fedora kernels are packaged in RPM format so that they are easy to upgrade and verify using the **Package Management Tool**, or the **yum** command. The **Package Management Tool** automatically queries the Fedora servers and determines which packages need to be updated on your machine, including the kernel. This chapter is *only* useful for those individuals that require manual updating of kernel packages, without using the **yum** command.

> **Warning**
>
> Building a custom kernel is not supported by the Red Hat Global Services Support team, and therefore is not explored in this manual.

> **Tip**
>
> The use of **yum** is *highly* recommended by Red Hat for installing upgraded kernels.

29.1. Overview of Kernel Packages

Fedora contains the following kernel packages (some may not apply to your architecture):

- **kernel** — Contains the kernel for multi-processor systems. For x86 system, only the first 4GB of RAM is used. As such, x86 systems with over 4GB of RAM should use the **kernel-PAE**.
- **kernel-devel** — Contains the kernel headers and makefiles sufficient to build modules against the **kernel** package.
- **kernel-PAE** (only for i686 systems) — This package offers the following key configuration option (in addition to the options already enabled for the **kernel** package):
    - PAE (Physical Address Extension) support for systems with more than 4GB of RAM, and reliably up to 16GB.

    > **Important**
    >
    > Physical Address Extension allows x86 processors to address up to 64GB of physical RAM, but due to differences between the Red Hat Enterprise Linux 4 and 5 kernels, only Red Hat Enterprise Linux 4 (with the **kernel-hugemem** package) is able to reliably address all 64GB of memory. Additionally, the Red Hat Enterprise Linux 5 PAE variant does not allow 4GB of addressable memory per-process like the Red Hat Enterprise Linux 4 **kernel-hugemem** variant does. However, the x86_64 kernel does not suffer from any of these limitations, and is the suggested Red Hat Enterprise Linux 5 architecture to use with large-memory systems.

- **kernel-PAE-devel** — Contains the kernel headers and makefiles required to build modules against the **kernel-PAE** package
- **kernel-doc** — Contains documentation files from the kernel source. Various portions of the Linux kernel and the device drivers shipped with it are documented in these files. Installation of this package provides a reference to the options that can be passed to Linux kernel modules at load time.

  By default, these files are placed in the **/usr/share/doc/kernel-doc-<*version*>/** directory.
- **kernel-headers** — Includes the C header files that specify the interface between the Linux kernel and userspace libraries and programs. The header files define structures and constants that are needed for building most standard programs.
- **kernel-xen** — Includes a version of the Linux kernel which is needed to run Virtualization.
- **kernel-xen-devel** — Contains the kernel headers and makefiles required to build modules against the **kernel-xen** package

> **Note**
>
> The **kernel-source** package has been removed and replaced with an RPM that can only be retrieved from Red Hat Network. This ***.src.rpm** package must then be rebuilt locally using the **rpmbuild** command. For more information on obtaining and installing the kernel source package, refer to the latest updated Release Notes (including all updates) at *http://www.redhat.com/docs/manuals/enterprise/*

29.2. Preparing to Upgrade

Before upgrading the kernel, it is recommended that you take some precautionary steps. The first step is to make sure working boot media exists for the system in case a problem occurs. If the boot loader is not configured properly to boot the new kernel, the system cannot be booted into Fedora without working boot media.

To create a boot diskette, login as root, and run the command **/sbin/mkbootdisk \`uname -r\`** at a shell prompt.

> **Tip**
>
> Refer to the **mkbootdisk** man page for more options. You can create bootable media via CD-Rs, CD-RWs, and USB flash drives, provided that your system BIOS also supports it.

Reboot the machine with the boot media and verify that it works before continuing.

To determine which kernel packages are installed, execute the command **rpm -qa | grep kernel** at a shell prompt:

The output contains some or all of the following packages, depending on the system's architecture (the version numbers and packages may differ):

```
kernel-2.6.9-5.EL
kernel-devel-2.6.9-5.EL
kernel-utils-2.6.9-5.EL
kernel-doc-2.6.9-5.EL
kernel-smp-2.6.9-5.EL
kernel-smp-devel-2.6.9-5.EL
kernel-hugemem-devel-2.6.9-5.EL
```

From the output, determine which packages need to be download for the kernel upgrade. For a single processor system, the only required package is the **kernel** package. Refer to *Section 29.1, “Overview of Kernel Packages”* for descriptions of the different packages.

In the file name, each kernel package contains the architecture for which the package was built. The format is kernel-<*variant*>-<*version*>.<*arch*>.rpm, where <*variant*> is one of either **PAE**, **xen**, and so forth. The <*arch*> is one of the following:

- **x86_64** for the AMD64 and Intel EM64T architectures
- **ia64** for the Intel®Itanium™ architecture
- **ppc64** for the IBM®eServer™pSeries™ architecture
- **s390** for the IBM®S/390® architecture
- **s390x** for the IBM®eServer™System z® architecture
- **i686** for Intel®Pentium® II, Intel®Pentium® III, Intel®Pentium® 4, AMD Athlon®, and AMD Duron® systems

29.3. Downloading the Upgraded Kernel

There are several ways to determine if an updated kernel is available for the system.

- Security Errata — Refer to *http://www.redhat.com/security/updates/* for information on security errata, including kernel upgrades that fix security issues.
- Via Red Hat Network — Download and install the kernel RPM packages. Red Hat Network can download the latest kernel, upgrade the kernel on the system, create an initial RAM disk image if needed, and configure the boot loader to boot the new kernel. For more information, refer to *http://www.redhat.com/docs/manuals/RHNetwork/*[1].

If Red Hat Network was used to download and install the updated kernel, follow the instructions in *Section 29.5, “Verifying the Initial RAM Disk Image”* and *Section 29.6, “Verifying the Boot Loader”*, only *do not* change the kernel to boot by default. Red Hat Network automatically changes the default kernel to the latest version. To install the kernel manually, continue to *Section 29.4, “Performing the Upgrade”*.

29.4. Performing the Upgrade

After retrieving all of the necessary packages, it is time to upgrade the existing kernel.

Important

It is strongly recommended that you keep the old kernel in case there are problems with the new kernel.

At a shell prompt, change to the directory that contains the kernel RPM packages. Use `-i` argument with the **`rpm`** command to keep the old kernel. Do *not* use the `-U` option, since it overwrites the currently installed kernel, which creates boot loader problems. For example:

```
rpm -ivh kernel-<kernel version>.<arch>.rpm
```

The next step is to verify that the initial RAM disk image has been created. Refer to *Section 29.5, "Verifying the Initial RAM Disk Image"* for details.

29.5. Verifying the Initial RAM Disk Image

If the system uses the ext3 file system, a SCSI controller, or labels to reference partitions in **`/etc/fstab`**, an initial RAM disk is needed. The initial RAM disk allows a modular kernel to have access to modules that it might need to boot from before the kernel has access to the device where the modules normally reside.

On architectures other than IBM eServer iSeries, the initial RAM disk can be created with the **`mkinitrd`** command. However, this step is performed automatically if the kernel and its associated packages are installed or upgraded from the RPM packages distributed by Red Hat; in such cases, you do not need to create the initial RAM disk manually. To verify that an initial RAM disk already exists, use the command **`ls -l /boot`** to make sure the **`initrd-<version>.img`** file was created (the version should match the version of the kernel just installed).

On iSeries systems, the initial RAM disk file and **`vmlinux`** file are combined into one file, which is created with the **`addRamDisk`** command. This step is performed automatically if the kernel and its associated packages are installed or upgraded from the RPM packages distributed by Red Hat, Inc.; thus, it does not need to be executed manually. To verify that it was created, use the command **`ls -l /boot`** to make sure the **`/boot/vmlinitrd-<kernel-version>`** file already exists (the **`<kernel-version>`** should match the version of the kernel just installed).

The next step is to verify that the boot loader has been configured to boot the new kernel. Refer to *Section 29.6, "Verifying the Boot Loader"* for details.

29.6. Verifying the Boot Loader

The **`kernel`** RPM package configures the boot loader to boot the newly installed kernel (except for IBM eServer iSeries systems). However, it does not configure the boot loader to boot the new kernel by default.

It is always a good idea to confirm that the boot loader has been configured correctly. This is a crucial step. If the boot loader is configured incorrectly, the system will not boot into Fedora properly. If this happens, boot the system with the boot media created earlier and try configuring the boot loader again.

29.6.1. x86 Systems

All x86 systems (including all AMD64 systems) use GRUB as the boot loader.

29.6.1.1. GRUB

Confirm that the file **`/boot/grub/grub.conf`** contains a `title` section with the same version as the **`kernel`** package just installed

```
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/hda2
#          initrd /initrd-version.img
#boot=/dev/hda
default=1 timeout=10
splashimage=(hd0,0)/grub/splash.xpm.gz
title Red Hat Enterprise Linux (2.6.9-5.EL)
         root (hd0,0)
  kernel /vmlinuz-2.6.9-5.EL ro root=LABEL=/
  initrd /initrd-2.6.9-5.EL.img
title Red Hat Enterprise Linux (2.6.9-1.906_EL)
         root (hd0,0)
  kernel /vmlinuz-2.6.9-1.906_EL ro root=LABEL=/
  initrd /initrd-2.6.9-1.906_EL.img
```

If a separate **`/boot/`** partition was created, the paths to the kernel and **`initrd`** image are relative to **`/boot/`**.

Notice that the default is not set to the new kernel. To configure GRUB to boot the new kernel by default, change the value of the **`default`** variable to the title section number for the title section that contains the new kernel. The count starts with 0. For example, if the new kernel is the first title section, set **`default`** to **`0`**.

Begin testing the new kernel by rebooting the computer and watching the messages to ensure that the hardware is detected properly.

29.6.2. Itanium Systems

Itanium systems use ELILO as the boot loader, which uses **`/boot/efi/EFI/redhat/elilo.conf`** as the configuration file. Confirm that this file contains an `image` section with the same version as the **`kernel`** package just installed:

```
prompt timeout=50 default=old  image=vmlinuz-2.6.9-5.EL
         label=linux
  initrd=initrd-2.6.9-5.EL.img         read-only
  append="root=LABEL=/" image=vmlinuz-2.6.9-1.906_EL
  label=old
  initrd=initrd-2.6.9-1.906.img         read-only
  append="root=LABEL=/"
```

Notice that the default is not set to the new kernel. To configure ELILO to boot the new kernel, change the value of the `default` variable to the value of the `label` for the `image` section that contains the new kernel.

Begin testing the new kernel by rebooting the computer and watching the messages to ensure that the hardware is detected properly.

29.6.3. IBM S/390 and IBM System z Systems

The IBM S/390 and IBM System z systems use z/IPL as the boot loader, which uses **`/etc/zipl.conf`** as the configuration file. Confirm that the file contains a section with the same version as the kernel package just installed:

```
[defaultboot] default=old target=/boot/
[linux]
         image=/boot/vmlinuz-2.6.9-5.EL
  ramdisk=/boot/initrd-2.6.9-5.EL.img
  parameters="root=LABEL=/"
[old]
         image=/boot/vmlinuz-2.6.9-1.906_EL
  ramdisk=/boot/initrd-2.6.9-1.906_EL.img
  parameters="root=LABEL=/"
```

Notice that the default is not set to the new kernel. To configure z/IPL to boot the new kernel by default, change the value of the `default` variable to the name of the section that contains the new kernel. The first line of each section contains the name in brackets.

After modifying the configuration file, run **`/sbin/zipl`** as root to enable the changes.

Begin testing the new kernel by rebooting the computer and watching the messages to ensure that the hardware is detected properly.

29.6.4. IBM eServer iSeries Systems

The **`/boot/vmlinitrd-<`*`kernel-version`*`>`** file is installed when you upgrade the kernel. However, you must use the **dd** command to configure the system to boot the new kernel:

1. As root, issue the command **`cat /proc/iSeries/mf/side`** to determine the default side (either A, B, or C).

2. As root, issue the following command, where <*`kernel-version`*> is the version of the new kernel and <*`side`*> is the side from the previous command:

   `dd if=/boot/vmlinitrd-<`*`kernel-version`*`> of=/proc/iSeries/mf/<`*`side`*`>/vmlinux bs=8k`

Begin testing the new kernel by rebooting the computer and watching the messages to ensure that the hardware is detected properly.

29.6.5. IBM eServer pSeries Systems

IBM eServer pSeries systems use YABOOT as the boot loader, which uses **`/etc/aboot.conf`** as the configuration file. Confirm that the file contains an `image` section with the same version as the **`kernel`** package just installed:

```
boot=/dev/sda1 init-message=Welcome to Fedora! Hit <TAB> for boot options
partition=2 timeout=30 install=/usr/lib/yaboot/yaboot delay=10 nonvram
image=/vmlinux--2.6.9-5.EL
         label=old
  read-only
  initrd=/initrd--2.6.9-5.EL.img
  append="root=LABEL=/"
image=/vmlinux-2.6.9-5.EL
  label=linux
  read-only
  initrd=/initrd-2.6.9-5.EL.img
  append="root=LABEL=/"
```

Notice that the default is not set to the new kernel. The kernel in the first image is booted by default. To change the default kernel to boot either move its image stanza so that it is the first one listed or add the directive `default` and set it to the `label` of the image stanza that contains the new kernel.

Begin testing the new kernel by rebooting the computer and watching the messages to ensure that the hardware is detected properly.

Chapter 30.

General Parameters and Modules

This chapter is provided to illustrate *some* of the possible parameters available for common hardware device *drivers*[1], which under Fedora are called kernel *modules*. In most cases, the default parameters do work. However, there may be times when extra module parameters are necessary for a device to function properly or to override the module's default parameters for the device.

During installation, Fedora uses a limited subset of device drivers to create a stable installation environment. Although the installation program supports installation on many different types of hardware, some drivers (including those for SCSI adapters and network adapters) are not included in the installation kernel. Rather, they must be loaded as modules by the user at boot time.

Once installation is completed, support exists for a large number of devices through kernel modules.

Important

Red Hat provides a large number of unsupported device drivers in groups of packages called **kernel-smp-unsupported-*<kernel-version>*** and **kernel-hugemem-unsupported-*<kernel-version>***. Replace *<kernel-version>* with the version of the kernel installed on the system. These packages are not installed by the Fedora installation program, and the modules provided are not supported by Red Hat, Inc.

30.1. Kernel Module Utilities

A group of commands for managing kernel modules is available if the **module-init-tools** package is installed. Use these commands to determine if a module has been loaded successfully or when trying different modules for a piece of new hardware.

The command **/sbin/lsmod** displays a list of currently loaded modules. For example:

```
Module                  Size  Used by
tun                    11585  1
autofs4                21573  1
hidp                   16193  2
rfcomm                 37849  0
l2cap                  23873  10 hidp,rfcomm
bluetooth              50085  5 hidp,rfcomm,l2cap
sunrpc                153725  1
dm_mirror              29073  0
dm_mod                 57433  1 dm_mirror
video                  17221  0
sbs                    16257  0
i2c_ec                  5569  1 sbs
container               4801  0
button                  7249  0
battery                10565  0
asus_acpi              16857  0
```

[1] A driver is software which enables Linux to use a particular hardware device. Without a driver, the kernel cannot communicate with attached devices.

```
ac                       5701  0
ipv6                   246113  12
lp                      13065  0
parport_pc              27493  1
parport                 37001  2 lp,parport_pc
uhci_hcd                23885  0
floppy                  57317  1
sg                      34653  0
snd_ens1371             26721  1
gameport                16073  1 snd_ens1371
snd_rawmidi             24897  1 snd_ens1371
snd_ac97_codec          91360  1 snd_ens1371
snd_ac97_bus             2753  1 snd_ac97_codec
snd_seq_dummy            4293  0
snd_seq_oss             32705  0
serio_raw                7493  0
snd_seq_midi_event       8001  1 snd_seq_oss
snd_seq                 51633  5
 snd_seq_dummy,snd_seq_oss,snd_seq_midi_event
snd_seq_device           8781  4
 snd_rawmidi,snd_seq_dummy,snd_seq_oss,snd_seq
snd_pcm_oss             42849  0
snd_mixer_oss           16833  1 snd_pcm_oss
snd_pcm                 76485  3 snd_ens1371,snd_ac97_codec,snd_pcm_oss
snd_timer               23237  2 snd_seq,snd_pcm
snd                     52933  12
 snd_ens1371,snd_rawmidi,snd_ac97_codec,snd_seq_oss,snd_seq,snd_seq_device,snd_pcm_oss,snd_
soundcore               10145  1 snd
i2c_piix4                8909  0
ide_cd                  38625  3
snd_page_alloc          10569  1 snd_pcm
i2c_core                21697  2 i2c_ec,i2c_piix4
pcnet32                 34117  0
cdrom                   34913  1 ide_cd
mii                      5825  1 pcnet32
pcspkr                   3521  0
ext3                   129737  2
jbd                     58473  1 ext3
mptspi                  17353  3
scsi_transport_spi      25025  1 mptspi
mptscsih                23361  1 mptspi
sd_mod                  20929  16
scsi_mod               134121  5
 sg,mptspi,scsi_transport_spi,mptscsih,sd_mod
mptbase                 52193  2 mptspi,mptscsih
```

For each line, the first column is the name of the module, the second column is the size of the module, and the third column is the use count.

The **`/sbin/lsmod`** output is less verbose and easier to read than the output from viewing **`/proc/modules`**.

To load a kernel module, use the **/sbin/modprobe** command followed by the kernel module name. By default, **modprobe** attempts to load the module from the **/lib/modules/*<kernel-version>*/kernel/drivers/** subdirectories. There is a subdirectory for each type of module, such as the **net/** subdirectory for network interface drivers. Some kernel modules have module dependencies, meaning that other modules must be loaded first for it to load. The **/sbin/modprobe** command checks for these dependencies and loads the module dependencies before loading the specified module.

For example, the command

```
/sbin/modprobe e100
```

loads any module dependencies and then the **e100** module.

To print to the screen all commands as **/sbin/modprobe** executes them, use the `-v` option. For example:

```
/sbin/modprobe -v e100
```

Output similar to the following is displayed:

```
/sbin/insmod /lib/modules/2.6.9-5.EL/kernel/drivers/net/e100.ko
Using /lib/modules/2.6.9-5.EL/kernel/drivers/net/e100.ko
Symbol version prefix 'smp_'
```

The **/sbin/insmod** command also exists to load kernel modules; however, it does not resolve dependencies. Thus, it is recommended that the **/sbin/modprobe** command be used.

To unload kernel modules, use the **/sbin/rmmod** command followed by the module name. The **rmmod** utility only unloads modules that are not in use and that are not a dependency of other modules in use.

For example, the command

```
/sbin/rmmod e100
```

unloads the **e100** kernel module.

Another useful kernel module utility is **modinfo**. Use the command **/sbin/modinfo** to display information about a kernel module. The general syntax is:

```
/sbin/modinfo [options]<module>
```

Options include **-d**, which displays a brief description of the module, and **-p**, which lists the parameters the module supports. For a complete list of options, refer to the **modinfo** man page (**man modinfo**).

30.2. Persistent Module Loading

Kernel modules are usually loaded directly by the facility that requires them, which is given correct settings in the `/etc/modprobe.conf` file. However, it is sometimes necessary to explicitly force the loading of a module at boot time.

Fedora checks for the existence of the `/etc/rc.modules` file at boot time, which contains various commands to load modules. The `rc.modules` should be used, and *not*`rc.local` because `rc.modules` is executed earlier in the boot process.

For example, the following commands configure loading of the `foo` module at boot time (as root):

```
# echo modprobe foo >> /etc/rc.modules
# chmod +x /etc/rc.modules
```

Tip

This approach is not necessary for network and SCSI interfaces because they have their own specific mechanisms.

30.3. Specifying Module Parameters

In some situations, it may be necessary to supply parameters to a module as it is loaded for it to function properly.

For instance, to enable full duplex at 100Mbps connection speed for an Intel Ether Express/100 card, load the `e100` driver with the `e100_speed_duplex=4` option.

Caution

When a parameter has commas, be sure *not* to put a space after a comma.

Tip

The `modinfo` command is also useful for listing various information about a kernel module, such as version, dependencies, paramater options, and aliases.

30.4. Storage parameters

Hardware	Module	Parameters
3ware Storage Controller and 9000 series	`3w-xxxx.ko, 3w-9xxx.ko`	
Adaptec Advanced Raid Products, Dell PERC2, 2/Si, 3/Si, 3/Di, HP NetRAID-4M,	`aacraid.ko`	*nondasd* — Control scanning of hba for nondasd devices. 0=off, 1=on

Hardware	Module	Parameters
IBM ServeRAID, and ICP SCSI driver		*dacmode* — Control whether dma addressing is using 64 bit DAC. 0=off, 1=on *commit* — Control whether a COMMIT_CONFIG is issued to the adapter for foreign arrays. This is typically needed in systems that do not have a BIOS. 0=off, 1=on *startup_timeout* — The duration of time in seconds to wait for adapter to have it's kernel up and running. This is typically adjusted for large systems that do not have a BIOS *aif_timeout* — The duration of time in seconds to wait for applications to pick up AIFs before deregistering them. This is typically adjusted for heavily burdened systems. *numacb* — Request a limit to the number of adapter control blocks (FIB) allocated. Valid values are 512 and down. Default is to use suggestion from Firmware. *acbsize* — Request a specific adapter control block (FIB) size. Valid values are 512, 2048, 4096 and 8192. Default is to use suggestion from Firmware.
Adaptec 28xx, R9xx, 39xx AHA-284x, AHA-29xx, AHA-394x, AHA-398x, AHA-274x, AHA-274xT, AHA-2842, AHA-2910B, AHA-2920C, AHA-2930/U/U2, AHA-2940/W/U/UW/AU/, U2W/U2/U2B/, U2BOEM, AHA-2944D/WD/UD/UWD, AHA-2950U2/W/B, AHA-3940/U/W/UW/, AUW/U2W/U2B, AHA-3950U2D, AHA-3985/U/W/UW, AIC-777x,	**aic7xxx.ko**	*verbose* — Enable verbose/diagnostic logging *allow_memio* — Allow device registers to be memory mapped *debug* — Bitmask of debug values to enable *no_probe* — Toggle EISA/VLB controller probing *probe_eisa_vl* — Toggle EISA/VLB controller probing

Hardware	Module	Parameters
AIC-785x, AIC-786x, AIC-787x, AIC-788x , AIC-789x, AIC-3860		*no_reset* — Supress initial bus resets *extended* — Enable extended geometry on all controllers *periodic_otag* — Send an ordered tagged transaction periodically to prevent tag starvation. This may be required by some older disk drives or RAID arrays. *tag_info:<tag_str>* — Set per-target tag depth *global_tag_depth:<int>* — Global tag depth for every target on every bus *seltime:<int>* — Selection Timeout (0/256ms,1/128ms,2/64ms,3/32ms)
IBM ServeRAID	**ips.ko**	
LSI Logic MegaRAID Mailbox Driver	**megaraid_mbox.ko**	*unconf_disks* — Set to expose unconfigured disks to kernel (default=0) *busy_wait* — Max wait for mailbox in microseconds if busy (default=10) *max_sectors* — Maximum number of sectors per IO command (default=128) *cmd_per_lun* — Maximum number of commands per logical unit (default=64) *fast_load* — Faster loading of the driver, skips physical devices! (default=0) *debug_level* — Debug level for driver (default=0)
Emulex LightPulse Fibre Channel SCSI driver	**lpfc.ko**	*lpfc_poll* — FCP ring polling mode control: 0 - none, 1 - poll with interrupts enabled 3 - poll and disable FCP ring interrupts

Hardware	Module	Parameters
		lpfc_log_verbose — Verbose logging bit-mask *lpfc_lun_queue_depth* — Max number of FCP commands we can queue to a specific LUN *lpfc_hba_queue_depth* — Max number of FCP commands we can queue to a lpfc HBA *lpfc_scan_down* — Start scanning for devices from highest ALPA to lowest *lpfc_nodev_tmo* — Seconds driver will hold I/O waiting for a device to come back *lpfc_topology* — Select Fibre Channel topology *lpfc_link_speed* — Select link speed *lpfc_fcp_class* — Select Fibre Channel class of service for FCP sequences *lpfc_use_adisc* — Use ADISC on rediscovery to authenticate FCP devices *lpfc_ack0* — Enable ACK0 support *lpfc_cr_delay* — A count of milliseconds after which an interrupt response is generated *lpfc_cr_count* — A count of I/O completions after which an interrupt response is generated *lpfc_multi_ring_support* — Determines number of primary SLI rings to spread IOCB entries across *lpfc_fdmi_on* — Enable FDMI support

Hardware	Module	Parameters
		lpfc_discovery_threads — Maximum number of ELS commands during discovery *lpfc_max_luns* — Maximum allowed LUN *lpfc_poll_tmo* — Milliseconds driver will wait between polling FCP ring
HP Smart Array	cciss.ko	
LSI Logic MPT Fusion	mptbase.ko mptctl.ko mptfc.ko mptlan.ko mptsas.ko mptscsih.ko mptspi.ko	*mpt_msi_enable* — MSI Support Enable *mptfc_dev_loss_tmo* — Initial time the driver programs the transport to wait for an rport to return following a device loss event. *mpt_pt_clear* — Clear persistency table *mpt_saf_te* — Force enabling SEP Processor
QLogic Fibre Channel Driver	qla2xxx.ko	*ql2xlogintimeout* — Login timeout value in seconds. *qlport_down_retry* — Maximum number of command retries to a port that returns a PORT-DOWN status *ql2xplogiabsentdevice* — Option to enable PLOGI to devices that are not present after a Fabric scan. *ql2xloginretrycount* — Specify an alternate value for the NVRAM login retry count. *ql2xallocfwdump* — Option to enable allocation of memory for a firmware dump during HBA initialization. Default is 1 - allocate memory. *extended_error_logging* — Option to enable extended error logging.

Hardware	Module	Parameters
		`ql2xfdmienable` — Enables FDMI registratons.
NCR, Symbios and LSI 8xx and 1010	sym53c8xx	`cmd_per_lun` — The maximum number of tags to use by default `tag_ctrl` — More detailed control over tags per LUN `burst` — Maximum burst. 0 to disable, 255 to read from registers `led` — Set to 1 to enable LED support `diff` — 0 for no differential mode, 1 for BIOS, 2 for always, 3 for not GPIO3 `irqm` — 0 for open drain, 1 to leave alone, 2 for totem pole `buschk` — 0 to not check, 1 for detach on error, 2 for warn on error `hostid` — The SCSI ID to use for the host adapters `verb` — 0 for minimal verbosity, 1 for normal, 2 for excessive `debug` — Set bits to enable debugging `settle` — Settle delay in seconds. Default 3 `nvram` — Option currently not used `excl` — List ioport addresses here to prevent controllers from being attached `safe` — Set other settings to a "safe mode"

Table 30.1. Storage Module Parameters

30.5. Ethernet Parameters

Important

Most modern Ethernet-based network interface cards (NICs), do not require module parameters to alter settings. Instead, they can be configured using **`ethtool`** or **`mii-tool`**. Only after these tools fail to work should module parameters be adjusted. Module paramaters can be viewed using the **`modinfo`** command.

Note

For information about using these tools, consult the man pages for **`ethtool`**, **`mii-tool`**, and **`modinfo`**.

Hardware	Module	Parameters
3Com EtherLink PCI III/XL Vortex (3c590, 3c592, 3c595, 3c597) Boomerang (3c900, 3c905, 3c595)	**`3c59x.ko`**	*`debug`* — 3c59x debug level (0-6) *`options`* — 3c59x: Bits 0-3: media type, bit 4: bus mastering, bit 9: full duplex *`global_options`* — 3c59x: same as options, but applies to all NICs if options is unset *`full_duplex`* — 3c59x full duplex setting(s) (1) *`global_full_duplex`* — 3c59x: same as full_duplex, but applies to all NICs if full_duplex is unset *`hw_checksums`* — 3c59x Hardware checksum checking by adapter(s) (0-1) *`flow_ctrl`* — 3c59x 802.3x flow control usage (PAUSE only) (0-1) *`enable_wol`* — 3c59x: Turn on Wake-on-LAN for adapter(s) (0-1) *`global_enable_wol`* — 3c59x: same as enable_wol, but applies to all NICs if enable_wol is unset

Hardware	Module	Parameters
		rx_copybreak — 3c59x copy breakpoint for copy-only-tiny-frames *max_interrupt_work* — 3c59x maximum events handled per interrupt *compaq_ioaddr* — 3c59x PCI I/O base address (Compaq BIOS problem workaround) *compaq_irq* — 3c59x PCI IRQ number (Compaq BIOS problem workaround) *compaq_device_id* — 3c59x PCI device ID (Compaq BIOS problem workaround) *watchdog* — 3c59x transmit timeout in milliseconds *global_use_mmio* — 3c59x: same as use_mmio, but applies to all NICs if options is unset *use_mmio* — 3c59x: use memory-mapped PCI I/O resource (0-1)
RTL8139, SMC EZ Card Fast Ethernet, RealTek cards using RTL8129, or RTL8139 Fast Ethernet chipsets	**8139too.ko**	
Broadcom 4400 10/100 PCI ethernet driver	**b44.ko**	*b44_debug* — B44 bitmapped debugging message enable value
Broadcom NetXtreme II BCM5706/5708 Driver	**bnx2.ko**	*disable_msi* — Disable Message Signaled Interrupt (MSI)
Intel Ether Express/100 driver	**e100.ko**	*debug* — Debug level (0=none,...,16=all) *eeprom_bad_csum_allow* — Allow bad eeprom checksums
Intel EtherExpress/1000 Gigabit	**e1000.ko**	*TxDescriptors* — Number of transmit descriptors *RxDescriptors* — Number of receive descriptors

Hardware	Module	Parameters
		Speed — Speed setting *Duplex* — Duplex setting *AutoNeg* — Advertised auto-negotiation setting *FlowControl* — Flow Control setting *XsumRX* — Disable or enable Receive Checksum offload *TxIntDelay* — Transmit Interrupt Delay *TxAbsIntDelay* — Transmit Absolute Interrupt Delay *RxIntDelay* — Receive Interrupt Delay *RxAbsIntDelay* — Receive Absolute Interrupt Delay *InterruptThrottleRate* — Interrupt Throttling Rate *SmartPowerDownEnable* — Enable PHY smart power down *KumeranLockLoss* — Enable Kumeran lock loss workaround
Myricom 10G driver (10GbE)	**myri10ge.ko**	*myri10ge_fw_name* — Firmware image name *myri10ge_ecrc_enable* — Enable Extended CRC on PCI-E *myri10ge_max_intr_slots* — Interrupt queue slots *myri10ge_small_bytes* — Threshold of small packets *myri10ge_msi* — Enable Message Signalled Interrupts *myri10ge_intr_coal_delay* — Interrupt coalescing delay *myri10ge_flow_control* — Pause parameter

Hardware	Module	Parameters
		myri10ge_deassert_wait — Wait when deasserting legacy interrupts *myri10ge_force_firmware* — Force firmware to assume aligned completions *myri10ge_skb_cross_4k* — Can a small skb cross a 4KB boundary? *myri10ge_initial_mtu* — Initial MTU *myri10ge_napi_weight* — Set NAPI weight *myri10ge_watchdog_timeout* — Set watchdog timeout *myri10ge_max_irq_loops* — Set stuck legacy IRQ detection threshold
NatSemi DP83815 Fast Ethernet	**natsemi.ko**	*mtu* — DP8381x MTU (all boards) *debug* — DP8381x default debug level *rx_copybreak* — DP8381x copy breakpoint for copy-only-tiny-frames *options* — DP8381x: Bits 0-3: media type, bit 17: full duplex *full_duplex* — DP8381x full duplex setting(s) (1)
AMD PCnet32 and AMD PCnetPCI	**pcnet32.ko**	
PCnet32 and PCnetPCI	**pcnet32.ko**	*debug* — pcnet32 debug level *max_interrupt_work* — pcnet32 maximum events handled per interrupt *rx_copybreak* — pcnet32 copy breakpoint for copy-only-tiny-frames

Hardware	Module	Parameters
		tx_start_pt — pcnet32 transmit start point (0-3) *pcnet32vlb* — pcnet32 Vesa local bus (VLB) support (0/1) *options* — pcnet32 initial option setting(s) (0-15) *full_duplex* — pcnet32 full duplex setting(s) (1) *homepna* — pcnet32 mode for 79C978 cards (1 for HomePNA, 0 for Ethernet, default Ethernet
RealTek RTL-8169 Gigabit Ethernet driver	**r8169.ko**	*media* — force phy operation. Deprecated by ethtool (8). *rx_copybreak* — Copy breakpoint for copy-only-tiny-frames *use_dac* — Enable PCI DAC. Unsafe on 32 bit PCI slot. *debug* — Debug verbosity level (0=none, ..., 16=all)
Neterion Xframe 10GbE Server Adapter	**s2io.ko**	
SIS 900/701G PCI Fast Ethernet	**sis900.ko**	*multicast_filter_limit* — SiS 900/7016 maximum number of filtered multicast addresses *max_interrupt_work* — SiS 900/7016 maximum events handled per interrupt *sis900_debug* — SiS 900/7016 bitmapped debugging message level
Adaptec Starfire Ethernet driver	**starfire.ko**	*max_interrupt_work* — Maximum events handled per interrupt *mtu* — MTU (all boards) *debug* — Debug level (0-6) *rx_copybreak* — Copy breakpoint for copy-only-tiny-frames

Hardware	Module	Parameters
		intr_latency — Maximum interrupt latency, in microseconds *small_frames* — Maximum size of receive frames that bypass interrupt latency (0,64,128,256,512) *options* — Deprecated: Bits 0-3: media type, bit 17: full duplex *full_duplex* — Deprecated: Forced full-duplex setting (0/1) *enable_hw_cksum* — Enable/disable hardware cksum support (0/1)
Broadcom Tigon3	**tg3.ko**	*tg3_debug* — Tigon3 bitmapped debugging message enable value
ThunderLAN PCI	**tlan.ko**	*aui* — ThunderLAN use AUI port(s) (0-1) *duplex* — ThunderLAN duplex setting(s) (0-default, 1-half, 2-full) *speed* — ThunderLAN port speen setting(s) (0,10,100) *debug* — ThunderLAN debug mask *bbuf* — ThunderLAN use big buffer (0-1)
Digital 21x4x Tulip PCI Ethernet cards SMC EtherPower 10 PCI(8432T/8432BT) SMC EtherPower 10/100 PCI(9332DST) DEC EtherWorks 100/10 PCI(DE500-XA) DEC EtherWorks 10 PCI(DE450) DEC QSILVER's, Znyx 312 etherarray Allied Telesis LA100PCI-T Danpex EN-9400, Cogent EM110	**tulip.ko**	*ioio_port*

Hardware	Module	Parameters
VIA Rhine PCI Fast Ethernet cards with either the VIA VT86c100A Rhine-II PCI or 3043 Rhine-I D-Link DFE-930-TX PCI 10/100	**`via-rhine.ko`**	*`max_interrupt_work`* — VIA Rhine maximum events handled per interrupt *`debug`* — VIA Rhine debug level (0-7) *`rx_copybreak`* — VIA Rhine copy breakpoint for copy-only-tiny-frames *`avoid_D3`* — Avoid power state D3 (work-around for broken BIOSes)

Table 30.2. Ethernet Module Parameters

30.5.1. Using Multiple Ethernet Cards

It is possible to use multiple Ethernet cards on a single machine. For each card there must be an **`alias`** and, possibly, **`options`** lines for each card in **`/etc/modprobe.conf`**.

For additional information about using multiple Ethernet cards, refer to the *Linux Ethernet-HOWTO* online at *http://www.redhat.com/mirrors/LDP/HOWTO/Ethernet-HOWTO.html*.

30.5.2. The Channel Bonding Module

Fedora allows administrators to bind NICs together into a single channel using the **`bonding`** kernel module and a special network interface, called a *channel bonding interface*. Channel bonding enables two or more network interfaces to act as one, simultaneously increasing the bandwidth and providing redundancy.

To channel bond multiple network interfaces, the administrator must perform the following steps:

1. Add the following line to **`/etc/modprobe.conf`**:

   ```
   alias bond<N> bonding
   ```

   Replace <*N*> with the interface number, such as **`0`**. For each configured channel bonding interface, there must be a corresponding entry in **`/etc/modprobe.conf`**.

2. Configure a channel bonding interface as outlined in *Section 4.2.3, “Channel Bonding Interfaces”*.

3. To enhance performance, adjust available module options to ascertain what combination works best. Pay particular attention to the **`miimon`** or **`arp_interval`** and the **`arp_ip_target`** parameters. Refer to *Section 30.5.2.1, “**`bonding`** Module Directives”* for a listing of available options.

4. After testing, place preferred module options in **`/etc/modprobe.conf`**.

30.5.2.1. bonding Module Directives

Before finalizing the settings for the **`bonding`** module, it is a good idea to test which settings work best. To do this, open a shell prompt as root and type:

```
tail -f /var/log/messages
```

Open another shell prompt and use the **/sbin/insmod** command to load the **bonding** module with different parameters while observing the kernel messages for errors.

The **/sbin/insmod** command is issued in the following format:

```
/sbin/insmod bond<N><parameter=value>
```

Replace <*N*> with the number for the bonding interface. Replace <*parameter=value*> with a space separated list of desired parameters for the interface.

Once satisfied with the performance of the bonding interface (and verifying that there are no errors), add the appropriate **bonding** module parameters to **ifcfg-bond<*N*>** (where **<*N*>** is the number of the interface). These parameters are set as **BONDING_OPTS="<*parameters*>"**. For more information, refer to *Section 4.2.3, “Channel Bonding Interfaces”*.

The following is a list of available parameters for the **bonding** module:

- **mode=** — Specifies one of four policies allowed for the **bonding** module. Acceptable values for this parameter are:

  0 — Sets a round-robin policy for fault tolerance and load balancing. Transmissions are received and sent out sequentially on each bonded slave interface beginning with the first one available.

  1 — Sets an active-backup policy for fault tolerance. Transmissions are received and sent out via the first available bonded slave interface. Another bonded slave interface is only used if the active bonded slave interface fails.

  2 — Sets an XOR (exclusive-or) policy for fault tolerance and load balancing. Using this method, the interface matches up the incoming request's MAC address with the MAC address for one of the slave NICs. Once this link is established, transmissions are sent out sequentially beginning with the first available interface.

  3 — Sets a broadcast policy for fault tolerance. All transmissions are sent on all slave interfaces.

  4 — Sets an IEEE 802.3ad dynamic link aggregation policy. Creates aggregation groups that share the same speed and duplex settings. Transmits and receives on all slaves in the active aggregator. Requires a switch that is 802.3ad compliant.

  5 — Sets a Transmit Load Balancing (TLB) policy for fault tolerance and load balancing. The outgoing traffic is distributed according to the current load on each slave interface. Incoming traffic is received by the current slave. If the receiving slave fails, another slave takes over the MAC address of the failed slave.

  6 — Sets an Active Load Balancing (ALB) policy for fault tolerance and load balancing. Includes transmit and receive load balancing for IPV4 traffic. Receive load balancing is achieved through ARP negotiation.

- **miimon=** — Specifies (in milliseconds) how often MII link monitoring occurs. This is useful if high availability is required because MII is used to verify that the NIC is active. To verify that the driver for a particular NIC supports the MII tool, type the following command as root:

```
ethtool <interface-name> | grep "Link detected:"
```

In this command, replace *<interface-name>* with the name of the device interface, such as **eth0**, not the **bond** interface. If MII is supported, the command returns:

```
Link detected: yes
```

If using a bonded interface for high availability, the module for each NIC must support MII.

Setting the value to **0** (the default), turns this feature off. When configuring this setting, a good starting point for this parameter is **100**.

- **downdelay=** — Specifies (in milliseconds) how long to wait after link failure before disabling the link. The value must be a multiple of the value specified in the **miimon** parameter. The value is set to **0** by default, which disables it.

- **updelay=** — Specifies (in milliseconds) how long to wait before enabling a link. The value must be a multiple of the value specified in the **miimon** parameter. The value is set to **0** by default, which disables it.

- **arp_interval=** — Specifies (in milliseconds) how often ARP monitoring occurs.

  If using this setting while in **mode0** or **2** (the two load-balancing modes), the network switch must be configured to distribute packets evenly across the NICs. For more information on how to accomplish this, refer to

  /usr/share/doc/kernel-doc-<*kernel-version*>/Documentation/networking/bonding.txt

  The value is set to **0** by default, which disables it.

- **arp_ip_target=** — Specifies the target IP address of ARP requests when the **arp_interval** parameter is enabled. Up to 16 IP addresses can be specified in a comma separated list.

- **arp_validate=** — validate source/distribution of ARP probes; default is **none**. Other valid values are **active**, **backup**, and **all**.

- **lacp_rate=** — Specifies the rate at which link partners should transmit LACPDU packets in 802.3ad mode. Possible values are:

  - **slow** or **0** — Default setting. This specifies that partners should transmit LACPDUs every 30 seconds.

  - **fast** or **1** — Specifies that partners should transmit LACPDUs every 1 second.

- **primary=** — Specifies the interface name, such as **eth0**, of the primary device. The **primary** device is the first of the bonding interfaces to be used and is not abandoned unless it fails. This setting is particularly useful when one NIC in the bonding interface is faster and, therefore, able to handle a bigger load.

  This setting is only valid when the bonding interface is in active-backup mode. Refer to

  /usr/share/doc/kernel-doc-<*kernel-version*>/Documentation/networking/bonding.txt

for more information.

- **`use_carrier=`** — Specifies whether or not **`miimon`** should use MII/ETHTOOL ioctls or **`netif_carrier_ok()`** to determine the link state. The **`netif_carrier_ok()`** relies on the device driver to maintains its state with **`netif_carrier_on/off`**; most device drivers support this function.

  The MII/ETHROOL ioctls tools utilize a deprecated calling sequence within the kernel. However, this is still configurable in case your device driver does not support **`netif_carrier_on/off`**.

  Valid values are:

  - **`1`** — Default setting. Enables the use of **`netif_carrier_ok()`**.

  - **`0`** — Enables the use of MII/ETHTOOL ioctls.

  ### Tip

  If **`bonding`** insists that the link is up when it should not be, it is possible that your network device driver does not support **`netif_carrier_on/off`**.

- **`xmit_hash_policy`** — Selects the transmit hash policy used for slave selection in **`balance-xor`** and 802.3ad modes. Possible values are:

  - **`0`** or **`layer2`** — Default setting. This option uses the XOR of hardware MAC addresses to generate the hash. The formula used is:

    ```
    (<source> <MAC> <XOR> <destination> <MAC>) <modulo slave count>
    ```

    This algorithm will place all traffic to a particular network peer on the same slave, and is 802.3ad compliant.

  - **`1`** or **`layer3+4`** — Uses upper layer protocol information (when available) to generate the hash. This allows for traffic to a particular network peer to span multiple slaves, although a single connection will not span multiple slaves.

    The formula for unfragmented TCP and UDP packets used is:

    ```
    ((<source> <port> <XOR> <destination>) <XOR>
     ((<source> <IP> <XOR> <destination> <IP>) AND 0xffff)
      <modulo slave count>
    ```

    For fragmented TCP or UDP packets and all other IP protocol traffic, the source and destination port information is omitted. For non-IP traffic, the formula is the same as the **`layer2`** transmit hash policy.

    This policy intends to mimic the behavior of certain switches; particularly, Cisco switches with PFC2 as well as some Foundry and IBM products.

The algorithm used by this policy is not 802.3ad compliant.

> **Important**
>
> It is essential that either the **`arp_interval`** and **`arp_ip_target`** or **`miimon`** parameters are specified. Failure to due so can cause degradation of network performance in the event a link fails.

Refer to the following file for more information (note that you must have the **`kernel-doc`** package installed to read this file):

```
/usr/share/doc/kernel-doc-<kernel-version>/Documentation/networking/
bonding.txt
```

for detailed instructions regarding bonding interfaces.

30.6. Additional Resources

For more information on kernel modules and their utilities, refer to the following resources.

30.6.1. Installed Documentation

- **`lsmod`** man page — description and explanation of its output.
- **`insmod`** man page — description and list of command line options.
- **`modprobe`** man page — description and list of command line options.
- **`rmmod`** man page — description and list of command line options.
- **`modinfo`** man page — description and list of command line options.
- **`/usr/share/doc/kernel-doc-<version>/Documentation/kbuild/modules.txt`** — how to compile and use kernel modules. Note you must have the **`kernel-doc`** package installed to read this file.

30.6.2. Useful Websites

- *http://tldp.org/HOWTO/Module-HOWTO/* — *Linux Loadable Kernel Module HOWTO* from the Linux Documentation Project.

Appendix A. Revision History

Revision 7	Mon Sep 14 2009	Douglas Silas *dhensley@redhat.com*

Resolve BZ#214326: Server Security FTP Banner instructions: questions re: vsftpd.conf
Resolve BZ#466239: insert line into Postfix config file to allow connecting remotely
Resolve BZ#510847: missing footnotes in PDF output
Resolve BZ#510851: rewrite /proc/cmdline minor section to make more sense
Resolve BZ#523070: various minor fixes; --redhatprovides to rpm -q --whatprovides

Revision 3	Mon Sep 14 2009	Douglas Silas *dhensley@redhat.com*

Resolve BZ#214326: Server Security FTP Banner instructions: questions re: vsftpd.conf
Resolve BZ#466239: insert line into Postfix config file to allow connecting remotely
Resolve BZ#499053: path for racoon daemon is /usr/sbin/racoon, not /sbin/racoon
Resolve BZ#510851: rewrite /proc/cmdline minor section to make more sense
Update all Deployment Guide support files (Book_Info, etc.) to recent versions

Revision 2	Wed Sep 02 2009	Douglas Silas *dhensley@redhat.com*

Resolve BZ#492539: "This directive is useful..." to "This directive must be used in machines containing more than one NIC to ensure..."
Resolve BZ#453071: incorrect tag use led to config files and other screen elements being displayed on single lines.
Resolve BZ#466239: line omitted from Postfix configuration meant connecting remotely failed.
Resolving other MODIFIED BZs (fixed previously): 468483, 480324, 481246, 481247, 438823, 454841, 485187, 429989, 452065, 453466

Revision 1	Fri Jul 25 2009	Douglas Silas *dhensley@redhat.com*

Initialization of the Fedora 12 Deployment Guide

Index