Fedora Draft Documentation System Administrator's Guide
Deployment, Configuration, and Administration of Fedora 20 Edition 20.0.1

Author    Jaromír Hradílek   jhradilek@redhat.com
Author    Douglas Silas      silas@redhat.com
Author    Martin Prpič       mprpic@redhat.com
Author    Stephen Wadeley    swadeley@redhat.com
Author    Eliška Slobodová  eslobodo@redhat.com
Author    Tomáš Čapek         tcapek@redhat.com
Author    Petr Kovář         pkovar@redhat.com
Author    Miroslav Svoboda   msvoboda@redhat.com

Copyright © 2014 Red Hat, Inc. and others.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at http://creativecommons.org/licenses/by-sa/3.0/. The original authors of this document, and Red Hat, designate the Fedora Project as the “Attribution Party” for purposes of CC-BY-SA. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

For guidelines on the permitted uses of the Fedora trademarks, refer to https://fedoraproject.org/wiki/Legal:Trademark_guidelines.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

All other trademarks are the property of their respective owners.

The System Administrator's Guide documents relevant information regarding the deployment, configuration, and administration of Fedora 20. It is oriented towards system administrators with a basic understanding of the system.
# Preface

1. Target Audience ................................................................. xv
2. How to Read this Book ...................................................... xv
3. Document Conventions ........................................................ xvi
   3.1. Typographic Conventions ........................................... xviii
   3.2. Pull-quote Conventions .............................................. xix
   3.3. Notes and Warnings .................................................. xix
4. We Need Feedback! ............................................................. xx
5. Acknowledgments ................................................................ xx

# I. Basic System Configuration

1. Opening Graphical Applications ........................................ 3
   1.1. Opening graphical applications from the command line .... 3
   1.2. Launching Applications with Alt+F2 ............................ 4
   1.3. Launching applications from the Desktop Menu .......... 7
      1.3.1. Using GNOME menus .................................. 7
      1.3.2. Using KDE menus ...................................... 9
      1.3.3. Using menus in LXDE, MATE, and XFCE .......... 11

2. Configuring the Language and Keyboard ......................... 15
   2.1. Changing the Language ............................................. 15
   2.2. Changing the Date, Time, and Numeric Format .......... 16
   2.3. Changing the Keyboard Layout ................................. 18
   2.4. Viewing the Current Configuration ............................. 20

3. Configuring the Date and Time ........................................... 21
   3.1. Using the Date and Time Configuration Tool ............... 21
   3.2. Using the Command Line Tools .................................. 22
      3.2.1. Changing the Date ..................................... 22
      3.2.2. Changing the Time ..................................... 22
      3.2.3. Configuring the Network Time Protocol .......... 22
   3.3. Additional Resources .............................................. 22
      3.3.1. Installed Documentation ............................... 23

4. Managing Users and Groups .............................................. 25
   4.1. Introduction to Users and Groups ............................... 25
      4.1.1. User Private Groups ................................... 25
      4.1.2. Shadow Passwords ...................................... 25
   4.2. Using the User Accounts Tool ................................... 26
      4.2.1. Configuring an Account ................................. 27
      4.2.2. Adding a New User ..................................... 28
      4.2.3. Removing a User ....................................... 28
   4.3. Using the User Manager Tool .................................... 29
      4.3.1. Viewing Users and Groups ......................... 29
      4.3.2. Adding a New User .................................... 30
      4.3.3. Adding a New Group .................................. 31
      4.3.4. Modifying User Properties .......................... 31
      4.3.5. Modifying Group Properties ....................... 32
   4.4. Using Command Line Tools ...................................... 33
      4.4.1. Adding a New User .................................... 33
      4.4.2. Adding a New Group .................................. 36
      4.4.3. Enabling Password Aging ............................. 37
      4.4.4. Adding Automatic Logouts ......................... 38
      4.4.5. Creating Group Directories ......................... 39
8. Configuring Authentication

8.1. Configuring System Authentication ................................................................. 93
  8.1.1. Launching the Authentication Configuration Tool UI ................................ 93
  8.1.2. Selecting the Identity Store for Authentication ....................................... 94
  8.1.3. Configuring Alternative Authentication Features ................................. 99
  8.1.4. Setting Password Options .................................................................... 101
  8.1.5. Configuring Authentication from the Command Line .......................... 103
  8.1.6. Using Custom Home Directories ......................................................... 106

8.2. Using and Caching Credentials with SSSD .................................................. 107
  8.2.1. About the sssd.conf File ...................................................................... 107
  8.2.2. Starting and Stopping SSSD ................................................................. 107
  8.2.3. Configuring SSSD to Work with System Services .............................. 108
  8.2.4. Creating Domains ............................................................................. 113
  8.2.5. Configuring Access Control for SSSD Domains .................................. 129
  8.2.6. Configuring Domain Failover .............................................................. 130
  8.2.7. Managing the SSSD Cache ................................................................. 131
  8.2.8. Configuring OpenSSH to Check SSSD for Cached Keys (TECH PREVIEW) ................................................................................. 133
  8.2.9. Using NSCD with SSSD ................................................................. 135
  8.2.10. Troubleshooting SSSD .................................................................... 136

9. OpenSSH .............................................................................................................. 141

9.1. The SSH Protocol ......................................................................................... 141
  9.1.1. Why Use SSH? ................................................................................. 141
  9.1.2. Main Features ................................................................................... 142
  9.1.3. Protocol Versions ............................................................................. 142
  9.1.4. Event Sequence of an SSH Connection ........................................... 142

9.2. An OpenSSH Configuration ............................................................................ 144
  9.2.1. Configuration Files .......................................................................... 144
  9.2.2. Starting an OpenSSH Server ............................................................. 146
  9.2.3. Requiring SSH for Remote Connections ......................................... 147
  9.2.4. Using a Key-Based Authentication .................................................... 147

9.3. OpenSSH Clients .......................................................................................... 150
  9.3.1. Using the ssh Utility ........................................................................ 151
  9.3.2. Using the scp Utility ...................................................................... 152
  9.3.3. Using the sftp Utility ...................................................................... 152

9.4. More Than a Secure Shell ............................................................................ 153
  9.4.1. X11 Forwarding .............................................................................. 153
  9.4.2. Port Forwarding .............................................................................. 154

9.5. Additional Resources .................................................................................. 155
  9.5.1. Installed Documentation .................................................................. 155
  9.5.2. Useful Websites ............................................................................. 155

IV. Servers .............................................................................................................. 157

10. Web Servers ...................................................................................................... 159

10.1. The Apache HTTP Server ........................................................................ 159
  10.1.1. New Features ............................................................................... 159
  10.1.2. Notable Changes .......................................................................... 159
  10.1.3. Updating the Configuration ............................................................ 159
  10.1.4. Running the httpd Service ............................................................ 160
11. Mail Servers ........................................ 201
11.1. Email Protocols ........................................ 201
  11.1.1. Mail Transport Protocols .......................... 201
  11.1.2. Mail Access Protocols .............................. 202
11.2. Email Program Classifications ......................... 204
  11.2.1. Mail Transport Agent ............................... 204
  11.2.2. Mail Delivery Agent ................................. 205
  11.2.3. Mail User Agent ................................. 205
11.3. Mail Transport Agents .................................. 205
  11.3.1. Postfix ........................................ 205
  11.3.2. Sendmail ........................................... 207
  11.3.3. Fetchmail ............................................ 212
  11.3.4. Mail Transport Agent (MTA) Configuration .......... 216
11.4. Mail Delivery Agents ................................... 217
  11.4.1. Procmail Configuration .......................... 217
  11.4.2. Procmail Recipes .................................. 218
11.5. Mail User Agents ....................................... 223
  11.5.1. Securing Communication .......................... 223
11.6. Additional Resources .................................. 225
  11.6.1. Installed Documentation ........................... 225
  11.6.2. Useful Websites ..................................... 226
  11.6.3. Related Books ........................................ 227

12. Directory Servers ........................................ 229
12.1. OpenLDAP .................................................. 229
  12.1.1. Introduction to LDAP ............................... 229
  12.1.2. Installing the OpenLDAP Suite ..................... 231
  12.1.3. Configuring an OpenLDAP Server .................. 233
  12.1.4. Running an OpenLDAP Server ........................ 238
  12.1.5. Configuring a System to Authenticate Using OpenLDAP ............................ 239
  12.1.6. Additional Resources ................................. 240

13. File and Print Servers ........................................ 243
13.1. Samba ....................................................... 243
  13.1.1. Introduction to Samba ............................... 243
  13.1.2. Samba Daemons and Related Services ............... 244
  13.1.3. Connecting to a Samba Share ........................ 245
  13.1.4. Configuring a Samba Server ....................... 247
  13.1.5. Starting and Stopping Samba ....................... 248
  13.1.6. Samba Server Types and the smb.conf File .............. 249
  13.1.7. Samba Security Modes .............................. 256
  13.1.8. Samba Account Information Databases ............... 258
  13.1.9. Samba Network Browsing .......................... 259
  13.1.10. Samba with CUPS Printing Support ................... 260
  13.1.11. Samba Distribution Programs ...................... 260
  13.1.12. Additional Resources ............................... 265
13.2. FTP ......................................................... 267
  13.2.1. The File Transfer Protocol ......................... 267
  13.2.2. FTP Servers ......................................... 268
## 13. Printer Configuration

13.3. Printer Configuration ............................................................... 280
   13.3.1. Starting the Printer Configuration Tool ........................................................ 280
   13.3.2. Starting Printer Setup ........................................................................... 280
   13.3.3. Adding a Local Printer ........................................................................ 281
   13.3.4. Adding an AppSocket/HP JetDirect printer ........................................... 282
   13.3.5. Adding an IPP Printer .......................................................................... 283
   13.3.6. Adding an LPD/LPR Host or Printer .................................................... 284
   13.3.7. Adding a Samba (SMB) printer ............................................................ 285
   13.3.8. Selecting the Printer Model and Finishing ........................................... 287
   13.3.9. Printing a test page .......................................................................... 290
   13.3.10. Modifying Existing Printers ............................................................ 291
   13.3.11. Additional Resources .................................................................... 297

## 14. Configuring NTP Using the chrony Suite 299

14.1. Introduction to the chrony Suite ........................................................ 299
   14.1.1. Differences Between ntpd and chronyd ................................................ 299
   14.1.2. Choosing Between NTP Daemons ...................................................... 300
14.2. Understanding chrony and Its Configuration ........................................... 300
   14.2.1. Understanding chrony ....................................................................... 300
   14.2.2. Understanding chronyc ..................................................................... 300
   14.2.3. Understanding the chrony Configuration Commands ....................... 301
   14.2.4. Security with chronyc ..................................................................... 304
14.3. Using chrony ............................................................................... 305
   14.3.1. Checking if chrony is Installed .......................................................... 305
   14.3.2. Installing chrony ............................................................................ 306
   14.3.3. Checking the Status of chrony ........................................................... 306
   14.3.4. Starting chronyd ............................................................................ 306
   14.3.5. Stopping chronyd ........................................................................... 306
   14.3.6. Checking if chrony is Synchronized .................................................. 306
   14.3.7. Manually Adjusting the System Clock .............................................. 310
14.4. Setting Up chrony for Different Environments ..................................... 310
   14.4.1. Setting Up chrony for a System Which is Infrequently Connected ........ 310
   14.4.2. Setting Up chrony for a System in an Isolated Network ....................... 311
14.5. Using chronyc ............................................................................. 312
   14.5.1. Using chronyc to Control chronyd ..................................................... 312
   14.5.2. Using chronyc for Remote Administration ......................................... 312
14.6. Additional Resources ..................................................................... 313
   14.6.1. Installed Documentation ................................................................. 313
   14.6.2. Useful Websites ............................................................................. 313

## 15. Configuring NTP Using ntpd 315

15.1. Introduction to NTP ................................................................. 315
15.2. NTP Strata ............................................................................... 315
15.3. Understanding NTP ................................................................. 316
15.4. Understanding the Drift File ......................................................... 317
15.5. UTC, Timezones, and DST ........................................................... 317
15.6. Authentication Options for NTP .................................................... 318
15.7. Managing the Time on Virtual Machines ........................................... 318
15.8. Understanding Leap Seconds ......................................................... 318
15.9. Understanding the ntpd Configuration File ........................................ 319
15.10. Understanding the ntpd Sysconfig File ............................................. 320
15.11. Disabling chrony ................................................................. 320
15.12. Checking if the NTP Daemon is Installed ............................... 321
15.13. Installing the NTP Daemon (ntpd) ........................................... 321
15.14. Checking the Status of NTP ................................................... 321
15.15. Configure the Firewall to Allow Incoming NTP Packets .......... 321
  15.15.1. Change the Firewall Settings ............................................ 322
  15.15.2. Open Ports in the Firewall for NTP Packets ...................... 322
15.16. Configure NTP ........................................................................ 322
  15.16.1. Configure Access Control to an NTP Service ..................... 322
  15.16.2. Configure Rate Limiting Access to an NTP Service ............ 323
  15.16.3. Adding a Peer Address .................................................... 324
  15.16.4. Adding a Server Address ................................................ 324
  15.16.5. Adding a Broadcast or Multicast Server Address ............... 324
  15.16.6. Adding a Multicast Client Address ..................................... 325
  15.16.7. Adding a Broadcast Client Address ..................................... 325
  15.16.8. Adding a Multicast Server Address ..................................... 325
  15.16.9. Adding a Multicast Client Address ..................................... 325
  15.16.10. Configuring the Burst Option .......................................... 326
  15.16.11. Configuring the iburst Option ......................................... 326
  15.16.12. Configuring Symmetric Authentication Using a Key ........... 326
  15.16.13. Configuring the Poll Interval .......................................... 326
  15.16.15. Configuring the Time-to-Live for NTP Packets ................. 327
  15.16.16. Configuring the NTP Version to Use ................................. 327
15.17. Configuring the Hardware Clock Update .................................... 327
15.18. Configuring Clock Sources .................................................... 328
15.19. Additional Resources ............................................................ 328
  15.19.1. Installed Documentation .................................................. 328
  15.19.2. Useful Websites ............................................................. 328
15.19.2.1. Open Ports in the Firewall for NTP Packets ...................... 322
15.19.2.2. Useful Websites ............................................................. 328
15.19.2.3. Installed Documentation .................................................. 328
16. Configuring PTP Using ptp4l ........................................................... 329
  16.1. Introduction to PTP .............................................................. 329
    16.1.1. Understanding PTP ....................................................... 329
    16.1.2. Advantages of PTP ....................................................... 331
  16.2. Using PTP ............................................................................ 331
    16.2.1. Checking for Driver and Hardware Support ....................... 331
    16.2.2. Installing PTP .............................................................. 332
    16.2.3. Starting ptp4l .............................................................. 332
  16.3. Specifying a Configuration File .............................................. 333
  16.4. Using the PTP Management Client ........................................ 334
  16.5. Synchronizing the Clocks ...................................................... 335
  16.6. Verifying Time Synchronization ............................................ 335
  16.7. Serving PTP Time with NTP ................................................ 337
  16.8. Serving NTP Time with PTP ................................................. 337
  16.9. Improving Accuracy ............................................................. 338
  16.10. Additional Resources .......................................................... 338
    16.10.1. Installed Documentation ............................................... 338
    16.10.2. Useful Websites .......................................................... 338

V. Monitoring and Automation .......................................................... 339

17. System Monitoring Tools ............................................................. 341
  17.1. Viewing System Processes .................................................... 341
    17.1.1. Using the ps Command ................................................... 341
19.2.4. Additional Command Line Options ......................................................  395
19.2.5. Controlling Access to At and Batch ..................................................  395
19.2.6. Starting and Stopping the Service ....................................................  395
19.3. Additional Resources ..........................................................................  396
   19.3.1. Installed Documentation ...............................................................  396

20. Automatic Bug Reporting Tool (ABRT) ..................................................... 397
   20.1. Overview ..........................................................................................  397
   20.2. Installing ABRT and Starting its Services ........................................  397
   20.3. Running ABRT .................................................................................. 399
       20.3.1. Using the Graphical User Interface ............................................. 400
       20.3.2. Using the Command Line Interface .......................................... 406
   20.4. Configuring ABRT ............................................................................ 409
       20.4.1. ABRT Events ............................................................................ 410
       20.4.2. Standard ABRT Installation Supported Events ....................... 411
       20.4.3. Event Configuration in ABRT GUI .......................................... 412
       20.4.4. ABRT Specific Configuration .................................................... 414
   20.5. Configuring Centralized Crash Collection ........................................ 417
       20.5.1. Configuration Steps Required on a Dedicated System ............... 417
       20.5.2. Configuration Steps Required on a Client System ..................... 418
       20.5.3. Saving Package Information ..................................................... 418
       20.5.4. Testing ABRT’s Crash Detection ............................................... 419

21. OProfile .................................................................................................. 421
   21.1. Overview of Tools ........................................................................... 421
   21.2. Configuring OProfile ....................................................................... 422
       21.2.1. Specifying the Kernel .................................................................. 422
       21.2.2. Setting Events to Monitor ......................................................... 423
       21.2.3. Separating Kernel and User-space Profiles ................................. 425
   21.3. Starting and Stopping OProfile .......................................................... 426
   21.4. Saving Data ..................................................................................... 427
   21.5. Analyzing the Data .......................................................................... 427
       21.5.1. Using opreport .......................................................................... 428
       21.5.2. Using opreport on a Single Executable .................................... 429
       21.5.3. Getting more detailed output on the modules ......................... 430
       21.5.4. Using opannotate ...................................................................... 431
   21.6. Understanding /dev/oprofile/ ............................................................ 432
   21.7. Example Usage ................................................................................ 432
   21.8. OProfile Support for Java ................................................................ 432
       21.8.1. Profiling Java Code ................................................................... 433
   21.9. Graphical Interface ......................................................................... 433
   21.10. OProfile and SystemTap ................................................................. 435
   21.11. Additional Resources .................................................................... 436
       21.11.1. Installed Docs ........................................................................... 436
       21.11.2. Useful Websites ....................................................................... 436

VI. Kernel, Module and Driver Configuration ................................................ 437

22. Manually Upgrading the Kernel ................................................................. 439
   22.1. Overview of Kernel Packages .......................................................... 439
   22.2. Preparing to Upgrade ........................................................................ 440
   22.3. Downloading the Upgraded Kernel .................................................. 441
   22.4. Performing the Upgrade ................................................................... 441
A. RPM

A.1. RPM Design Goals ................................................................. 480
A.2. Using RPM ........................................................................... 480
A.2.1. Finding RPM Packages ....................................................... 481
A.2.2. Installing and Upgrading .................................................... 481
A.2.3. Configuration File Changes ............................................... 484
A.2.4. Uninstalling ................................................................. 484
A.2.5. Freshening ................................................................. 485
A.2.6. Querying ................................................................. 486
A.2.7. Verifying ........................................................................ 487
A.3. Checking a Package's Signature ........................................... 488
A.3.1. Importing Keys ............................................................... 488
A.3.2. Verifying Signature of Packages ....................................... 488
A.4. Practical and Common Examples of RPM Usage ....................... 489
A.5. Additional Resources ............................................................ 491
A.5.1. Installed Documentation ................................................... 491

23. Working with Kernel Modules ................................................. 449

23.1. Listing Currently-Loaded Modules ........................................ 449
23.2. Displaying Information About a Module ................................ 450
23.3. Loading a Module ............................................................... 452
23.4. Unloading a Module ............................................................. 453
23.5. Setting Module Parameters .................................................. 454
23.6. Persistent Module Loading ..................................................... 455
23.7. Specific Kernel Module Capabilities ....................................... 456
23.7.1. Using Multiple Ethernet Cards ......................................... 456
23.7.2. Using Channel Bonding .................................................... 456
23.8. Additional Resources ............................................................ 462
23.8.1. Installed Documentation ................................................... 462
23.8.2. Useful Websites ............................................................... 463

24. The kdump Crash Recovery Service ........................................... 465

24.1. Installing the kdump Service .................................................. 465
24.2. Configuring the kdump Service ............................................... 465
24.2.1. Configuring the kdump at First Boot ................................... 465
24.2.2. Using the Kernel Dump Configuration Utility ....................... 466
24.2.3. Configuring kdump on the Command Line ......................... 470
24.2.4. Testing the Configuration .................................................. 473
24.3. Analyzing the Core Dump ...................................................... 474
24.3.1. Running the crash Utility ................................................... 474
24.3.2. Displaying the Message Buffer ........................................... 475
24.3.3. Displaying a Backtrace ...................................................... 476
24.3.4. Displaying a Process Status .............................................. 476
24.3.5. Displaying Virtual Memory Information .............................. 477
24.3.6. Displaying Open Files ........................................................ 477
24.3.7. Exiting the Utility ............................................................. 478
24.4. Additional Resources ............................................................ 478
24.4.1. Installed Documentation ................................................... 478
24.4.2. Useful Websites ............................................................... 478

24.4.1. Installed Documentation ................................................... 478
24.4.2. Useful Websites ............................................................... 478

24.2.4. Testing the Configuration .................................................. 473
24.3.3. Displaying a Backtrace ...................................................... 476
24.3.5. Displaying Virtual Memory Information .............................. 477
24.3.6. Displaying Open Files ........................................................ 477
24.3.7. Exiting the Utility ............................................................. 478
24.4. Additional Resources ............................................................ 478
24.4.1. Installed Documentation ................................................... 478
24.4.2. Useful Websites ............................................................... 478

A. RPM

A.1. RPM Design Goals ................................................................. 480
A.2. Using RPM ........................................................................... 480
A.2.1. Finding RPM Packages ....................................................... 481
A.2.2. Installing and Upgrading .................................................... 481
A.2.3. Configuration File Changes ............................................... 484
A.2.4. Uninstalling ................................................................. 484
A.2.5. Freshening ................................................................. 485
A.2.6. Querying ................................................................. 486
A.2.7. Verifying ........................................................................ 487
A.3. Checking a Package's Signature ........................................... 488
A.3.1. Importing Keys ............................................................... 488
A.3.2. Verifying Signature of Packages ....................................... 488
A.4. Practical and Common Examples of RPM Usage ....................... 489
A.5. Additional Resources ............................................................ 491
A.5.1. Installed Documentation ................................................... 491
Preface

The System Administrator's Guide contains information on how to customize the Fedora 20 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:

- Installing and managing packages using the graphical PackageKit and command line Yum package managers
- Configuring Apache HTTP Server, Postfix, Sendmail and other enterprise-class servers and software
- Gathering information about your system, including obtaining user-space crash data with the Automatic Bug Reporting Tool, and kernel-space crash data with kdump
- Working with kernel modules and upgrading the kernel

1. Target Audience

The System Administrator's Guide assumes you have a basic understanding of the Fedora operating system. If you need help with the installation of this system, refer to the Fedora 20 Installation Guide.

2. How to Read this Book

This manual is divided into the following main categories:

Part I, "Basic System Configuration"

This part covers basic system administration tasks such as keyboard configuration, date and time configuration, and managing users and groups.

Chapter 2, Configuring the Language and Keyboard covers basic language and keyboard setup. Read this chapter if you need to configure the language of your desktop, change the keyboard layout, or add the keyboard layout indicator to the panel.

Chapter 3, Configuring the Date and Time covers the configuration of the system date and time. Read this chapter if you need to set or change the date and time.

Chapter 4, Managing Users and Groups covers the management of users and groups in a graphical user interface and on the command line. Read this chapter if you need to manage users and groups on your system, or enable password aging.

Part II, "Package Management"

This part describes how to manage software packages on Fedora using both Yum and the PackageKit suite of graphical package management tools.

Chapter 5, Yum describes the Yum package manager. Read this chapter for information how to search, install, update, and uninstall packages on the command line.

Chapter 6, PackageKit describes the PackageKit suite of graphical package management tools. Read this chapter for information how to search, install, update, and uninstall packages using a graphical user interface.

Part III, “Infrastructure Services”

This part provides information on how to configure services and daemons, configure authentication, and enable remote logins.

Chapter 7, Services and Daemons covers the configuration of the services to be run when a system is started, and provides information on how to start, stop, and restart the services on the command line using the systemctl utility.

Chapter 8, Configuring Authentication describes how to configure user information retrieval from Lightweight Directory Access Protocol (LDAP), Network Information Service (NIS), and Winbind user account databases, and provides an introduction to the System Security Services Daemon (SSSD). Read this chapter if you need to configure authentication on your system.

Chapter 9, OpenSSH describes how to enable a remote login via the SSH protocol. It covers the configuration of the sshd service, as well as a basic usage of the ssh, scp, sftp client utilities. Read this chapter if you need a remote access to a machine.

Part IV, “Servers”

This part discusses various topics related to servers such as how to set up a Web server or share files and directories over the network.

Chapter 10, Web Servers focuses on the Apache HTTP Server 2.2, a robust, full-featured open source web server developed by the Apache Software Foundation. Read this chapter if you need to configure a web server on your system.

Chapter 11, Mail Servers reviews modern email protocols in use today, and some of the programs designed to send and receive email, including Postfix, Sendmail, Fetchmail, and Procmail. Read this chapter if you need to configure a mail server on your system.

Chapter 12, Directory Servers covers the installation and configuration of OpenLDAP 2.4, an open source implementation of the LDAPv2 and LDAPv3 protocols. Read this chapter if you need to configure a directory server on your system.

Chapter 13, File and Print Servers guides you through the installation and configuration of Samba, the primary FTP server shipped with Fedora. Additionally, it explains how to use the Printer Configuration tool to configure printers. Read this chapter if you need to configure a file or print server on your system.

Chapter 14, Configuring NTP Using the chrony Suite covers the installation and configuration of the chrony suite, a client and a server for the Network Time Protocol (NTP). Read this chapter if you need to configure the system to synchronize the clock with a remote NTP server, or set up an NTP server on this system.

Chapter 15, Configuring NTP Using ntpd covers the installation and configuration of the NTP daemon, ntpd, for the Network Time Protocol (NTP). Read this chapter if you need to configure the system to synchronize the clock with a remote NTP server, or set up an NTP server on this system, and you prefer not to use the chrony application.

Chapter 16, Configuring PTP Using ptp4l covers the installation and configuration of the Precision Time Protocol application, ptp4l, an application for use with network drivers that support the Precision Network Time Protocol (PTP). Read this chapter if you need to configure the system to synchronize the system clock with a master PTP clock.

Part V, “Monitoring and Automation”

This part describes various tools that allow system administrators to monitor system performance, automate system tasks, and report bugs.
Chapter 17, System Monitoring Tools discusses applications and commands that can be used to retrieve important information about the system. Read this chapter to learn how to gather essential system information.

Chapter 18, Viewing and Managing Log Files describes the configuration of the rsyslog daemon, and explains how to locate, view, and monitor log files. Read this chapter to learn how to work with log files.

Chapter 19, Automating System Tasks provides an overview of the cron, at, and batch utilities. Read this chapter to learn how to use these utilities to perform automated tasks.

Chapter 20, Automatic Bug Reporting Tool (ABRT) concentrates on ABRT, a system service and a set of tools to collect crash data and send a report to the relevant issue tracker. Read this chapter to learn how to use ABRT on your system.

Chapter 21, OProfile covers OProfile, a low overhead, system-wide performance monitoring tool. Read this chapter for information on how to use OProfile on your system.

Part VI, “Kernel, Module and Driver Configuration”
This part covers various tools that assist administrators with kernel customization.

Chapter 22, Manually Upgrading the Kernel provides important information on how to manually update a kernel package using the rpm command instead of yum. Read this chapter if you cannot update a kernel package with the Yum package manager.

Chapter 23, Working with Kernel Modules explains how to display, query, load, and unload kernel modules and their dependencies, and how to set module parameters. Additionally, it covers specific kernel module capabilities such as using multiple Ethernet cards and using channel bonding. Read this chapter if you need to work with kernel modules.

Chapter 24, The kdump Crash Recovery Service explains how to configure, test, and use the kdump service in Fedora, and provides a brief overview of how to analyze the resulting core dump using the crash debugging utility. Read this chapter to learn how to enable kdump on your system.

Appendix A, RPM
This appendix concentrates on the RPM Package Manager (RPM), an open packaging system used by Fedora, and the use of the rpm utility. Read this appendix if you need to use rpm instead of yum.

Appendix B, The X Window System
This appendix covers the configuration of the X Window System, the graphical environment used by Fedora. Read this appendix if you need to adjust the configuration of your X Window System.

3. 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 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.
3.1. 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 keycaps 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 keycap, all presented in mono-spaced bold and all distinguishable thanks to context.

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

Press `Enter` to execute the command.

Press `Ctrl+Alt+F2` to switch to the first virtual terminal. Press `Ctrl+Alt+F1` to return to your X-Windows session.

The first paragraph highlights the particular keycap to press. The second highlights two key combinations (each a set of three keycaps with 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; dialog box text; labeled buttons; check-box and radio button labels; menu titles and sub-menu titles. For example:

Choose `System → Preferences → 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.

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:

```
Publican is a DocBook publishing system.
```

### 3.2. Pull-quote Conventions
Terminal output and source code listings 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 add syntax highlighting 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"));
    }
}
```

### 3.3. Notes and Warnings
Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.
Note

Notes are tips, shortcuts or alternative approaches 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 a box labeled 'Important' will not cause data loss but may cause irritation and frustration.

Warning

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

4. 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: https://bugzilla.redhat.com/enter_bug.cgi?product= Fedora Documentation&component=system-administrator’s-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.

5. Acknowledgments


Section 17.6, “Monitoring Performance with Net-SNMP” is based on an article written by Michael Solberg.

The authors of this book would like to thank the following people for their valuable contributions: Adam Tkáč, Andrew Fitzsimon, Andrius Benokraitis, Brian Cleary Edward Bailey, Garrett LeSage, Jeffrey Fearn, Joe Orton, Joshua Wulf, Karsten Wade, Lucy Ringland, Marcela Mašláňová, Mark Johnson, Michael Behm, Miroslav Lichvár, Radek Vokál, Rahul Kavalapara, Rahul Sundaram, Sandra Moore, Zbyšek Mráz, Jan Včelák, Peter Hutterer, T.C. Hollingsworth, and James Antill, among many others.
Part I. Basic System Configuration

This part covers basic system administration tasks such as keyboard configuration, date and time configuration, installation and initial setup of an NTP server, and managing users and groups.
Opening Graphical Applications

Fedora provides graphical applications in addition to command line utilities for configuring many features. This chapter describes methods for opening **Graphical User Interface**, or **GUI**, applications in various environments.

1.1. Opening graphical applications from the command line

Graphical applications can be launched from a terminal window or console session by simply typing the name of the application.

```
[ fedorauser@localhost ] $ firefox
```

**File names vs Application names**

Programs are opened from the command line using the name of executable file provided in the program's package. An entry in the desktop menu will often be named differently from the file it executes. For example, the GNOME disk management utility appears in the menu as **Disks**, and the file it executes is `/usr/bin/gnome-disks`.

When a program is executed on the command line, the terminal is occupied until the program completes. When a graphical application is executed from the command line, the program's error output, or **STDERR**, is sent to the terminal window. This can be especially useful when troubleshooting.

**Example 1.1. Viewing errors by launching graphical applications from the command line**

```
[ fedorauser@localhost ] $ astromenace-wrapper
AstroMenace 1.3.1 121212
Open XML file: /home/fedorauser/.config/astromenace/amconfig.xml
VFS file was opened /usr/share/astromenace/gamedata.vfs
Vendor     : OpenAL Community
Renderer   : OpenAL Soft
Version    : 1.1 ALSOFT 1.15.1
ALut ver   : 1.1
Font initialized: DATA/FONT/LiberationMono-Bold.ttf
Current Video Mode: 3200x1080 32bit
Xinerama/TwinView detected.
Screen count: 2
Screen #0: (0, 0) x (1920, 1080)
Screen #1: (1920, 0) x (1280, 1024)
Supported resolutions list:
640x480 16bit
640x480 32bit
640x480 0bit
768x480 16bit
<output truncated>
```
To launch a graphical application, but fork the additional output into the background and return the terminal for immediate use, use the shell's job control feature.

[fedorauser@localhost]$ emacs foo.txt &

**Ending a session**

Applications that hold the command line prompt until they complete will close when the terminal session ends, even if they are forked into the background.

GUI programs can also be launched on one TTY and displayed on another by specifying the DISPLAY variable. This can be useful when running multiple graphical sessions, or for troubleshooting problems with a desktop session.

1. Switch to another TTY using the key combination Ctrl-Alt-F2 and log in. Note that consoles are available by default with F2 through F6.

2. Identify the X session you want to target. The DISPLAY variable is always an integer preceded by a colon, and will be :0 in most cases. Check the arguments of the currently running X process to verify the value. The command below shows both the DISPLAY variable as well as the TTY that X is running on, tty1.

   [fedorauser@localhost]$ ps aux|grep /usr/bin/X
   root      1498  7.1  1.0 521396 353984 tty1 Ss+  00:04  66:34 /usr/bin/X :0 vt1 -
   background none -nolisten tcp -auth /var/run/kdm/A:0-22Degc
   root     23874  0.0  0.0 109184   900 pts/21   S+   15:35   0:00 grep --color=auto /usr/bin/X

3. Specify the DISPLAY variable when executing the program.

   [fedorauser@localhost]$ DISPLAY=:0 gnome-shell --replace &

4. Switch back to the TTY the graphical session is running on. Since the example above shows X running on vt1, pressing Ctrl+Alt+F1 will return to the desktop environment.

### 1.2. Launching Applications with Alt+F2

Most desktop environments follow the convention of using the key combination Alt+F2 for opening new applications. Pressing Alt+F2 brings up a prompt for a command to be entered into.

Commands entered into this dialog box function much as they would if entered in a terminal. Applications are known by their file name, and can accept arguments.
Draft

Launching Applications with \textit{Alt}+\textit{F2}

Figure 1.1. Using \textit{Alt}+\textit{F2} with \textbf{GNOME}

Figure 1.2. Using \textit{Alt}+\textit{F2} with \textbf{KDE}
Chapter 1. Opening Graphical Applications

Figure 1.3. Using Alt+F2 with LXDE

Figure 1.4. Using Alt+F2 with MATE
1.3. Launching applications from the Desktop Menu

Applications can also be opened from the menu system provided by the desktop environment in use. While the presentation may vary between desktop environments, the menu entries and their categories are provided by the individual application and standardized by the freedesktop.org Desktop Menu Specification\(^1\). Some desktop environments also provide search functionality in their menu system to allow quick and easy access to applications.

**1.3.1. Using GNOME menus**

The GNOME menu, called the overview, can be accessed by either clicking the Activities button in the top left of the primary display, by moving the mouse past the top left hot corner, or by pressing the Super (Windows) key. The overview presents documents in addition to applications.

Selecting an item from the menu is best accomplished using the search box. Simply bring up the overview, and begin typing the name of the application you want to launch. Pressing enter will launch the highlighted application, or you can use the arrow keys or mouse to choose an alternative.

---

\(^1\) [http://standards.freedesktop.org/menu-spec/menu-spec-latest.html](http://standards.freedesktop.org/menu-spec/menu-spec-latest.html)
Figure 1.6. Using the GNOME search box

The **overview** can also be browsed. The bar on the left, called the **dash**, shows frequently used applications and grid icon. Clicking on the grid icon brings up a grid in the center of the window that displays more frequent applications. The grid will display all available applications if selected using the **All** button at the bottom of the screen.
To learn more about using **GNOME shell**, visit [https://wiki.gnome.org/GnomeShell/CheatSheet](https://wiki.gnome.org/GnomeShell/CheatSheet)

### 1.3.2. Using KDE menus

The KDE menu is opened by clicking the Fedora button at the bottom left corner of the screen. The menu initially displays favorite applications, which can be added to by right clicking any menu entry.
Hovering over the icons in the lower portion of the menu will display applications, file systems, recently used applications, or options for logging out of the system.

![Image of the KDE desktop menu](image)

**Figure 1.8. The KDE desktop menu.**

Search functionality is also available in the KDE menu system. To search for applications, open the menu and begin typing. The menu will display matching entries.
1.3.3. Using menus in LXDE, MATE, and XFCE

Menus in LXDE, MATE, and XFCE have a varied appearance but a very similar structure. They categorize applications, and the contents of a category are displayed by hovering the cursor over the entry. Applications are launched by clicking on an entry.
Chapter 1. Opening Graphical Applications

Figure 1.10. The LXDE menu

Figure 1.11. MATE menu
Figure 1.12. XFCE Menu
Configuring the Language and Keyboard

Fedora 20 is shipped with the Region and Language configuration tool, which allows you to configure keyboard layouts, the language of your desktop environment, and other regional settings. To start the tool, open the System Settings window by selecting Applications → System Tools → System Settings from the Activities menu, and click Region and Language.

2.1. Changing the Language

To configure the language of your desktop, select the Language tab of the Region and Language application. You will be presented with a short list of common languages.

By default, this list only contains a few of the available languages. To add another language, click the + (the plus sign) button below the list. A dialog window appears, allowing you to select the desired language. The input field at the bottom part of the dialog window allows you to reduce the number of displayed items by first few letters part of the language name in it (for example, "slov" for the Slovak language). Once you select a language, click the Select button to confirm your choice.
To choose a particular language from the list, click its name to select it. The changes will take effect the next time you log in to the system.

2.2. Changing the Date, Time, and Numeric Format

To change the default date, time, number, and currency format, select the Formats tab of the Region and Language application. You will be presented with a short list of available formats.
By default, this list only contains a few of the available formats. To add another format, click the plus sign button below the list. A dialog window appears, allowing you to select the desired format according to a region. The input field at the bottom part of the dialog window allows you to reduce the number of displayed items by typing first few letters of the region name in it (for example, "slov" for Slovakia). Once you select a region, click the Select button to confirm your choice.
To choose a particular format from the list, click its name to select it. The changes will take effect the next time you log in to the system.

2.3. Changing the Keyboard Layout

Although the installation program allows a system administrator to configure a keyboard layout during the system installation, the default settings may not always suit your current needs. To change the default keyboard layout, select the **Layouts** tab of the **Region and Language** application. You will be presented with a list of currently enabled layouts.
Figure 2.5. Changing the keyboard layout

To add a layout to the list, click the + (the plus sign) button below the list. A dialog window appears, allowing you to select the desired keyboard layout. The input field at the bottom part of the dialog window allows you to reduce the number of displayed items by typing first few letters of the layout name in it (for example, “slov” for a Slovak layout). Once you select a layout, click the Add button to confirm your choice.

Figure 2.6. Adding a keyboard layout
The first layout in the list is always considered the default. To move a particular layout up or down in the list, select it and click the \(^\wedge\) (the upwards arrow) or \(\downarrow\) (the downwards arrow) buttons respectively. To remove a layout, click the \(-\) (that is, the minus sign) button. Additionally, by selecting an option button on the right side of the window, you can choose if you want to use different keyboard layouts for individual windows, or a single layout for all windows.

When more than one layout is enabled, a keyboard indicator appears on the panel in order to allow you to switch between the layouts.

![Figure 2.7. The keyboard layout indicator](image)

### 2.4. Viewing the Current Configuration

To view the current configuration, select the **System** tab of the **Region and Language** application. You will be presented with a comparison of your own configuration and system-wide settings.

![Figure 2.8. Viewing the current configuration](image)
Configuring the Date and Time

This chapter covers setting the system date and time in Fedora, both manually and using the Network Time Protocol (NTP), as well as setting the time zone. Two methods are covered: setting the date and time using the Date and Time configuration tool, and doing so on the command line.

3.1. Using the Date and Time Configuration Tool

Fedora 20 is shipped with the Date and Time configuration tool, which allows you to change the date and time of the system, to configure the time zone used by the system, and to set up the Network Time Protocol daemon to synchronize the system clock with a time server. To start the tool, either select Applications → System Tools → System Settings from the Activities menu and click the Date and Time icon, or click the time in the panel and select Date and Time Settings from the drop-down menu.

![Figure 3.1. The Date and Time configuration tool](image)

By default, the tool only allows you to review the current settings. This is because only root is allowed to set the system date and time. To unlock the configuration tool for changes, click the Unlock button in the top-right corner of the window, and provide the correct password when prompted.

To change the current time of your system, either configure the system to synchronize it over the network by clicking the Network Time switch, or set it manually by clicking the up and down arrows above and below the numbers. You can also select 24-hour or AM/PM to enable or disable the 24-hour time format.

To change the time zone, either click on the map, or select the region and city from the Region and City drop-down lists.

To change the current date of your system, select a month from the drop-down list below the time, and use the up and down arrows to choose the day and year.

The changes take effect immediately.
3.2. Using the Command Line Tools

Fedora 20 provides command line tools that allow you to configure the date and time both manually and using the NTP protocol.

3.2.1. Changing the Date

To change the system date, type the following at a shell prompt as root:

```bash
date +%D -s YYYY-MM-DD
```

...where YYYY is a four-digit year, MM is a two-digit month, and DD is a two-digit day of the month. For example, to change the date to 2 June 2010, type:

```
~]# date +%D -s 2010-06-02
```

You can verify the current settings by running `date` without any additional argument.

3.2.2. Changing the Time

To change the current time, run the following command as root:

```bash
date +%T -s HH:MM:SS
```

...where HH stands for an hour, MM is a minute, and SS is a second, all typed in a two-digit form. If your system clock is set to use UTC (Coordinated Universal Time), also add the following option:

```bash
date +%T -s HH:MM:SS -u
```

For instance, to set the system clock to 11:26 PM using the UTC, type:

```
~]# date +%T -s 23:26:00 -u
```

You can verify the current settings by running `date` without any additional argument. You should not use this command to set the time if the system clock is being maintained by `chrony`, `ntpd`, or any other similar automated process.

3.2.3. Configuring the Network Time Protocol

Fedora includes the `chrony` suite of programs to automatically adjust the system clock using the Network Time Protocol (NTP). See Chapter 14, Configuring NTP Using the chrony Suite for information on configuring and enabling `chrony`.

It is also possible to use `ntpd` to adjust the system clock using the Network Time Protocol (NTP). See Chapter 15, Configuring NTP Using ntpd for information on configuring `ntpd`.

3.3. Additional Resources

For more information about the date and time configuration, refer to the following resources.
3.3.1. Installed Documentation

- `date(1)` — The manual page for the `date` utility.
Managing Users and Groups

The control of users and groups is a core element of Fedora system administration. This chapter explains how to add, manage, and delete users and groups in the graphical user interface and on the command line, and covers advanced topics, such as enabling password aging or creating group directories.

4.1. Introduction to Users and Groups

While 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 user ID (UID). Likewise, each group is associated with a group ID (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 root, and access permissions can be changed by both the root user and file owner.

Additionally, Fedora supports access control lists (ACLs) for files and directories which allow permissions for specific users outside of the owner to be set. Refer to For more information about this feature, refer to the Access Control Lists[1] chapter of the Storage Administration Guide[2].

4.1.1. User Private Groups

Fedora uses a user private group (UPG) scheme, which makes UNIX groups easier to manage. A user private group is created whenever a new user is added to the system. It has the same name as the user for which it was created and that user is the only member of the user private group.

User private groups 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.

4.1.2. Shadow Passwords

Especially in environments with multiple users, it is very important to use shadow passwords provided by the shadow-utils package to enhance the security of system authentication files. For this reason, the installation program enables shadow passwords by default.

The following is a list of the advantages shadow passwords have over the traditional way of storing passwords on UNIX-based systems:

---

• Shadow passwords improve 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.

• Shadow passwords store information about password aging.

• Shadow passwords allow 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 utilities and commands that do not work without first enabling shadow passwords:

• The `chage` utility.

• The `gpasswd` utility.

• The `usermod` command with the `-e` or `-f` option.

• The `useradd` command with the `-e` or `-f` option.

### 4.2. Using the User Accounts Tool

The User Accounts configuration tool allows you to view, modify, add, and delete local users. To run the tool, select **Applications → System Tools → System Settings** from the Activities menu and click the **User Accounts** icon.

![Figure 4.1. The User Accounts configuration tool](image)

By default, the tool only allows you to change certain settings regarding your account. This is because only the root user is allowed to configure users and groups. To unlock the configuration tool for all kinds of changes, click the Unlock button in the top-right corner of the window, and provide the correct password when prompted.
4.2.1. Configuring an Account

To change the image associated with an account, click the icon next to the account name and either select a picture from the pulldown list, or click Browse for more pictures... to use an image from your local drive.

To change the name associated with an account, click the name next to the icon to edit it.

To change the account type, click the text next to the Account type label. Note that this change requires the configuration tool to be unlocked even if you are changing your own account.

To change the default language for an account, click the text next to the Language label and select a language from the list.

To change the password, click the field next to the Password label. A dialog box appears, allowing you to set the new password. Note that the current password must be provided in order to confirm the change. Once done, click the Change button to save the change.

![Figure 4.2. Changing the password](image)

Password security advice

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.

Finally, to set up automatic login for a particular account, enable the Automatic Login switch. The configuration tool must be unlocked to make this change.
4.2.2. Adding a New User

To add a new user, make sure the configuration tool is unlocked, and click the + button (that is, the plus sign) below the account list. A dialog window appears, allowing you to supply user details.

![Create new account](image)

Figure 4.3. Creating a new account

Take the following steps to create an account:

1. Select an account type from the Account type pulldown list. Available account types are Administrator and Standard (the default option).
2. Fill in the Full name input field to set the name associated with the account. This name will be used by the login manager, and will be displayed on the panel.
3. Either select a suggested username from the Username pulldown list, or fill in the corresponding input field.
4. Click the Create button to confirm the settings.

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, a unique group with the same name as the user is created.

When a new account is created, default configuration files are copied from the /etc/skel/ directory into the new home directory.

4.2.3. Removing a User

To remove a user, make sure the configuration tool is unlocked, select the desired account from the account list, and click the − button (that is, the minus sign) below the account list. A dialog window appears, allowing you to confirm or cancel the change.
To delete files and directories that belong to the user (that is, the home directory, mail spool, and temporary files), click the **Delete Files** button. To keep these files intact and only delete the user account, click **Keep Files**. To abort the deletion, click **Cancel**.

### 4.3. Using the User Manager Tool

The **User Manager** application allows you to view, modify, add, and delete local users and groups in the graphical user interface. To start the application, either select **Applications → Other → Users and Groups** from the Activities menu, or type `system-config-users` at a shell prompt. Note that unless you have superuser privileges, the application will prompt you to authenticate as **root**.

#### 4.3.1. Viewing Users and Groups

The main window of the **User Manager** is divided into two tabs: The **Users** tab provides a list of local users along with additional information about their user ID, primary group, home directory, login shell, and full name. The **Groups** tab provides a list of local groups with information about their group ID and group members.

![User Manager Window](image)

**Figure 4.5. Viewing users and groups**

To find a specific user or group, type the first few letters of the name in the **Search filter** field and either press **Enter**, or click the **Apply filter** button. You can also sort the items according to any of the available columns by clicking the column header.
Fedora reserves user and group IDs below 1000 for system users and groups. By default, the User Manager does not display the system users. To view all users and groups, select Edit → Preferences to open the Preferences dialog box, and clear the Hide system users and groups check box.

### 4.3.2. Adding a New User

To add a new user, click the Add User button. A window as shown in Figure 4.6, “Adding a new user” appears.

![Figure 4.6. Adding a new user](image)

The Add New User dialog box allows you to provide information about the newly created user. In order to create a user, enter the username and full name in the appropriate fields and then type the user's password in the Password and Confirm Password fields. The password must be at least six characters long.

#### Password security advice

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.

The Login Shell pulldown list allows you to select a login shell for the user. If you are not sure which shell to select, accept the default value of /bin/bash.
By default, the User Manager application creates the home directory for a new user in /home/username/. You can choose not to create the home directory by clearing the Create home directory check box, or change this directory by editing the content of the Home Directory text box. Note that when the home directory is created, default configuration files are copied into it from the /etc/skel/ directory.

Fedora uses a user private group (UPG) scheme. Whenever you create a new user, a unique group with the same name as the user is created by default. If you do not want to create this group, clear the Create a private group for the user check box.

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 1000 is assigned to the new user. Because Fedora reserves user IDs below 1000 for system users, it is not advisable to manually assign user IDs 1–999.

Clicking the OK button creates the new user. To configure more advanced user properties, such as password expiration, modify the user's properties after adding the user.

### 4.3.3. Adding a New Group

To add a new user group, select Add Group from the toolbar. A window similar to Figure 4.7, “New Group” appears. Type the name of the new group. 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 1000 for system groups.

![Figure 4.7. New Group](image)

Click OK to create the group. The new group appears in the group list.

### 4.3.4. 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 4.8, “User Properties” appears.
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.

### 4.3.5. 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 4.9, “Group Properties” appears.
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.

### 4.4. Using Command Line Tools

The easiest way to manage users and groups on Fedora is to use the **User Manager** application as described in Section 4.3, "Using the User Manager Tool". However, if you prefer command line tools or do not have the X Window System installed, you can use command line utilities that are listed in Table 4.1, “Command line utilities for managing users and groups”.

#### Table 4.1. Command line utilities for managing users and groups

<table>
<thead>
<tr>
<th>Utilities</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>useradd</code>, <code>usermod</code>, <code>userdel</code></td>
<td>Standard utilities for adding, modifying, and deleting user accounts.</td>
</tr>
<tr>
<td><code>groupadd</code>, <code>groupmod</code>, <code>groupdel</code></td>
<td>Standard utilities for adding, modifying, and deleting groups.</td>
</tr>
<tr>
<td><code>gpasswd</code></td>
<td>Standard utility for administering the <code>/etc/group</code> configuration file.</td>
</tr>
<tr>
<td><code>pwck</code>, <code>grpck</code></td>
<td>Utilities that can be used for verification of the password, group, and associated shadow files.</td>
</tr>
<tr>
<td><code>pwconv</code>, <code>pwunconv</code></td>
<td>Utilities that can be used for the conversion of passwords to shadow passwords, or back from shadow passwords to standard passwords.</td>
</tr>
</tbody>
</table>

#### 4.4.1. Adding a New User

To add a new user to the system, typing the following at a shell prompt as root:

```bash
useradd [options] username
```

...where **options** are command line options as described in Table 4.2, "useradd command line options".
By default, the **useradd** command creates a locked user account. To unlock the account, run the following command as root to assign a password:

```
passwd username
```

Optionally, you can set password aging policy. Refer to Section 4.4.3, “Enabling Password Aging” for information on how to enable password aging.

**Table 4.2. useradd command line options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c 'comment'</td>
<td><code>comment</code> can be replaced with any string. This option is generally used to specify the full name of a user.</td>
</tr>
<tr>
<td>-d <em>home_directory</em></td>
<td>Home directory to be used instead of default <em>/home/username/</em>.</td>
</tr>
<tr>
<td>-e <em>date</em></td>
<td>Date for the account to be disabled in the format YYYY-MM-DD.</td>
</tr>
<tr>
<td>-f <em>days</em></td>
<td>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.</td>
</tr>
<tr>
<td>-g <em>group_name</em></td>
<td>Group name or group number for the user's default group. The group must exist prior to being specified here.</td>
</tr>
<tr>
<td>-G <em>group_list</em></td>
<td>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.</td>
</tr>
<tr>
<td>-m</td>
<td>Create the home directory if it does not exist.</td>
</tr>
<tr>
<td>-M</td>
<td>Do not create the home directory.</td>
</tr>
<tr>
<td>-N</td>
<td>Do not create a user private group for the user.</td>
</tr>
<tr>
<td>-p <em>password</em></td>
<td>The password encrypted with crypt.</td>
</tr>
<tr>
<td>-r</td>
<td>Create a system account with a UID less than 1000 and without a home directory.</td>
</tr>
<tr>
<td>-s</td>
<td>User's login shell, which defaults to /bin/bash.</td>
</tr>
<tr>
<td>-u <em>uid</em></td>
<td>User ID for the user, which must be unique and greater than 999.</td>
</tr>
</tbody>
</table>

**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*:

```
juan:x:501:501::/home/juan:/bin/bash
```

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 999 is created. Under Fedora, UIDs below 1000 are reserved for system use and should not be assigned to users.
Adding a New User

- A GID greater than 999 is created. Under Fedora, GIDs below 1000 are reserved for system use and should not be assigned to users.
- 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:

   juan:!!:14798:0:99999:7:::

   The line has the following characteristics:
   - It begins with the username juan.
   - Two exclamation marks (!!) appear in the password field of the /etc/shadow file, which locks the account.
   - The password is set to never expire.

3. A new line for a group named juan is created in /etc/group:

   juan:x:501:

   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 4.1.1, “User Private Groups”.

   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:

   juan:!!:

   The line has the following characteristics:
   - It begins with the group name juan.
• An exclamation mark (!) 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:

```
~]# ls -l /home
total 4
 drwx------. 4 juan juan 4096 Mar  3 18:23 juan
```

This directory is owned by user juan and group juan. 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. The contents of `/etc/skel/` may vary depending on installed applications.

```
~]# ls -la /home/juan
total 28
 drwx------. 4 juan juan 4096 Mar  3 18:23 .
 drwxr-xr-x. 5 root root 4096 Mar  3 18:23 ..
 -rw-r--r--. 1 juan juan  18 Jul 09 08:43 .bash_logout
 -rw-r--r--. 1 juan juan  176 Jul 09 08:43 .bash_profile
 -rw-r--r--. 1 juan juan  124 Jul 09 08:43 .bashrc
 drwxr-xr-x. 4 juan juan 4096 Jul 09 08:43 .mozilla
 -rw-r--r--. 1 juan juan  658 Jul 09 08:43 .zshrc
```

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.

### 4.4.2. Adding a New Group

To add a new group to the system, type the following at a shell prompt as root:

```
groupadd [options] group_name
```

...where `options` are command line options as described in Table 4.3, “`groupadd` command line options”.

#### Table 4.3. `groupadd` command line options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f, --force</code></td>
<td>When used with <code>-g gid</code> and <code>gid</code> already exists, <code>groupadd</code> will choose another unique <code>gid</code> for the group.</td>
</tr>
<tr>
<td><code>-g gid</code></td>
<td>Group ID for the group, which must be unique and greater than 999.</td>
</tr>
<tr>
<td><code>-K, --key key=value</code></td>
<td>Override <code>/etc/login.defs</code> defaults.</td>
</tr>
<tr>
<td><code>-o, --non-unique</code></td>
<td>Allow to create groups with duplicate.</td>
</tr>
<tr>
<td><code>-p, --password password</code></td>
<td>Use this encrypted password for the new group.</td>
</tr>
<tr>
<td><code>-r</code></td>
<td>Create a system group with a GID less than 1000.</td>
</tr>
</tbody>
</table>
4.4.3. Enabling Password Aging

For security reasons, it is advisable to require users to change their passwords periodically. This can either be done when adding or editing a user on the Password Info tab of the User Manager application, or by using the `chage` command.

**Shadow passwords must be enabled to use `chage`**

Shadow passwords must be enabled to use the `chage` command. For more information, see Section 4.1.2, “Shadow Passwords”.

To configure password expiration for a user from a shell prompt, run the following command as `root`:

```
chage [options] username
```

...where `options` are command line options as described in Table 4.4, “`chage` command line options”. When the `chage` command is followed directly by a username (that is, when no command line options are specified), it displays the current password aging values and allows you to change them interactively.

**Table 4.4. `chage` command line options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-d days</code></td>
<td>Specifies the number of days since January 1, 1970 the password was changed.</td>
</tr>
<tr>
<td><code>-E date</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>-I days</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>-l</code></td>
<td>Lists current account aging settings.</td>
</tr>
<tr>
<td><code>-m days</code></td>
<td>Specify the minimum number of days after which the user must change passwords. If the value is 0, the password does not expire.</td>
</tr>
<tr>
<td><code>-M days</code></td>
<td>Specify 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 <code>-d</code> option is less than the current day, the user must change passwords before using the account.</td>
</tr>
<tr>
<td><code>-W days</code></td>
<td>Specifies the number of days before the password expiration date to warn the user.</td>
</tr>
</tbody>
</table>

You can configure a password to expire the first time a user logs in. This forces users to change passwords immediately.

1. Set up an initial password. There are two common approaches to this step: you can either assign a default password, or you can use a null password.

   To assign a default password, type the following at a shell prompt as `root`:
Chapter 4. Managing Users and Groups

4.4.4. Enabling Automatic Logouts

Especially when the user is logged in as root, an unattended login session may pose a significant security risk. To reduce this risk, you can configure the system to automatically log out idle users after a fixed period of time:

1. Make sure the screen package is installed. You can do so by running the following command as root:

   ```bash
   yum install screen
   ```

   For more information on how to install packages in Fedora, refer to Section 5.2.4, “Installing Packages”.

2. As root, add the following line at the beginning of the `/etc/profile` file to make sure the processing of this file cannot be interrupted:

   ```bash
   trap "" 1 2 3 15
   ```

3. Add the following lines at the end of the `/etc/profile` file to start a `screen` session each time a user logs in to a virtual console or remotely:

   ```bash
   SCREENEXEC="screen"
   if [ -w $tty ]; then
       trap "exec $SCREENEXEC" 1 2 3 15
       echo -n 'Starting session in 10 seconds'
   ```

To assign a null password instead, use the following command:

```
passwd -d username
```

Avoid using null passwords whenever possible

Using a null password, while convenient, is a highly insecure practice, as any third party can log in first and access the system using the insecure username. Always make sure that the user is ready to log in before unlocking an account with a null password.

2. Force immediate password expiration by running the following command as root:

```
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.

Upon the initial log in, the user is now prompted for a new password.
Note that each time a new session starts, a message will be displayed and the user will have to wait ten seconds. To adjust the time to wait before starting a session, change the value after the `sleep` command.

4. Add the following lines to the `/etc/screenrc` configuration file to close the `screen` session after a given period of inactivity:

```
idle 120 quit
autodetach off
```

This will set the time limit to 120 seconds. To adjust this limit, change the value after the `idle` directive.

Alternatively, you can configure the system to only lock the session by using the following lines instead:

```
idle 120 lockscreen
autodetach off
```

This way, a password will be required to unlock the session.

The changes take effect the next time a user logs in to the system.

### 4.4.5. Creating Group Directories

System administrators usually like to create a group for each major project and assign people to the group when they need to access that project's files. With this traditional scheme, file managing is 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 becomes difficult to associate the right files with the right group. However, with the UPG scheme, 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.

For example, a group of people need to work on files in the `/opt/myproject/` directory. Some people are trusted to modify the contents of this directory, but not everyone.

1. As root, create the `/opt/myproject/` directory by typing the following at a shell prompt:

```
mkdir /opt/myproject
```

2. Add the `myproject` group to the system:

```
groupadd myproject
```

3. Associate the contents of the `/opt/myproject/` directory with the `myproject` group:

```
chown root:myproject /opt/myproject
```
4. Allow users to create files within the directory, and set the setgid bit:

```bash
chmod 2775 /opt/myproject
```

At this point, all members of the `myproject` group can create and edit files in the `/opt/myproject/` directory without the administrator having to change file permissions every time users write new files. To verify that the permissions have been set correctly, run the following command:

```
[-]# ls -l /opt
total 4
drwxrwsr-x. 3 root myproject 4096 Mar  3 18:31 myproject
```

### 4.5. Additional Resources

Refer to the following resources for more information about managing users and groups.

#### 4.5.1. Installed Documentation

For information about various utilities for managing users and groups, refer to the following manual pages:

- **chage**(1) — A command to modify password aging policies and account expiration.
- **gpasswd**(1) — A command to administer the `/etc/group` file.
- **groupadd**(8) — A command to add groups.
- **grpck**(8) — A command to verify the `/etc/group` file.
- **groupdel**(8) — A command to remove groups.
- **groupmod**(8) — A command to modify group membership.
- **pwck**(8) — A command to verify the `/etc/passwd` and `/etc/shadow` files.
- **pwconv**(8) — A tool to convert standard passwords to shadow passwords.
- **pwunconv**(8) — A tool to convert shadow passwords to standard passwords.
- **useradd**(8) — A command to add users.
- **userdel**(8) — A command to remove users.
- **usermod**(8) — A command to modify users.

For information about related configuration files, see:

- **group**(5) — The file containing group information for the system.
- **passwd**(5) — The file containing user information for the system.
- **shadow**(5) — The file containing passwords and account expiration information for the system.
Part II. 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 both Yum and the PackageKit suite of graphical package management tools.
Yum

Yum is the The Fedora Project 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 plug-ins 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 5.3, “Configuring Yum and Yum Repositories” for details on enabling signature-checking with Yum, or Section A.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 6, PackageKit for details on using PackageKit.

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.

5.1. Checking For and Updating Packages

5.1.1. Checking For Updates

To see which installed packages on your system have updates available, use the following command:

```
yum check-update
```

For example:
Chapter 5. Yum

---

```bash
# yum check-update
```

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

- **PackageKit** — the name of the package
- **x86_64** — the CPU architecture the package was built for
- **0.6.14** — 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`.

### 5.1.2. 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.

#### Updating a Single Package

To update a single package, run the following command as `root`:

```bash
yum update package_name
```

For example, to update the `udev` package, type:

```bash
# yum update udev
```

```
Loaded plugins: langpacks, presto, refresh-packagekit
Updating Red Hat repositories.
INFO:rhsm-app.repolib:repos updated: 0
Setting up Update Process
Resolving Dependencies
---> Running transaction check
--- Package gdb.x86_64 0:7.2.90.20110411-34.fc15 will be updated
```
This output contains several items of interest:

1. **Loaded plugins**: — `yum` always informs you which Yum plug-ins are installed and enabled. Here, `yum` is using the `langpacks`, `presto`, and `refresh-packagekit` plug-ins. Refer to Section 5.4, “Yum Plug-ins” for general information on Yum plug-ins, or to Section 5.4.3, “Plug-in Descriptions” for descriptions of specific plug-ins.

2. **gdb.x86_64** — you can download and install new `gdb` package.

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 transaction history by using the `yum history` command as described in Section 5.2.6, “Working with Transaction History”.

### Updating and installing kernels with Yum

`yum` always installs 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 A.2.2, “Installing and Upgrading” for more information on installing/updating kernels with RPM.

### Updating All Packages and Their Dependencies

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

```
yum update
```
Chapter 5. Yum

Updating Security-Related Packages

Discovering which packages have security updates available and then updating those packages quickly and easily is important. Yum provides the plug-in for this purpose. The security plug-in extends the yum command with a set of highly-useful security-centric commands, subcommands and options. Refer to Section 5.4.3, “Plug-in Descriptions” for specific information.

5.1.3. 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 A.2.2, “Installing and Upgrading” for details on how to manage changes to configuration files across package upgrades.

5.2. Packages and Package Groups

5.2.1. Searching Packages

You can search all RPM package names, descriptions and summaries by using the following command:

```
yum search term...
```

This command displays the list of matches for each term. For example, to list all packages that match “meld” or “kompare”, type:

```
~]# yum search meld kompare
```

Loaded plugins: langpacks, presto, refresh-packagekit
================================================================== N/S Matched: meld ===============
meld.noarch : Visual diff and merge tool
python-meld3.x86_64 : HTML/XML templating system for Python

================================================================== N/S Matched: kompare ===============
komparator.x86_64 : Kompare and merge two folders

Name and summary matches only, use "search all" for everything.

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

5.2.2. Listing Packages

```
yum list
```

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

All of Yum’s list commands allow you to filter the results by appending one or more glob expressions as arguments. Glob expressions are normal strings of characters which contain one or more of the wildcard characters * (which expands to match any character multiple times) and ? (which expands to match any one character).
Filtering results with glob expressions

Be careful to escape the glob expressions when passing them as arguments to a `yum` command, otherwise the Bash shell will interpret these expressions as *pathname expansions*, and potentially pass all files in the current directory that match the globs to `yum`. To make sure the glob expressions are passed to `yum` as intended, either:

- escape the wildcard characters by preceding them with a backslash character
- double-quote or single-quote the entire glob expression.

Refer to `Example 5.1, “Listing all ABRT addons and plug-ins using glob expressions”` and `Example 5.4, “Listing available packages using a single glob expression with escaped wildcard characters”` for an example usage of both these methods.

**yum list glob_expression**
Lists information on installed and available packages matching all glob expressions.

**Example 5.1. Listing all ABRT addons and plug-ins using glob expressions**
Packages with various ABRT addons and plug-ins either begin with "abrt-addon-", or "abrt-plugin-". To list these packages, type the following at a shell prompt:

```bash
~ ]# yum list abrt-addon* abrt-plugin*
Loaded plugins: langpacks, presto, refresh-packagekit
Installed Packages
abrt-addon-ccpp.x86_64               2.0.2-5.fc15     @fedora
abrt-addon-kerneloops.x86_64         2.0.2-5.fc15     @fedora
abrt-addon-python.x86_64             2.0.2-5.fc15     @fedora
abrt-plugin-bugzilla.x86_64          2.0.2-5.fc15     @fedora
abrt-plugin-logger.x86_64            2.0.2-5.fc15     @fedora
Available Packages
abrt-plugin-mailx.x86_64             2.0.2-5.fc15     updates
abrt-plugin-reportuploader.x86_64    2.0.2-5.fc15     updates
abrt-plugin-rhtsupport.x86_64        2.0.2-5.fc15     updates
```

**yum list all**
Lists all installed *and* available packages.

**Example 5.2. Listing all installed and available packages**

```bash
~ ]# yum list all
Loaded plugins: langpacks, presto, refresh-packagekit
Installed Packages
ConsoleKit.x86_64                       0.4.4-1.fc15                  @fedora
ConsoleKit-libs.x86_64                  0.4.4-1.fc15                  @fedora
ConsoleKit-x11.x86_64                   0.4.4-1.fc15                  @fedora
GConf2.x86_64                           2.32.3-1.fc15                 @fedora
GConf2-gtk.x86_64                       2.32.3-1.fc15                 @fedora
ModemManager.x86_64                     0.4-7.git20110201.fc15       @fedora
NetworkManager.x86_64                   1:0.8.998-4.git20110427.fc15  @fedora
NetworkManager-glib.x86_64              1:0.8.998-4.git20110427.fc15  @fedora
NetworkManager-gnome.x86_64             1:0.8.998-4.git20110427.fc15  @fedora
```
Chapter 5. Yum

Draft

NetworkManager-openconnect.x86_64 0.8.1-9.git20110419.fc15 @fedora

[youtput truncated]

**yum list installed**
Lists all packages installed on your system. The rightmost column in the output lists the repository from which the package was retrieved.

**Example 5.3. Listing installed packages using a double-quoted glob expression**
To list all installed packages that begin with “krb” followed by exactly one character and a hyphen, type:

```
]# yum list installed "krb?--*"
```

**yum list available**
Lists all available packages in all enabled repositories.

**Example 5.4. Listing available packages using a single glob expression with escaped wildcard characters**
To list all available packages with names that contain “gstreamer” and then “plugin”, run the following command:

```
]# yum list available gstreamer\*plugin\*
```

**yum grouplist**
Lists all package groups.

**Example 5.5. Listing all package groups**

```
]# yum grouplist
```
**yum repolist**

Lists the repository ID, name, and number of packages it provides for each *enabled* repository.

Example 5.6. Listing enabled repositories

```
~$]# yum repolist
loaded plugins: langpacks, presto, refresh-packagekit
repo id     repo name                    status
fedora      Fedora 15 - i386             19,365
updates     Fedora 15 - i386 - Updates   3,848
repolist:   23,213
```

---

### 5.2.3. Displaying Package Information

To display information about one or more packages (glob expressions are valid here as well), use the following command:

```
yum info package_name...
```

For example, to display information about the `abrt` package, type:

```
~$]# yum info abrt
loaded plugins: langpacks, presto, refresh-packagekit
Installed Packages
Name        : abrt
Arch        : x86_64
Version     : 2.0.1
Release     : 2.fc15
Size        : 806 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 a bug report with all informations needed by maintainer
              to fix it. It uses plugin system to extend its functionality.
```

The `yum info package_name` command 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).

You can also query the Yum database for alternative and useful information about a package by using the following command:

```
yumdb info package_name
```

This command provides additional information about a package, including the checksum of the package (and algorithm used to produce it, such as SHA-256), the command given on the command line that was invoked to install the package (if any), and the reason that the package is installed on the system (where *user* indicates it was installed by the user, and *dep* means it was brought in as a dependency). For example, to display additional information about the `yum` package, type:
5.2.4. Installing Packages

Yum allows you to install both a single package and multiple packages, as well as a package group of your choice.

**Installing Individual Packages**

To install a single package and all of its non-installed dependencies, enter a command in the following form:

```
yum install package_name
```

You can also install multiple packages simultaneously by appending their names as arguments:

```
yum install package_name package_name...
```

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 is available in an enabled repository) by appending `.arch` to the package name. For example, to install the `sqlite2` package for i586, type:

```
~# 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.
Finding which package owns a file

If you know you want to install the package that contains the named binary, but you do not know in which bin or sbin directory is the file installed, use the yum provides command with a glob expression:

```bash
-# yum provides "*bin/named"
Loaded plugins: langpacks, presto, refresh-packagekit
32:bind-9.8.0-3.P1.fc15.i686 : The Berkeley Internet Name Domain (BIND) DNS
            : (Domain Name System) server
Repo        : fedora
Matched from:
Filename    : /usr/sbin/named
```

**yum provides "*/file_name"** is a common and useful trick to find the packages that contain file_name.

Installing a Package Group

A package group is similar to a package: it is not useful by itself, but installing one pulls 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 in the following example:

```bash
-# yum -v grouplist kde\*
Not loading "blacklist" plugin, as it is disabled
Loading "langpacks" plugin
Loading "presto" plugin
Loading "refresh-packagekit" plugin
Not loading "whiteout" plugin, as it is disabled
Adding en_US to language list
Config time: 0.900
Yum Version: 3.2.29
Setting up Group Process
rpmdb time: 0.002
group time: 0.995
Available Groups:
            KDE Software Compilation (kde-desktop)
            KDE Software Development (kde-software-development)
Done
```

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

```
yum groupinstall group_name
```

You can also install by groupid:

```
yum groupinstall groupid
```

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):
For example, the following are alternative but equivalent ways of installing the KDE Desktop group:

```
~]# yum groupinstall "KDE Desktop"
~]# yum groupinstall kde-desktop
~]# yum install @kde-desktop
```

### 5.2.5. Removing Packages

Similarly to package installation, Yum allows you to uninstall (remove in RPM and Yum terminology) both individual packages and a package group.

#### Removing Individual Packages

To uninstall a particular package, as well as any packages that depend on it, run the following command as root:

```
yum remove package_name...
```

As when you install multiple packages, you can remove several at once by adding more package names to the command. For example, to remove totem, rhythmbox, and sound-juicer, type the following at a shell prompt:

```
~]# yum remove totem rhythmbox sound-juicer
```

Similar to `install`, `remove` can take these arguments:

- package names
- glob expressions
- file lists
- package provides

---

**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 A.2.4, “Uninstalling” in the RPM chapter.

#### Removing a Package Group

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

```
yum groupremove group
```
The following are alternative but equivalent ways of removing the **KDE Desktop** group:

```
~# yum groupremove "KDE Desktop"
~# yum groupremove kde-desktop
~# yum remove @kde-desktop
```

**Intelligent package group removal**

When you tell `yum` to remove a package group, it will remove every package in that group, even if those packages are members of other package groups or dependencies of other installed packages. However, you can instruct `yum` to remove only those packages which are not required by any other packages or groups by adding the `groupremove_leaf_only=1` directive to the `[main]` section of the `/etc/yum.conf` configuration file. For more information on this directive, refer to Section 5.3.1, “Setting `[main]` Options”.

### 5.2.6. Working with Transaction History

The `yum history` command allows users to review information about a timeline of Yum transactions, the dates and times on when they occurred, the number of packages affected, whether transactions succeeded or were aborted, and if the RPM database was changed between transactions. Additionally, this command can be used to undo or redo certain transactions.

#### Listing Transactions

To display a list of twenty most recent transactions, as root, either run `yum history` with no additional arguments, or type the following at a shell prompt:

```
yum history list
```

To display all transactions, add the `all` keyword:

```
yum history list all
```

To display only transactions in a given range, use the command in the following form:

```
yum history list start_id..end_id
```

You can also list only transactions regarding a particular package or packages. To do so, use the command with a package name or a glob expression:

```
yum history list glob_expression...
```

For example, the list of first five transactions may look as follows:

```
~# yum history list 1..5
Loaded plugins: langpacks, presto, refresh-packagekit
```
All forms of the `yum history list` command produce tabular output with each row consisting of the following columns:

- **ID** — an integer value that identifies a particular transaction.

- **Login user** — the name of the user whose login session was used to initiate a transaction. This information is typically presented in the Full Name `<username>` form. For transactions that were not issued by a user (such as an automatic system update), `System <unset>` is used instead.

- **Date and time** — the date and time when a transaction was issued.

- **Action(s)** — a list of actions that were performed during a transaction as described in Table 5.1, “Possible values of the Action(s) field”.

- **Altered** — the number of packages that were affected by a transaction, possibly followed by additional information as described in Table 5.2, “Possible values of the Altered field”.

### Table 5.1. Possible values of the Action(s) field

<table>
<thead>
<tr>
<th>Action</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Downgrade</td>
<td>D</td>
<td>At least one package has been downgraded to an older version.</td>
</tr>
<tr>
<td>Erase</td>
<td>E</td>
<td>At least one package has been removed.</td>
</tr>
<tr>
<td>Install</td>
<td>I</td>
<td>At least one new package has been installed.</td>
</tr>
<tr>
<td>Obsoleting</td>
<td>O</td>
<td>At least one package has been marked as obsolete.</td>
</tr>
<tr>
<td>Reinstall</td>
<td>R</td>
<td>At least one package has been reinstalled.</td>
</tr>
<tr>
<td>Update</td>
<td>U</td>
<td>At least one package has been updated to a newer version.</td>
</tr>
</tbody>
</table>

### Table 5.2. Possible values of the Altered field

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td>Before the transaction finished, the <code>rpmdb</code> database was changed outside Yum.</td>
</tr>
<tr>
<td>&gt;</td>
<td>After the transaction finished, the <code>rpmdb</code> database was changed outside Yum.</td>
</tr>
<tr>
<td>*</td>
<td>The transaction failed to finish.</td>
</tr>
<tr>
<td>#</td>
<td>The transaction finished successfully, but <code>yum</code> returned a non-zero exit code.</td>
</tr>
<tr>
<td>E</td>
<td>The transaction finished successfully, but an error or a warning was displayed.</td>
</tr>
<tr>
<td>P</td>
<td>The transaction finished successfully, but problems already existed in the <code>rpmdb</code> database.</td>
</tr>
<tr>
<td>s</td>
<td>The transaction finished successfully, but the <code>--skip-broken</code> command line option was used and certain packages were skipped.</td>
</tr>
</tbody>
</table>

Yum also allows you to display a summary of all past transactions. To do so, run the command in the following form as `root`:

```
yum history summary
```
To display only transactions in a given range, type:

```
yum history summary start_id..end_id
```

Similarly to the `yum history list` command, you can also display a summary of transactions regarding a certain package or packages by supplying a package name or a glob expression:

```
yum history summary glob_expression...
```

For instance, a summary of the transaction history displayed above would look like the following:

```
~]# yum history summary 1..5
Loaded plugins: langpacks, presto, refresh-packagekit
Login user | Time       | Action(s) | Altered
-----------|------------|-----------|---------
Jaromir ... <jhradilek> | Last day   | Install   |        1
Jaromir ... <jhradilek> | Last week  | Install   |        1
Jaromir ... <jhradilek> | Last 2 weeks | I, U     |       73
System <unset> | Last 2 weeks | I, U     | 1107
history summary
```

All forms of the `yum history summary` command produce simplified tabular output similar to the output of `yum history list`.

As shown above, both `yum history list` and `yum history summary` are oriented towards transactions, and although they allow you to display only transactions related to a given package or packages, they lack important details, such as package versions. To list transactions from the perspective of a package, run the following command as `root`:

```
yum history package-list glob_expression...
```

For example, to trace the history of `subscription-manager` and related packages, type the following at a shell prompt:

```
~]# yum history package-list subscription-manager\`
```

```
ID | Action(s) | Package
---|-----------|---------
 3 | Updated   | subscription-manager-0.95.11-1.el6.x86_64
 3 | Update    | 0.95.11-1.el6.x86_64
 3 | Updated   | subscription-manager-firstboot-0.95.11-1.el6.x86_64
 3 | Update    | 0.95.11-1.el6.x86_64
 3 | Updated   | subscription-manager-gnome-0.95.11-1.el6.x86_64
 3 | Update    | 0.95.11-1.el6.x86_64
 1 | Install   | subscription-manager-0.95.11-1.el6.x86_64
 1 | Install   | subscription-manager-firstboot-0.95.11-1.el6.x86_64
 1 | Install   | subscription-manager-gnome-0.95.11-1.el6.x86_64
history package-list
```

In this example, three packages were installed during the initial system installation: `subscription-manager`, `subscription-manager-firstboot`, and `subscription-manager-gnome`. In the third transaction, all these packages were updated from version 0.95.11 to version 0.95.17.

**Examining Transactions**

To display the summary of a single transaction, as `root`, use the `yum history summary` command in the following form:
To examine a particular transaction or transactions in more detail, run the following command as root:

`yum history info id...`

The `id` argument is optional and when you omit it, `yum` automatically uses the last transaction. Note that when specifying more than one transaction, you can also use a range:

`yum history info start_id..end_id`

The following is sample output for two transactions, each installing one new package:

```
~# yum history info 4..5
Loaded plugins: langpacks, presto, refresh-packagekit
Transaction ID: 4..5
Begin time: Thu Jul 21 15:10:46 2011
Begin rpmdb: 1107:0c67c32210c1999f92ed8da7572b4c6f64edae3a
End time: 15:33:15 2011 (22 minutes)
End rpmdb: 1109:1171025a56b65f68d8b30d63590f590f1c1f3242
User: Jaromir Hradilek <jhradilek>
Return-Code: Success
Command Line: install screen
Command Line: install yum-plugin-fs-snapshot
Transaction performed with:
  Installed rpm-4.8.0-16.el6.x86_64
  Installed yum-3.2.29-17.el6.noarch
  Installed yum-metadata-parser-1.1.2-16.el6.x86_64
Packages Altered:
  Install screen-4.0.3-16.el6.x86_64
  Install yum-plugin-fs-snapshot-1.1.30-6.el6.noarch
history info
```

You can also view additional information, such as what configuration options were used at the time of the transaction, or from what repository and why were certain packages installed. To determine what additional information is available for a certain transaction, type the following at a shell prompt as root:

`yum history addon-info id`

Similarly to `yum history info`, when no `id` is provided, `yum` automatically uses the latest transaction. Another way to refer to the latest transaction is to use the `last` keyword:

`yum history addon-info last`

For instance, for the first transaction in the previous example, the `yum history addon-info` command would provide the following output:

```
~# yum history addon-info 4
Loaded plugins: langpacks, presto, refresh-packagekit
Transaction ID: 4
Available additional history information:
  config-main
  config-repos
```
In this example, three types of information are available:

- **config-main** — global Yum options that were in use during the transaction. Refer to Section 5.3.1, “Setting [main] Options” for information on how to change global options.

- **config-repos** — options for individual Yum repositories. Refer to Section 5.3.2, “Setting [repository] Options” for information on how to change options for individual repositories.

- **saved_tx** — the data that can be used by the `yum load-transaction` command in order to repeat the transaction on another machine (see below).

To display selected type of additional information, run the following command as root:

```
yum history addon-info id information
```

### Reverting and Repeating Transactions

Apart from reviewing the transaction history, the `yum history` command provides means to revert or repeat a selected transaction. To revert a transaction, type the following at a shell prompt as root:

```
yum history undo id
```

To repeat a particular transaction, as root, run the following command:

```
yum history redo id
```

Both commands also accept the last keyword to undo or repeat the latest transaction.

Note that both `yum history undo` and `yum history redo` commands merely revert or repeat the steps that were performed during a transaction: if the transaction installed a new package, the `yum history undo` command will uninstall it, and vice versa. If possible, this command will also attempt to downgrade all updated packages to their previous version, but these older packages may no longer be available. If you need to be able to restore the system to the state before an update, consider using the `fs-snapshot` plug-in described in Section 5.4.3, “Plug-in Descriptions”.

When managing several identical systems, Yum also allows you to perform a transaction on one of them, store the transaction details in a file, and after a period of testing, repeat the same transaction on the remaining systems as well. To store the transaction details to a file, type the following at a shell prompt as root:

```
yum -q history addon-info id saved_tx > file_name
```

Once you copy this file to the target system, you can repeat the transaction by using the following command as root:

```
yum load-transaction file_name
```

Note, however that the `rpmdb` version stored in the file must by identical to the version on the target system. You can verify the `rpmdb` version by using the `yum version nogroups` command.
Starting New Transaction History

Yum stores the transaction history in a single SQLite database file. To start new transaction history, run the following command as root:

```
yum history new
```

This will create a new, empty database file in the `/var/lib/yum/history/` directory. The old transaction history will be kept, but will not be accessible as long as a newer database file is present in the directory.

5.3. Configuring Yum and Yum Repositories

The configuration file for `yum` and related utilities is located at `/etc/yum.conf`. This file contains one mandatory `[main]` section, which allows you to set Yum options that have global effect, and may also contain one or more `[repository]` sections, which allow you to set repository-specific options. However, best practice is to define individual repositories in new or existing `.repo` files in the `/etc/yum.repos.d/` directory. The values you define in the `[main]` section of the `/etc/yum.conf` file may override values set in individual `[repository]` sections.

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 the `/etc/yum.repos.d/` directory so that dynamic version and architecture values are handled correctly;
- add, enable, and disable Yum repositories on the command line; and,
- set up your own custom Yum repository.

5.3.1. Setting `[main]` Options

The `/etc/yum.conf` configuration file contains exactly one `[main]` section, and while some of the key-value pairs in this section affect how `yum` operates, others affect how Yum treats repositories. You can add many additional options under the `[main]` section heading in `/etc/yum.conf`.

A sample `/etc/yum.conf` configuration file can look like this:

```
[main]
cachedir=/var/cache/yum/$basearch/$releasever
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
```
The following are the most commonly-used options in the [main] section:

- **assumeyes=value**
  - where value is one of:
    - 0 — yum should prompt for confirmation of critical actions it performs. This is the default.
    - 1 — Do not prompt for confirmation of critical yum actions. If assumeyes=1 is set, yum behaves in the same way that the command line option -y does.

- **cachedir=directory**
  - where directory is an absolute path to the directory where Yum should store its cache and database files. By default, Yum's cache directory is /var/cache/yum/$basearch/$releasever.
  - Refer to Section 5.3.3, “Using Yum Variables” for descriptions of the $basearch and $releasever Yum variables.

- **debuglevel=value**
  - where value is an integer between 1 and 10. Setting a higher debuglevel value causes yum to display more detailed debugging output. debuglevel=0 disables debugging output, while debuglevel=2 is the default.

- **exactarch=value**
  - where value is one of:
    - 0 — Do not take into account the exact architecture when updating packages.
    - 1 — Consider the exact architecture when updating packages. With this setting, yum will not install an i686 package to update an i386 package already installed on the system. This is the default.

- **exclude=package_name [more_package_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.

- **gpgcheck=value**
  - where value is one of:
    - 0 — Disable GPG signature-checking on packages in all repositories, including local package installation.
    - 1 — Enable GPG signature-checking on all packages in all repositories, including local package installation. gpgcheck=1 is the default, and thus all packages' signatures are checked.

  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 gpgcheck=value for individual repositories instead; that is, you can enable GPG-checking on one repository while disabling it on another. Setting gpgcheck=value for an individual repository in its corresponding .repo file overrides the default if it is present in /etc/yum.conf.

  For more information on GPG signature-checking, refer to Section A.3, “Checking a Package’s Signature”.

- **groupremove_leaf_only=value**
  - where value is one of:
Chapter 5. Yum

0 — yum should not check the dependencies of each package when removing a package group. With this setting, yum removes all packages in a package group, regardless of whether those packages are required by other packages or groups. groupremove_leaf_only=0 is the default.

1 — yum should check the dependencies of each package when removing a package group, and remove only those packages which are not not required by any other package or group.

For more information on removing packages, refer to Intelligent package group removal.

installonlypkgs=space separated list of packages
Here you can provide a space-separated list of packages which yum can install, but will never update. Refer to the yum.conf(5) manual page for the list of packages which are install-only by default.

If you add the installonlypkgs directive to /etc/yum.conf, you should ensure that you list all of the packages that should be install-only, including any of those listed under the installonlypkgs section of yum.conf(5). In particular, kernel packages should always be listed in installonlypkgs (as they are by default), and installonly_limit should always be set to a value greater than 2 so that a backup kernel is always available in case the default one fails to boot.

installonly_limit=value
...where value is an integer representing the maximum number of versions that can be installed simultaneously for any single package listed in the installonlypkgs directive.

The defaults for the installonlypkgs directive include several different kernel packages, so be aware that changing the value of installonly_limit will also affect the maximum number of installed versions of any single kernel package. The default value listed in /etc/yum.conf is installonly_limit=3, and it is not recommended to decrease this value, particularly below 2.

keepcache=value
...where value is one of:

0 — Do not retain the cache of headers and packages after a successful installation. This is the default.

1 — Retain the cache after a successful installation.

logfile=file_name
...where file_name is an absolute path to the file in which yum should write its logging output. By default, yum logs to /var/log/yum.log.

multilib_policy=value
...where value is one of:

best — install the best-choice architecture for this system. For example, setting multilib_policy=best on an AMD64 system causes yum to install 64-bit versions of all packages.

all — always install every possible architecture for every package. For example, with multilib_policy set to all on an AMD64 system, yum would install both the i586 and AMD64 versions of a package, if both were available.

obsoletes=value
...where value is one of:

0 — Disable yum's obsoletes processing logic when performing updates.
1 — Enable yum's obsoletes processing logic when performing updates. When one package declares in its spec file that it obsoletes another package, the latter package will be replaced by the former package when the former package is installed. Obsoletes are declared, for example, when a package is renamed. `obsoletes=1` the default.

```plaintext
plugins=value
...where value is one of:

0 — Disable all Yum plug-ins globally.
```

### Disabling all plug-ins is not advised

Disabling all plug-ins is not advised because certain plug-ins provide important Yum services. Disabling plug-ins globally is provided as a convenience option, and is generally only recommended when diagnosing a potential problem with Yum.

1 — Enable all Yum plug-ins globally. With `plugins=1`, you can still disable a specific Yum plug-in by setting `enabled=0` in that plug-in's configuration file.

For more information about various Yum plug-ins, refer to Section 5.4, “Yum Plug-ins”. For further information on controlling plug-ins, see Section 5.4.1, “Enabling, Configuring, and Disabling Yum Plug-ins”.

```plaintext
reposdir=directory
...where directory is an absolute path to the directory where .repo files are located. All .repo files contain repository information (similar to the [repository] sections 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. If reposdir is not set, yum uses the default directory /etc/yum.repos.d/.
```

```plaintext
retries=value
...where value is an integer 0 or greater. This value 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 10.
```

For a complete list of available [main] options, refer to the [main] OPTIONS section of the yum.conf(5) manual page.

### 5.3.2. Setting [repository] Options

The [repository] sections, where repository is a unique repository ID such as my_personal_repo (spaces are not permitted), allow you to define individual Yum repositories.

The following is a bare-minimum example of the form a [repository] section takes:

```plaintext
[repository]
name=repository_name
baseurl=repository_url
```

Every [repository] section must contain the following directives:
name=repository_name
...where repository_name is a human-readable string describing the repository.

baseurl=repository_url
...where repository_url is a URL to the directory where the repodata directory of a repository is located:

- If the repository is available over HTTP, use: http://path/to/repo
- 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 by prepending it to the URL 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 could be specified as http://user:password@www.example.com/repo/.

Usually this URL is an HTTP link, such as:

```
baseurl=http://path/to/repo/releases/$releasever/server/$basearch/os/
```

Note that Yum always expands the $releasever, $arch, and $basearch variables in URLs. For more information about Yum variables, refer to Section 5.3.3, “Using Yum Variables”.

Another useful [repository] directive is the following:

enabled=value
...where value is one of:

0 — Do not include this 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 or installs.

1 — Include this repository as a package source.

Turning repositories on and off can also be performed by passing either the --enablerepo=repo_name or --disablerepo=repo_name option to yum, or through the Add/Remove Software window of the PackageKit utility.

Many more [repository] options exist. For a complete list, refer to the [repository] OPTIONS section of the yum.conf(5) manual page.

5.3.3. Using Yum Variables

You can use and reference the following built-in variables in yum commands and in all Yum configuration files (that is, /etc/yum.conf and all .repo files in the /etc/yum.repos.d/ directory):

$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.

$YUMO-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.

To define a custom variable or to override the value of an existing one, create a file with the same name as the variable (without the "$" sign) in the `/etc/yum/vars/` directory, and add the desired value on its first line.

For example, repository descriptions often include the operating system name. To define a new variable called $osname, create a new file with “Fedora” on the first line and save it as `/etc/yum/vars/osname`:

```
~$]# echo "Fedora" > /etc/yum/vars/osname
```

Instead of “Fedora 20”, you can now use the following in the `.repo` files:

```
name=$osname $releasever
```

5.3.4. Viewing the Current Configuration
To display the current values of global Yum options (that is, the options specified in the [main] section of the `/etc/yum.conf` file), run the `yum-config-manager` with no command line options:

```
yum-config-manager
```

To list the content of a different configuration section or sections, use the command in the following form:

```
yum-config-manager section...
```

You can also use a glob expression to display the configuration of all matching sections:

```
yum-config-manager glob_expression...
```

For example, to list all configuration options and their corresponding values, type the following at a shell prompt:

```
~$ yum-config-manager main \
Loaded plugins: langpacks, presto, refresh-packagekit
================================= main ===================================
[main]
```
alwaysprompt = True
assumeyes = False
bandwith = 0
bugtracker_url = https://bugzilla.redhat.com/enter_bug.cgi?product=Red%20Hat%20Enterprise
%20Linux%206&component=yum
cache = 0
[output truncated]

5.3.5. Adding, Enabling, and Disabling a Yum Repository

Section 5.3.2, “Setting [repository] Options” described various options you can use to define a Yum repository. This section explains how to add, enable, and disable a repository by using the `yum-config-manager` command.

Adding a Yum Repository

To define a new repository, you can either add a `[repository]` section to the `/etc/yum.conf` file, or to a `.repo` file in the `/etc/yum.repos.d/` directory. All files with the `.repo` file extension in this directory are read by `yum`, and best practice is to define your repositories here instead of in `/etc/yum.conf`.

Be careful when using untrusted software sources

Obtaining and installing software packages from unverified or untrusted software sources constitutes a potential security risk, and could lead to security, stability, compatibility maintainability issues.

Yum repositories commonly provide their own `.repo` file. To add such a repository to your system and enable it, run the following command as root:

```
yum-config-manager --add-repo repository_url
```

...where `repository_url` is a link to the `.repo` file. For example, to add a repository located at http://www.example.com/example.repo, type the following at a shell prompt:

```
$ yum-config-manager --add-repo http://www.example.com/example.repo
Loaded plugins: langpacks, presto, refresh-packagekit
adding repo from: http://www.example.com/example.repo
grabbing file http://www.example.com/example.repo to /etc/yum.repos.d/example.repo
example.repo                                             |  413 B     00:00
repo saved to /etc/yum.repos.d/example.repo
```

Enabling a Yum Repository

To enable a particular repository or repositories, type the following at a shell prompt as root:

```
yum-config-manager --enable repository...
```

...where `repository` is the unique repository ID (use `yum repolist all` to list available repository IDs). Alternatively, you can use a glob expression to enable all matching repositories:
For example, to disable repositories defined in the [example], [example-debuginfo], and [example-source] sections, type:

```bash
-][# yum-config-manager --enable example*
```

When successful, the `yum-config-manager --enable` command displays the current repository configuration.

**Disabling a Yum Repository**

To disable a Yum repository, run the following command as root:

```bash
yum-config-manager --disable repository...
```

...where `repository` is the unique repository ID (use `yum repolist all` to list available repository IDs). Similarly to `yum-config-manager --enable`, you can use a glob expression to disable all matching repositories at the same time:

```bash
yum-config-manager --disable glob_expression...
```

When successful, the `yum-config-manager --disable` command displays the current configuration.

### 5.3.6. Creating a Yum Repository

To set up a Yum repository, follow these steps:

1. Install the `createrepo` package:
   ```bash
   -][# 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:
   ```bash
   -][# createrepo --database /mnt/local_repo
   ```

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

### 5.4. Yum Plug-ins
Yum provides plug-ins that extend and enhance its operations. Certain plug-ins are installed by default. Yum always informs you which plug-ins, if any, are loaded and active whenever you call any `yum` command. For example:

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

Note that the plug-in names which follow `Loaded plugins` are the names you can provide to the `--disableplugins=plugin_name` option.

### 5.4.1. Enabling, Configuring, and Disabling Yum Plug-ins

To enable Yum plug-ins, 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 plug-ins by changing this line to `plugins=0`.

**Disabling all plug-ins is not advised**

Disabling all plug-ins is not advised because certain plug-ins provide important Yum services. Disabling plug-ins globally is provided as a convenience option, and is generally only recommended when diagnosing a potential problem with Yum.

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

```
[main]
enabled=1
```

Plug-in 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 plug-in is enabled when you run `yum` commands.

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

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

If you want to disable one or more Yum plug-ins for a single `yum` command, add the `--disableplugin=plugin_name` option to the command. For example, to disable the `presto` plug-in while updating a system, type:

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

The plug-in 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 plug-ins by
separating their names with commas. In addition, you can match multiple plug-in names or shorten long ones by using glob expressions:

```
~]$ yum update --disableplugin=presto,refresh-pack*
```

### 5.4.2. Installing Additional Yum Plug-ins

Yum plug-ins usually adhere to the `yum-plugin-plugin_name` package-naming convention, but not always: the package which provides the `presto` plug-in is named `yum-presto`, for example. You can install a Yum plug-in in the same way you install other packages. For instance, to install the `security` plug-in, type the following at a shell prompt:

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

### 5.4.3. Plug-in Descriptions

The following list provides descriptions of a few useful Yum plug-ins:

**fs-snapshot** *(yum-plugin-fs-snapshot)*

The `fs-snapshot` plug-in extends Yum to create a snapshot of a file system before proceeding with a transaction such as a system update or package removal. When a user decides that the changes made by the transaction are unwanted, this mechanism allows the user to roll back to the changes that are stored in a snapshot.

In order for the plug-in to work, the root file system (that is, `/`) must be on an LVM (Logical Volume Manager) or Btrfs volume. To use the `fs-snapshot` plug-in on an LVM volume, take the following steps:

1. Make sure that the volume group with the root file system has enough free extents. The required size is a function of the amount of changes to the original logical volume that is expected during the life of the snapshot. The reasonable default is 50–80 % of the original logical volume size.

   To display detailed information about a particular volume group, run the `vgdisplay` command in the following form as root:

   ```
   vgdisplay volume_group
   ```

   The number of free extents is listed on the **Free PE / Size** line.

2. If the volume group with the root file system does not have enough free extents, add a new physical volume:

   a. As root, run the `pvcreate` command in the following form to initialize a physical volume for use with the Logical Volume Manager:

   ```
   pvcreate device
   ```

   b. Use the `vgextend` command in the following form as root to add the physical volume to the volume group:

   ```
   vgextend volume_group physical_volume
   ```
3. Edit the configuration file located in `/etc/yum/pluginconf.d/fs-snapshot.conf`, and make the following changes to the `[lvm]` section:

   a. Change the value of the `enabled` option to `1`:

   ```
   enabled = 1
   ```

   b. Remove the hash sign (that is, `#`) from the beginning of the `lvcreate_size_args` line, and adjust the number of logical extents to be allocated for a snapshot. For example, to allocate 80% of the size of the original logical volume, use:

   ```
   lvcreate_size_args = -l 80%ORG
   ```

   Refer to Table 5.3, “Supported `fs-snapshot.conf` directives” for a complete list of available configuration options.

4. Run the desired `yum` command, and make sure `fs-snapshot` is included in the list of loaded plug-ins (the `Loaded plugins` line) before you confirm the changes and proceed with the transaction. The `fs-snapshot` plug-in displays a line in the following form for each affected logical volume:

   ```
   fs-snapshot: snapshotting file_system (/dev/volume_group/logical_volume): logical_volume_yum_timestamp
   ```

5. Verify that the system is working as expected:

   If you decide to keep the changes, remove the snapshot by running the `lvremove` command as root:

   ```
   lvremove /dev/volume_group/logical_volume_yum_timestamp
   ```

   If you decide to revert the changes and restore the file system to a state that is saved in a snapshot, take the following steps:

   a. As root, run the command in the following form to merge a snapshot into its original logical volume:

   ```
   lvconvert --merge /dev/volume_group/logical_volume_yum_timestamp
   ```

   The `lvconvert` command will inform you that a restart is required in order for the changes to take effect.

   b. Restart the system as instructed. You can do so by typing the following at a shell prompt as root:

   ```
   reboot
   ```

To use the `fs-snapshot` plug-in on a Btrfs file system, take the following steps:

1. Run the desired `yum` command, and make sure `fs-snapshot` is included in the list of loaded plug-ins (the `Loaded plugins` line) before you confirm the changes and proceed with the transaction. The `fs-snapshot` plug-in displays a line in the following form for each affected file system:
fs-snapshot: snapshotting file_system: file_system/yum_timestamp

2. Verify that the system is working as expected:

   If you decide to keep the changes, you can optionally remove unwanted snapshots. To remove a Btrfs snapshot, use the command in the following form as root:

```
btrfs subvolume delete file_system/yum_timestamp
```

   If you decide to revert the changes and restore a file system to a state that is saved in a snapshot, take the following steps:

   a. Determine the identifier of a particular snapshot by using the following command as root:

```
btrfs subvolume list file_system
```

   b. As root, configure the system to mount this snapshot by default:

```
btrfs subvolume set-default id file_system
```

   c. Restart the system. You can do so by typing the following at a shell prompt as root:

```
reboot
```

For more information on logical volume management, Btrfs, and file system snapshots, see the `Fedora 20 Storage Administration Guide`
1. For additional information about the plug-in and its configuration, refer to the `yum-fs-snapshot(1)` and `yum-fs-snapshot.conf(5)` manual pages.

Table 5.3. Supported fs-snapshot.conf directives

<table>
<thead>
<tr>
<th>Section</th>
<th>Directive</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[main]</td>
<td>enabled=value</td>
<td>Allows you to enable or disable the plug-in. The value must be either 1 (enabled), or 0 (disabled). When installed, the plug-in is enabled by default.</td>
</tr>
<tr>
<td></td>
<td>exclude=list</td>
<td>Allows you to exclude certain file systems. The value must be a space-separated list of mount points you do not want to snapshot (for example, /srv /mnt/backup). This option is not included in the configuration file by default.</td>
</tr>
<tr>
<td>[lvm]</td>
<td>enabled=value</td>
<td>Allows you to enable or disable the use of the plug-in on LVM volumes. The value must be either 1 (enabled), or 0 (disabled). This option is disabled by default.</td>
</tr>
<tr>
<td></td>
<td>lvcreate_size_args=value</td>
<td>Allows you to specify the size of a logical volume snapshot. The value must be the -</td>
</tr>
</tbody>
</table>

---

Chapter 5. Yum

### Directive

<table>
<thead>
<tr>
<th>Section</th>
<th>Directive</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>-l</code> or <code>-L</code> command line option for the <code>lvcreate</code> utility followed by a valid argument (for example, <code>-l 80%ORIGIN</code>).</td>
<td></td>
</tr>
</tbody>
</table>

**presto** *(yum-presto)*

The *presto* plug-in 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 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 to apply the difference to the currently-installed package and thus create the full, updated package. This process takes CPU time on the installing machine. 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* plug-in 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, that is, by disabling *presto*.

**refresh-packagekit** *(PackageKit-yum-plugin)*

The *refresh-packagekit* plug-in updates metadata for *PackageKit* whenever *yum* is run. The *refresh-packagekit* plug-in is installed by default.

**rhnplugin** *(yum-rhn-plugin)*

The *rhnplugin* provides support for connecting to RHN Classic. This allows systems registered with RHN Classic to update and install packages from this system.

Refer to the *rhnplugin*(8) manual page for more information about the plug-in.

**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* plug-in, which extends *yum* with a set of highly-useful security-related commands, subcommands and options.

You can check for security-related updates as follows:

```
[output truncated]
```

You can then use either *yum update --security* or *yum update-minimal --security* to update those packages which are affected by security advisories. Both of these commands update all packages on the system for which a security advisory has been issued. *yum update-minimal --security* updates them to the latest packages which were released as part of a security advisory, while *yum update --security* will update all packages affected by a security advisory to the latest version of that package available.
In other words, if:

- the `kernel-2.6.38.4-20` package is installed on your system;
- the `kernel-2.6.38.6-22` package was released as a security update;
- then `kernel-2.6.38.6-26` was released as a bug fix update,

...then `yum update-minimal --security` will update you to `kernel-2.6.38.6-22`, and `yum update --security` will update you to `kernel-2.6.38.6-26`. Conservative system administrators may want to use `update-minimal` to reduce the risk incurred by updating packages as much as possible.

Refer to the `yum-security(8)` manual page for usage details and further explanation of the enhancements the `security` plug-in adds to `yum`.

### 5.5. Additional Resources

[http://yum.baseurl.org/wiki/Guides](http://yum.baseurl.org/wiki/Guides)

The Yum Guides section of the Yum wiki contains more documentation.
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 6.3, "PackageKit Architecture".

6.1. Updating Packages with Software Update

You can open Software Updates by clicking Applications → System Tools → Software Update from the Activities menu, 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.

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.

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.

### 6.1.1. Setting the Update-Checking Interval

Selecting Applications → Other → Software Updates from the Activities menu opens the Software Update Preferences window. The Update Settings tab allows you to 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. 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.

![Figure 6.2. Setting PackageKit's update-checking interval](image)

### 6.1.2. Setting the Software Sources

To select which package repositories to use to install software updates, select Applications → Other → Software Updates from the Activities menu, and click the Software Sources tab of the Software Update Preferences window.
Using Add/Remove Software

Figure 6.3. Setting PackageKit’s software sources

PackageKit refers to Yum repositories as software sources. It obtains all packages from enabled software sources. The Software Sources tab 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.

Showing source RPM, test, and debuginfo repositories

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

6.2. 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, select Applications → System Tools → Add/Remove Software from the Activities menu, or run the gpk-application command at the shell prompt.
6.2.1. Refreshing Software Sources (Yum Repositories)

To enable or disable a Yum repository, open a dialog box by sclicking System → Software Sources, and select the Software Sources tab. Refer to Section 6.1.2, “Setting the Software Sources” for more information on available configuration options.

After enabling and/or disabling the correct Yum repositories, make sure 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, that is, Yum repositories.

6.2.2. Finding Packages with Filters

You can view the list of all configured and unfiltered (see below) Yum repositories by opening Add/Remove Software and clicking System → Software Sources. 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 criterion. 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.
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.

The two remaining filters with submenus are:

**Graphical**

Narrows the search to either applications which provide a GUI interface (**Only Graphical**) or those that do not. 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](https://fedoraproject.org/wiki/Licensing#SoftwareLicenses) for details on approved licenses.

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):

---

1 https://fedoraproject.org/wiki/Licensing#SoftwareLicenses
• `package-devel`

• `package-libs`

• `package-libs-devel`

• `package-debuginfo`

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

Using the Only Newest Packages filter

Checking **Only Newest Packages** 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-1.10-32.1.el6.noarch.rpm`) are never filtered out by checking **Only Native Packages**. This filter has no affect on non-multilib systems, such as x86 machines.

### 6.2.3. 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; and the size of the package, which is relevant when we download and install it.
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, a colorful and enhanced version of the top process viewer, by opening a shell prompt and entering:

```bash
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 Available 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.
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.

6.2.4. 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.
6.2.5. Viewing the Transaction Log

PackageKit maintains a log of the transactions 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 Packages 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 Add/Remove Software, or Update System). 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.

Typing the name of a package in the top text entry field filters the list of transactions to those which affected that package.
6.3. 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 to perform the actual transactions, such as installing and removing packages, etc.

Table 6.1, “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.

<table>
<thead>
<tr>
<th>Window Title</th>
<th>Function</th>
<th>How to Open</th>
<th>Shell Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add/Remove Software</td>
<td>Install, remove or view package info</td>
<td>From the GNOME panel: System → Administration → Add/Remove Software</td>
<td>gpk-application</td>
</tr>
<tr>
<td>Software Update</td>
<td>Perform package updates</td>
<td>From the GNOME panel: System → Administration → Software Update</td>
<td>gpk-update-viewer</td>
</tr>
<tr>
<td>Software Sources</td>
<td>Enable and disable Yum repositories</td>
<td>From Add/Remove Software: System → Software Sources</td>
<td>gpk-repo</td>
</tr>
<tr>
<td>Software Log Viewer</td>
<td>View the transaction log</td>
<td>From Add/Remove Software: System → Software Log</td>
<td>gpk-log</td>
</tr>
<tr>
<td>Software Update Preferences</td>
<td>Set PackageKit preferences</td>
<td></td>
<td>gpk-prefs</td>
</tr>
<tr>
<td>(Notification Area Alert)</td>
<td>Alerts you when updates are available</td>
<td>From the GNOME panel: System → Preferences → Startup Applications, Startup Programs tab</td>
<td>gpk-update-icon</td>
</tr>
</tbody>
</table>

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 system. On Linux systems other than Red Hat Enterprise Linux and 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

---

\(^2\) 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. Daemons respond to the systemctl command and can be turned on or off permanently by using the systemctl enable or systemctl disable commands. They can typically be recognized by a “d” appended to their name, such as the packagekitd daemon. Refer to Chapter 7, Services and Daemons for information about system services.
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 front end called pkcon.

### 6.4. Additional Resources

**PackageKit** home page — [http://www.packagekit.org/index.html](http://www.packagekit.org/index.html)

Information about and mailing lists for PackageKit.

**PackageKit** FAQ — [http://www.packagekit.org/pk-faq.html](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](http://www.packagekit.org/pk-matrix.html)

Cross-reference PackageKit-provided features with the long list of package manager back ends.
Part III. Infrastructure Services

This part provides information on how to configure services and daemons, configure authentication, and enable remote logins.
Services and Daemons

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.

This chapter covers the configuration of the services to be run when a system is started, and provides information on how to start, stop, and restart the services on the command line using the `systemctl` utility.

Keep the system secure

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. For more information, refer to the Fedora 20 Security Guide.

7.1. Configuring Services

To allow you to configure which services are started at boot time, Fedora is shipped with the `systemctl` command line tool.

Do not use the ntsysv and chkconfig utilities

Although it is still possible to use the `ntsysv` and `chkconfig` utilities to manage services that have init scripts installed in the `/etc/rc.d/init.d/` directory, it is advised that you use the `systemctl` utility.

Enabling the irqbalance service

To ensure optimal performance on POWER architecture, it is recommended that the `irqbalance` service is enabled. In most cases, this service is installed and configured to run during the Fedora 20 installation. To verify that `irqbalance` is running, type the following at a shell prompt:

```
systemctl status irqbalance.service
```

7.1.1. Enabling the Service

To configure a service to be automatically started at boot time, use the `systemctl` command in the following form:
Chapter 7. Services and Daemons

7.1. Enabling the Service

The service will be started the next time you boot the system. For information on how to start the service immediately, refer to Section 7.2.2, “Running the Service”.

Example 7.1. Enabling the httpd service
Imagine you want to run the Apache HTTP Server on your system. Provided that you have the httpd package installed, you can enable the httpd service by typing the following at a shell prompt as root:

```bash
~]# systemctl enable httpd.service
```

7.1.2. Disabling the Service

To disable starting a service at boot time, use the `systemctl` command in the following form:

```bash
systemctl disable service_name.service
```

The next time you boot the system, the service will not be started. For information on how to stop the service immediately, refer to Section 7.2.3, “Stopping the Service”.

Example 7.2. Disabling the telnet service
In order to secure the system, users are advised to disable insecure connection protocols such as Telnet. You can make sure that the telnet service is disabled by running the following command as root:

```bash
~]# systemctl disable telnet.service
```

7.2. Running Services

The `systemctl` utility also allows you to determine the status of a particular service, as well as to start, stop, or restart a service.

**Do not use the service utility**

Although it is still possible to use the `service` utility to manage services that have init scripts installed in the `/etc/rc.d/init.d/` directory, it is advised that you use the `systemctl` utility.

7.2.1. Checking the Service Status

To determine the status of a particular service, use the `systemctl` command in the following form:

```bash
systemctl status service_name.service
```

This command provides detailed information on the service’s status. However, if you merely need to verify that a service is running, you can use the `systemctl` command in the following form instead:
Checking the Service Status

Example 7.3. Checking the status of the httpd service

Example 7.1, “Enabling the httpd service” illustrated how to enable starting the httpd service at boot time. Imagine that the system has been restarted and you need to verify that the service is really running. You can do so by typing the following at a shell prompt:

```
$ systemctl is-active httpd.service
active
```

You can also display detailed information about the service by running the following command:

```
$ systemctl status httpd.service
```

To display a list of all active system services, use the following command:

```
$ systemctl list-units --type=service
```

This command provides a tabular output with each line consisting of the following columns:

- **UNIT** — A systemd unit name. In this case, a service name.
- **LOAD** — Information whether the systemd unit was properly loaded.
- **ACTIVE** — A high-level unit activation state.
- **SUB** — A low-level unit activation state.
- **JOB** — A pending job for the unit.
- **DESCRIPTION** — A brief description of the unit.

Example 7.4. Listing all active services

You can list all active services by using the following command:

```
$ systemctl list-units --type=service
```

<table>
<thead>
<tr>
<th>UNIT</th>
<th>LOAD</th>
<th>ACTIVE</th>
<th>SUB</th>
<th>JOB</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>abrt-ccpp.service</td>
<td>loaded</td>
<td>active</td>
<td>exited</td>
<td>LSB: Installs coredump handler which saves segfault data</td>
<td></td>
</tr>
<tr>
<td>abrt-oops.service</td>
<td>loaded</td>
<td>active</td>
<td>running</td>
<td>LSB: Watches system log for oops messages, creates ABRT dump directories for each oops</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 7. Services and Daemons

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abrtd.service</td>
<td>loaded</td>
<td>ABRT Automated Bug Reporting Tool</td>
</tr>
<tr>
<td>accounts-daemon.service</td>
<td>loaded</td>
<td>Accounts Service</td>
</tr>
<tr>
<td>atd.service</td>
<td>loaded</td>
<td>Job spooling tools</td>
</tr>
</tbody>
</table>

[output truncated]

In the example above, the abrtd service is loaded, active, and running, and it does not have any pending jobs.

### 7.2.2. Running the Service

To run a service, use the `systemctl` command in the following form:

```
systemctl start service_name.service
```

This will start the service in the current session. To configure the service to be started at boot time, refer to Section 7.1.1, “Enabling the Service”.

Example 7.5. Running the httpd service

*Example 7.1, “Enabling the httpd service”* illustrated how to run the httpd service at boot time. You can start the service immediately by typing the following at a shell prompt as `root`:

```
~ ]# systemctl start httpd.service
```

### 7.2.3. Stopping the Service

To stop a service, use the `systemctl` command in the following form:

```
systemctl stop service_name.service
```

This will stop the service in the current session. To disable starting the service at boot time, refer to Section 7.1.1, “Enabling the Service”.

Example 7.6. Stopping the telnet service

*Example 7.2, “Disabling the telnet service”* illustrated how to disable starting the telnet service at boot time. You can stop the service immediately by running the following command as `root`:

```
~ ]# systemctl stop telnet.service
```

### 7.2.4. Restarting the Service

To restart a service, use the `systemctl` command in the following form:

```
systemctl restart service_name.service
```

Example 7.7. Restarting the sshd service

For any changes in the `/etc/ssh/sshd_config` configuration file to take effect, it is required that you restart the sshd service. You can do so by typing the following at a shell prompt as `root`:

```
~ ]# systemctl restart sshd.service
```
7.3. Additional Resources

7.3.1. Installed Documentation

• systemctl(1) — The manual page for the systemctl utility.

7.3.2. Related Books

Fedora 20 Security Guide

A guide to securing Fedora. It contains valuable information on how to set up the firewall, as well as the configuration of SELinux.
Configuring Authentication

Authentication is the way that a user is identified and verified to a system. The authentication process requires presenting some sort of identity and credentials, like a username and password. The credentials are then compared to information stored in some data store on the system. In Fedora, the Authentication Configuration Tool helps configure what kind of data store to use for user credentials, such as LDAP.

For convenience and potentially part of single sign-on, Fedora can use a central daemon to store user credentials for a number of different data stores. The System Security Services Daemon (SSSD) can interact with LDAP, Kerberos, and external applications to verify user credentials. The Authentication Configuration Tool can configure SSSD along with NIS, Winbind, and LDAP, so that authentication processing and caching can be combined.

8.1. Configuring System Authentication

When a user logs into a Fedora system, that user presents some sort of credential to establish the user identity. The system then checks those credentials against the configured authentication service. If the credentials match and the account is active, then the user is authenticated. (Once a user is authenticated, then the information is passed to the access control service to determine what the user is permitted to do. Those are the resources the user is authorized to access.)

The information to verify the user can be located on the local system or the local system can reference a user database on a remote system, such as LDAP or Kerberos.

The system must have a configured list of valid account databases for it to check for user authentication. On Fedora, the Authentication Configuration Tool has both GUI and command-line options to configure any user data stores.

A local system can use a variety of different data stores for user information, including Lightweight Directory Access Protocol (LDAP), Network Information Service (NIS), and Winbind. Additionally, both LDAP and NIS data stores can use Kerberos to authenticate users.

Important

If a medium or high security level is set during installation or with the Security Level Configuration Tool, then the firewall prevents NIS authentication. For more information about firewalls, see the "Firewalls" section of the Security Guide.

8.1.1. Launching the Authentication Configuration Tool UI

1. Log into the system as root.
2. Open the System.
3. Select the Administration menu.
4. Select the Authentication item.
Alternatively, run the `system-config-authentication` command.

**Important**

Any changes take effect immediately when the Authentication Configuration Tool UI is closed.

There are two configuration tabs in the Authentication dialog box:

- **Identity & Authentication**, which configures the resource used as the identity store (the data repository where the user IDs and corresponding credentials are stored).
- **Advanced Options**, which allows authentication methods other than passwords or certificates, like smart cards and fingerprint.

### 8.1.2. Selecting the Identity Store for Authentication

The **Identity & Authentication** tab sets how users should be authenticated. The default is to use local system authentication, meaning the users and their passwords are checked against local system accounts. A Fedora machine can also use external resources which contain the users and credentials, including LDAP, NIS, and Winbind.

Figure 8.1. Local Authentication

#### 8.1.2.1. Configuring LDAP Authentication

Either the `openldap-clients` package or the `sssd` package is used to configure an LDAP server for the user database. Both packages are installed by default.

1. Open the Authentication Configuration Tool, as in Section 8.1.1, “Launching the Authentication Configuration Tool UI”.

2. Select **LDAP** in the **User Account Database** drop-down menu.

3. Set the information that is required to connect to the LDAP server.

   - **LDAP Search Base DN** gives the root suffix or *distinguished name* (DN) for the user directory. All of the user entries used for identity/authentication will exist below this parent entry. For example, `ou=people,dc=example,dc=com`.

     This field is optional. If it is not specified, then SSSD attempts to detect the search base using the `namingContexts` and `defaultNamingContext` attributes in the LDAP server's configuration entry.

   - **LDAP Server** gives the URL of the LDAP server. This usually requires both the hostname and port number of the LDAP server, such as `ldap://ldap.example.com:389`.
Selecting the Identity Store for Authentication

Entering the secure protocol in the URL, ldaps://, enables the Download CA Certificate button.

• **Use TLS to encrypt connections** sets whether to use Start TLS to encrypt the connections to the LDAP server. This enables a secure connection over a standard port.

Selecting TLS enables the Download CA Certificate button, which retrieves the issuing CA certificate for the LDAP server from whatever certificate authority issued it. The CA certificate must be in the privacy enhanced mail (PEM) format.

---

**Important**

*Do not* select **Use TLS to encrypt connections** if the server URL uses a secure protocol (ldaps). This option uses Start TLS, which initiates a secure connection over a standard port; if a secure port is specified, then a protocol like SSL must be used instead of Start TLS.

---

4. Select the authentication method. LDAP allows simple password authentication or Kerberos authentication.

Using Kerberos is described in Section 8.1.2.4, “Using Kerberos with LDAP or NIS Authentication”.

The **LDAP password** option uses PAM applications to use LDAP authentication. This option requires either a secure (ldaps://) URL or the TLS option to connect to the LDAP server.

### 8.1.2.2. Configuring NIS Authentication

1. Install the **ypbind** package. This is required for NIS services, but is not installed by default.

   ```bash
   [root@server ~]# yum install ypbind
   ```

   When the **ypbind** service is installed, the portmap and ypbind services are started and enabled to start at boot time.

2. Open the Authentication Configuration Tool, as in Section 8.1.1, “Launching the Authentication Configuration Tool UI”.

3. Select **NIS** in the User Account Database drop-down menu.

4. Set the information to connect to the NIS server, meaning the NIS domain name and the server hostname. If the NIS server is not specified, the authconfig daemon scans for the NIS server.

5. Select the authentication method. NIS allows simple password authentication or Kerberos authentication.

   Using Kerberos is described in Section 8.1.2.4, “Using Kerberos with LDAP or NIS Authentication”.

For more information about NIS, see the “Securing NIS” section of the Security Guide.
8.1.2.3. Configuring Winbind Authentication

Using Winbind as an authentication provider requires the `samba-winbind` package, which is installed by default.

1. Open the Authentication Configuration Tool, as in Section 8.1.1, "Launching the Authentication Configuration Tool UI".

2. Select **Winbind** in the **User Account Database** drop-down menu.

3. Set the information that is required to connect to the Microsoft Active Directory domain controller.
   - **Winbind Domain** gives the Windows domain to connect to.
     - This should be in the Windows 2000 format, such as **DOMAIN**.
   - **Security Model** sets the security model to use for Samba clients. `authconfig` supports four types of security models:
     - **ads** configures Samba to act as a domain member in an Active Directory Server realm. To operate in this mode, the `krb5-server` package must be installed and Kerberos must be configured properly.
     - **domain** has Samba validate the username/password by authenticating it through a Windows primary or backup domain controller, much like a Windows server.
     - **server** has a local Samba server validate the username/password by authenticating it through another server, such as a Windows server. If the server authentication attempt fails, the system then attempts to authentication using **user** mode.
     - **user** requires a client to log in with a valid username and password. This mode does support encrypted passwords.
   
   The username format must be **domain\user**, such as **EXAMPLE\jsmith**.

   **Note**

   When verifying that a given user exists in the Windows domain, always use Windows 2000-style formats and escape the backslash (\) character. For example:

   ```bash
   [root@server ~]# getent passwd domain\\user DOMAIN\user:*:16777216:16777216:Name Surname:/home/DOMAIN/user:/bin/bash
   ```

   This is the default option.

   - **Winbind ADS Realm** gives the Active Directory realm that the Samba server will join. This is only used with the **ads** security model.

   - **Winbind Domain Controllers** gives the domain controller to use. For more information about domain controllers, refer to Section 13.1.6.3, "Domain Controller".

   - **Template Shell** sets which login shell to use for Windows user account settings.
• **Allow offline login** allows authentication information to be stored in a local cache. The cache is referenced when a user attempts to authenticate to system resources while the system is offline.

For more information about the *winbindd* service, refer to *Section 13.1.2, “Samba Daemons and Related Services”*.

### 8.1.2.4. Using Kerberos with LDAP or NIS Authentication

Both LDAP and NIS authentication stores support Kerberos authentication methods. Using Kerberos has a couple of benefits:

• It uses a security layer for communication while still allowing connections over standard ports.
• It automatically uses credentials caching with SSSD, which allows offline logins.

Using Kerberos authentication requires the *krb5-libs* and *krb5-workstation* packages.

The **Kerberos password** option from the Authentication Method drop-down menu automatically opens the fields required to connect to the Kerberos realm.
Figure 8.2. Kerberos Fields

- **Realm** gives the name for the realm for the Kerberos server. The realm is the network that uses Kerberos, composed of one or more key distribution centers (KDC) and a potentially large number of clients.
• **KDCs** gives a comma-separated list of servers that issue Kerberos tickets.

• **Admin Servers** gives a list of administration servers running the **kadmin** process in the realm.

• Optionally, use DNS to resolve server hostname and to find additional KDCs within the realm.

For more information about Kerberos, refer to section “Using Kerberos” of the Fedora 20 *Managing Single Sign-On and Smart Cards* guide.

### 8.1.3. Configuring Alternative Authentication Features

The Authentication Configuration Tool also configures settings related to authentication behavior, apart from the identity store. This includes entirely different authentication methods (fingerprint scans and smart cards) or local authentication rules. These alternative authentication options are configured in the **Advanced Options** tab.
8.1.3.1. Using Fingerprint Authentication

When there is appropriate hardware available, the Enable fingerprint reader support option allows fingerprint scans to be used to authenticate local users in addition to other credentials.
8.1.3.2. Setting Local Authentication Parameters

There are two options in the **Local Authentication Options** area which define authentication behavior on the local system:

- **Enable local access control** instructs the `/etc/security/access.conf` file to check for local user authorization rules.
- **Password Hashing Algorithm** sets the hashing algorithm to use to encrypt locally-stored passwords.

8.1.3.3. Enabling Smart Card Authentication

When there are appropriate smart card readers available, a system can accept smart cards (or tokens) instead of other user credentials to authenticate.

Once the **Enable smart card support** option is selected, then the behaviors of smart card authentication can be defined:

- **Card Removal Action** tells the system how to respond when the card is removed from the card reader during an active session. A system can either ignore the removal and allow the user to access resources as normal, or a system can immediately lock until the smart card is supplied.
- **Require smart card login** sets whether a smart card is required for logins or simply allowed for logins. When this option is selected, all other methods of authentication are immediately blocked.

**Warning**

Do not select this option until you have successfully authenticated to the system using a smart card.

Using smart cards requires the `pam_pkcs11` package.

8.1.3.4. Creating User Home Directories

There is an option (**Create home directories on the first login**) to create a home directory automatically the first time that a user logs in.

This option is beneficial with accounts that are managed centrally, such as with LDAP. However, this option should not be selected if a system like automount is used to manage user home directories.

8.1.4. Setting Password Options

Through the Authentication Configuration Tool, it is possible to set requirements of passwords and the classes of characters it will contain. Also, it can determine how consecutive characters will be repeated within a password. This options are configured in the **Password Options** tab.
The **Length** option allows a maximum of 30 characters and the **Character Classes** option allows to choose from one to four according the classes: **Lowercase**, **Uppercase**, **Digits** and **Other characters** which are going to be required to create a password if they are enabled.
Important

An adequate election to the length of password must be done correspondingly to the classes chosen. For example, the password `FedOrAworld` which has two character classes contains eleven characters and the password `Fed2rA1` contains seven characters but has three character classes.

The maximum amount of consecutive character repetition is set to characters and classes. Both, the Same Character option and the Same Class option have the length of 30 as character repetition allowed.

8.1.5. Configuring Authentication from the Command Line

The `authconfig` command-line tool updates all of the configuration files and services required for system authentication, according to the settings passed to the script. Along with allowing all of the identity and authentication configuration options that can be set through the UI, the `authconfig` tool can also be used to create backup and kickstart files.

For a complete list of `authconfig` options, check the help output and the man page.

8.1.5.1. Tips for Using authconfig

There are some things to remember when running `authconfig`:

- With every command, use either the `--update` or `--test` option. One of those options is required for the command to run successfully. Using `--update` writes the configuration changes. `--test` prints the changes to stdout but does not apply the changes to the configuration.

- Each enable option has a corresponding disable option.

8.1.5.2. Configuring LDAP User Stores

To use an LDAP identity store, use the `--enableldap`. To use LDAP as the authentication source, use `--enableldapauth` and then the requisite connection information, like the LDAP server name, base DN for the user suffix, and (optionally) whether to use TLS. The `authconfig` command also has options to enable or disable RFC 2307bis schema for user entries, which is not possible through the Authentication Configuration UI.

Be sure to use the full LDAP URL, including the protocol (ldap or ldaps) and the port number. Do not use a secure LDAP URL (ldaps) with the `--enableldaptls` option.

```
```

Instead of using `--ldapauth` for LDAP password authentication, it is possible to use Kerberos with the LDAP user store. These options are described in Section 8.1.5.5, “Configuring Kerberos Authentication”.

8.1.5.3. Configuring NIS User Stores

To use a NIS identity store, use the `--enablenis`. This automatically uses NIS authentication, unless the Kerberos parameters are explicitly set, so it uses Kerberos authentication (Section 8.1.5.5, “Configuring Kerberos Authentication”). The only parameters are to identify the NIS server and NIS domain; if these are not used, then the authconfig service scans the network for NIS servers.

```
authconfig --enablenis --nisdomain=EXAMPLE --nisserver=nis.example.com --update
```

8.1.5.4. Configuring Winbind User Stores

Windows domains have several different security models, and the security model used in the domain determines the authentication configuration for the local system.

For user and server security models, the Winbind configuration requires only the domain (or workgroup) name and the domain controller hostnames.

```
authconfig --enablewinbind --enablewinbindauth --smbsecurity=user|server --enablewinbindoffline --smbservers=ad.example.com --smbworkgroup=EXAMPLE --update
```

*Note*

The username format must be `domain\user`, such as `EXAMPLE\jsmith`.

When verifying that a given user exists in the Windows domain, always use Windows 2000-style formats and escape the backslash (`\`) character. For example:

```
[root@server ~]# getent passwd domain\user
DOMAIN\user:*:16777216:16777216:Name Surname:/home/DOMAIN/user:/bin/bash
```

For ads and domain security models, the Winbind configuration allows additional configuration for the template shell and realm (ads only). For example:

```
authconfig --enablewinbind --enablewinbindauth --smbsecurity ads --enablewinbindoffline --smbservers=ad.example.com --smbworkgroup=EXAMPLE --smbrealm EXAMPLE.COM --winbindtemplateshell=/bin/sh --update
```

There are a lot of other options for configuring Windows-based authentication and the information for Windows user accounts, such as name formats, whether to require the domain name with the username, and UID ranges. These options are listed in the `authconfig` help.

8.1.5.5. Configuring Kerberos Authentication

Both LDAP and NIS allow Kerberos authentication to be used in place of their native authentication mechanisms. At a minimum, using Kerberos authentication requires specifying the realm, the KDC, and the administrative server. There are also options to use DNS to resolve client names and to find additional admin servers.

```
authconfig NIS or LDAP options --enablekrb5 --krb5realm EXAMPLE --krb5kdc kdc.example.com:88,server.example.com:88 --krb5adminserver server.example.com:749 --enablekrb5kcdns --enablekrb5realmdns --update
```
8.1.5.6. Configuring Local Authentication Settings
The Authentication Configuration Tool can also control some user settings that relate to security, such as creating home directories, setting password hash algorithms, and authorization. These settings are done independently of identity/user store settings.

For example, to create user home directories:

    authconfig --enablemkhomedir --update

To set or change the hash algorithm used to encrypt user passwords:

    authconfig --passalgo=sha512 --update

8.1.5.7. Configuring Fingerprint Authentication
There is one option to enable support for fingerprint readers. This option can be used alone or in conjunction with other authconfig settings, like LDAP user stores.

    authconfig --enablefingerprint --update

8.1.5.8. Configuring Smart Card Authentication
All that is required to use smart cards with a system is to set the --enablersmartcard option:

    authconfig --enablersmartcard --update

There are other configuration options for smart cards, such as changing the default smart card module, setting the behavior of the system when the smart card is removed, and requiring smart cards for login.

For example, this command instructs the system to lock out a user immediately if the smart card is removed (a setting of 1 ignores it if the smart card is removed):

    authconfig --enablersmartcard --smartcardaction=0 --update

Once smart card authentication has been successfully configured and tested, then the system can be configured to require smart card authentication for users rather than simple password-based authentication.

    authconfig --enablersmartcard --update

**Warning**

Do not use the --enablersmartcard option until you have successfully authenticated to the system using a smart card. Otherwise, users may be unable to log into the system.

8.1.5.9. Managing Kickstart and Configuration Files
The --update option updates all of the configuration files with the configuration changes. There are a couple of alternative options with slightly different behavior:
• **--kickstart** writes the updated configuration to a kickstart file.

• **--test** prints the full configuration, with changes, to stdout but does not edit any configuration files.

Additionally, `authconfig` can be used to back up and restore previous configurations. All archives are saved to a unique subdirectory in the `/var/lib/authconfig/` directory. For example, the **--savebackup** option gives the backup directory as **2011-07-01**:

```
authconfig --savebackup=2011-07-01
```

This backs up all of the authentication configuration files beneath the `/var/lib/authconfig/backup-2011-07-01` directory.

Any of the saved backups can be used to restore the configuration using the **--restorebackup** option, giving the name of the manually-saved configuration:

```
authconfig --restorebackup=2011-07-01
```

Additionally, `authconfig` automatically makes a backup of the configuration before it applies any changes (with the **--update** option). The configuration can be restored from the most recent automatic backup, without having to specify the exact backup, using the **--restorelastbackup** option.

### 8.1.6. Using Custom Home Directories

If LDAP users have home directories that are not in `/home` and the system is configured to create home directories the first time users log in, then these directories are created with the wrong permissions.

1. Apply the correct SELinux context and permissions from the `/home` directory to the home directory that is created on the local system. For example:

   ```
   # semanage fcontext -a -e /home /home/locale
   ```

2. Install the `oddjob-mkhomedir` package on the system.

   This package provides the `pam_oddjob_mkhomedir.so` library, which the Authentication Configuration Tool uses to create home directories. The `pam_oddjob_mkhomedir.so` library, unlike the default `pam_mkhomedir.so` library, can create SELinux labels.

   The Authentication Configuration Tool automatically uses the `pam_oddjob_mkhomedir.so` library if it is available. Otherwise, it will default to using `pam_mkhomedir.so`.

3. Make sure the `oddjobd` service is running.

4. Re-run the Authentication Configuration Tool and enable home directories, as in Section 8.1.3, “Configuring Alternative Authentication Features”.

If home directories were created before the home directory configuration was changed, then correct the permissions and SELinux contexts. For example:

```
# semanage fcontext -a -e /home /home/locale
# restorecon -R -v /home/locale
```
8.2. Using and Caching Credentials with SSSD

The System Security Services Daemon (SSSD) provides access to different identity and authentication providers. SSSD is an intermediary between local clients and any configured data store. The local clients connect to SSSD and then SSSD contacts the external providers. This brings a number of benefits for administrators:

- **Reducing the load on identification/authentication servers.** Rather than having every client service attempt to contact the identification server directly, all of the local clients can contact SSSD which can connect to the identification server or check its cache.

- **Permitting offline authentication.** SSSD can optionally keep a cache of user identities and credentials that it retrieves from remote services. This allows users to authenticate to resources successfully, even if the remote identification server is offline or the local machine is offline.

- **Using a single user account.** Remote users frequently have two (or even more) user accounts, such as one for their local system and one for the organizational system. This is necessary to connect to a virtual private network (VPN). Because SSSD supports caching and offline authentication, remote users can connect to network resources simply by authenticating to their local machine and then SSSD maintains their network credentials.

The System Security Services Daemon does not require any additional configuration or tuning to work with the Authentication Configuration Tool. However, SSSD can work with other applications, and the daemon may require configuration changes to improve the performance of those applications.

8.2.1. About the sssd.conf File

SSSD services and domains are configured in a `.conf` file. The default file is `/etc/sssd/sssd.conf`, although alternative files can be passed to SSSD by using the `-c` option with the `sssd` command:

```bash
# sssd -c /etc/sssd/customfile.conf
```

Both services and domains are configured individually, in separate sections on the configuration identified by `[type/name]` divisions, such as `[domain/LDAP]`. The configuration file uses simple `key = value` lines to set the configuration. Comment lines are set by either a hash sign (`#`) or a semicolon (`;`). For example:

```plaintext
[section]
# Comment line
key1 = val1
key10 = val1,val2
```

8.2.2. Starting and Stopping SSSD

Configure at least one domain before starting SSSD for the first time. See Section 8.2.4, "Creating Domains".

Either the `service` command or the `/etc/init.d/sssd` script can start SSSD. For example:
# service sssd start

By default, SSSD is configured not to start automatically. There are two ways to change this behavior:

- Using the `authconfig` command:
  ```bash
  [root@server ~]# authconfig --enablesssd --enablessdauth --update
  ```

- Using the `chkconfig` command:
  ```bash
  [root@server ~]# chkconfig sssd on
  ```

## 8.2.3. Configuring SSSD to Work with System Services

SSSD worked with specialized services that run in tandem with the SSSD process itself. SSSD and its associated services are configured in the `sssd.conf` file. The `[sssd]` section also lists the services that are active and should be started when `sssd` starts within the `services` directive.

SSSD currently provides several services:

- A Name Service Switch (NSS) provider service that answers name service requests from the `sssd_nss` module. This is configured in the `[nss]` section of the SSSD configuration.

- A PAM provider service that manages a PAM conversation through the `sssd_pam` module. This is configured in the `[pam]` section of the configuration.

- `monitor`, a special service that monitors and starts or restarts all other SSSD services. Its options are specified in the `[sssd]` section of the `/etc/sssd/sssd.conf` configuration file.

### Note

If a DNS lookup fails to return an IPv4 address for a hostname, SSSD attempts to look up an IPv6 address before returning a failure. This only ensures that the asynchronous resolver identifies the correct address.

The hostname resolution behavior is configured in the `lookup family order` option in the `sssd.conf` configuration file.

### 8.2.3.1. Configuring NSS Services

SSSD provides an NSS module, `sssd_nss`, which instructs the system to use SSSD to retrieve user information. The NSS configuration must include a reference to the SSSD module, and then the SSSD configuration sets how SSSD interacts with NSS.

#### 8.2.3.1.1. About NSS Service Maps and SSSD

The Name Service Switch (NSS) provides a central configuration for services to look up a number of configuration and name resolution services. NSS provides one method of mapping system identities and services with configuration sources.

SSSD works with NSS as a provider services for several types of NSS maps:

- Passwords (`passwd`)
8.2.3.1.2. Configuring NSS Services to Use SSSD

NSS can use multiple identity and configuration providers for any and all of its service maps. The default is to use system files for services; for SSSD to be included, the **nss_sss** module has to be included for the desired service type.

1. Use the Authentication Configuration tool to enable SSSD. This automatically configured the **nsswitch.conf** file to use SSSD as a provider.

   ```sh
   [root@server ~]# authconfig --enablesssd --update
   ```

   This automatically configures the password, shadow, group, and netgroups services maps to use the SSSD module:

   ```
   passwd:     files sss
   shadow:     files sss
   group:      files sss
   netgroup:   files sss
   ```

2. The services map is not enabled by default when SSSD is enabled with **authconfig**. To include that map, open the **nsswitch.conf** file and add the **sss** module to the **services** map:

   ```sh
   [root@server ~]# vim /etc/nsswitch.conf
   ...
   services: file sss
   ...
   ```

8.2.3.1.3. Configuring SSSD to Work with NSS

The options and configuration that SSSD uses to service NSS requests are configured in the SSSD configuration file, in the [nss] services section.

1. Open the **sssd.conf** file.

   ```sh
   [root@server ~]# vim /etc/sssd/sssd.conf
   ```

2. Make sure that NSS is listed as one of the services that works with SSSD.

   ```
   [sssd]
   config_file_version = 2
   reconnection_retries = 3
   sbus_timeout = 30
   services = sss, pam
   ```

3. In the [nss] section, change any of the NSS parameters. These are listed in **Table 8.1, “SSSD [nss] Configuration Parameters”**.
Chapter 8. Configuring Authentication

4. Restart SSSD.

```
[root@server ~]# service sssd restart
```

Table 8.1. SSSD [nss] Configuration Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enum_cache_timeout</td>
<td>integer</td>
<td>Specifies how long, in seconds, <code>sssd_nss</code> should cache requests for information about all users (enumerations).</td>
</tr>
<tr>
<td>entry_cache_nowait_percentage</td>
<td>integer</td>
<td>Specifies how long <code>sssd_nss</code> should return cached entries before refreshing the cache. Setting this to zero (0) disables the entry cache refresh. This configures the entry cache to update entries in the background automatically if they are requested if the time before the next update is a certain percentage of the next interval. For example, if the interval is 300 seconds and the cache percentage is 75, then the entry cache will begin refreshing when a request comes in at 225 seconds — 75% of the interval. The allowed values for this option are 0 to 99, which sets the percentage based on the <code>entry_cache_timeout</code> value. The default value is 50%.</td>
</tr>
<tr>
<td>entry_negative_timeout</td>
<td>integer</td>
<td>Specifies how long, in seconds, <code>sssd_nss</code> should cache <em>negative</em> cache hits. A negative cache hit is a query for an invalid database entries, including non-existent entries.</td>
</tr>
<tr>
<td>filter_users, filter_groups</td>
<td>string</td>
<td>Tells SSSD to exclude certain users from being fetched from the NSS database. This is</td>
</tr>
</tbody>
</table>
### Parameter | Value Format | Description
--- | --- | ---
filter_users_in_groups | Boolean | Sets whether users listed in the `filter_users` list appear in group memberships when performing group lookups. If set to `FALSE`, group lookups return all users that are members of that group. If not specified, this value defaults to `true`, which filters the group member lists.
debug_level | integer, 0 - 9 | Sets a debug logging level.

## 8.2.3.2. Configuring the PAM Service

**Warning**

A mistake in the PAM configuration file can lock users out of the system completely. Always back up the configuration files before performing any changes, and keep a session open so that any changes can be reverted.

SSSD provides a PAM module, `sssd_pam`, which instructs the system to use SSSD to retrieve user information. The PAM configuration must include a reference to the SSSD module, and then the SSSD configuration sets how SSSD interacts with PAM.

To configure the PAM service:

1. Use `authconfig` to enable SSSD for system authentication.

   ```bash
   # authconfig --update --enablesssd --enablessdauth
   ```

   This automatically updates the PAM configuration to reference all of the SSSD modules:

   ```
   #!/PAM-1.0
   # This file is auto-generated.
   # User changes will be destroyed the next time authconfig is run.
   auth    required  pam_env.so
   auth    sufficient pam_unix.so nullok try_first_pass
   auth    requisite  pam_succeed_if.so uid >= 500 quiet
   auth    sufficient pam_sss.so use_first_pass
   auth    required  pam_deny.so
   account   required  pam_unix.so
   account   sufficient pam_localuser.so
   account   sufficient  pam_succeed_if.so uid < 500 quiet
   account   [default=bad success=ok user_unknown=ignore] pam_sss.so
   account   required  pam_permit.so
   password   requisite  pam_cracklib.so try_first_pass retry=3
   password   sufficient pam_unix.so sha512 shadow nullok try_first_pass use_authtok
   password   sufficient pam_sss.so use_authtok
   password   required  pam_deny.so
   ```
Chapter 8. Configuring Authentication

These modules can be set to include statements, as necessary.

2. Open the sssd.conf file.

```
# vim /etc/sssd/sssd.conf
```

3. Make sure that PAM is listed as one of the services that works with SSSD.

```
[sssd]
config_file_version = 2
reconnection_retries = 3
sbus_timeout = 30
services = nss, pam
```

4. In the [pam] section, change any of the PAM parameters. These are listed in Table 8.2, “SSSD [pam] Configuration Parameters”.

```
[pam]
reconnection_retries = 3
offline_credentials_expiration = 2
offline_failed_login_attempts = 3
offline_failed_login_delay = 5
```

5. Restart SSSD.

```
[root@server ~]# service sssd restart
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>offline_credentials_expiration</td>
<td>integer</td>
<td>Sets how long, in days, to allow cached logins if the authentication provider is offline. This value is measured from the last successful online login. If not specified, this defaults to zero (0), which is unlimited.</td>
</tr>
<tr>
<td>offline_failed_login_attempts</td>
<td>integer</td>
<td>Sets how many failed login attempts are allowed if the authentication provider is offline. If not specified, this defaults to zero (0), which is unlimited.</td>
</tr>
<tr>
<td>offline_failed_login_delay</td>
<td>integer</td>
<td>Sets how long to prevent login attempts if a user hits the failed login attempt limit. If set</td>
</tr>
</tbody>
</table>
8.2.4. Creating Domains

SSSD recognizes domains, which are associated with the different identity servers. Domains are a combination of an identity provider and an authentication method. SSSD works with LDAP identity providers (including OpenLDAP, Red Hat Directory Server, and Microsoft Active Directory) and can use native LDAP authentication or Kerberos authentication.

As long as they belong to different domains, SSSD can recognize different users with the same username. For example, SSSD can successfully authenticate both jsmith in the ldap.example.com domain and jsmith in the ldap.otherexample.com domain. SSSD allows requests using fully-qualified domain names, so requesting information for jsmith@ldap.example.com returns the proper user account. Specifying only the username returns the user for whichever domain comes first in the lookup order.

**Tip**

SSSD has a `filter_users` option, which excludes the specified users from being returned in a search.

Configuring a domain defines both where user information is stored and how those users are allowed to authenticate to the system. The possible combinations are listed in Table 8.3, "Identity Store and Authentication Type Combinations".

- Section 8.2.4.1, “General Rules and Options for Configuring a Domain”
- Section 8.2.4.2, “Configuring an LDAP Domain”
- Section 8.2.4.3, “Configuring Kerberos Authentication with a Domain”
- Section 8.2.4.4, “Configuring a Proxy Domain”

<table>
<thead>
<tr>
<th>Identification Provider</th>
<th>Authentication Provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP</td>
<td>LDAP</td>
</tr>
<tr>
<td>LDAP</td>
<td>Kerberos</td>
</tr>
<tr>
<td>proxy</td>
<td>LDAP</td>
</tr>
<tr>
<td>proxy</td>
<td>Kerberos</td>
</tr>
<tr>
<td>proxy</td>
<td>proxy</td>
</tr>
</tbody>
</table>

### Table 8.3. Identity Store and Authentication Type Combinations

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>to zero (0), the user cannot authenticate while the provider is offline once he hits the failed attempt limit. Only a successful online authentication can re-enable offline authentication. If not specified, this defaults to five (5).</td>
</tr>
</tbody>
</table>
8.2.4.1. General Rules and Options for Configuring a Domain

A domain configuration defines the identity provider, the authentication provider, and any specific configuration to access the information in those providers. There are two types of identity providers — LDAP and proxy — three types of authentication providers — LDAP, Kerberos, and proxy. The identity and authentication providers can be configured in any combination in a domain entry.

Along with the domain entry itself, the domain name must be added to the list of domains that SSSD will query. For example:

```
domains = LOCAL, Name
[domain/Name]
id_provider = type
auth_provider = type
provider_specific = value
global = value
```

Global attributes are available to any type of domain, such as cache and time out settings. Each identity and authentication provider has its own set of required and optional configuration parameters.

Table 8.4. General [domain] Configuration Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id_provider</td>
<td>string</td>
<td>Specifies the data provider identity backend to use for this domain. The supported identity backends are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ldap</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ipa, compatible with FreeIPA version 2.x and Identity Management in Fedora</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• proxy for a legacy NSS provider, such as nss_nis. Using a proxy ID provider also requires specifying the legacy NSS library to load to start successfully, set in the proxy_lib_name option.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• local, the SSSD internal local provider</td>
</tr>
<tr>
<td>auth_provider</td>
<td>string</td>
<td>Sets the authentication provider used for the domain. The default value for this option is the value of id_provider. The supported authentication providers are ldap, ipa, krb5 (Kerberos), proxy, and none.</td>
</tr>
<tr>
<td>min_id,max_id</td>
<td>integer</td>
<td>Optional. Specifies the UID and GID range for the domain. If a domain contains entries that are outside that range, they are ignored. The default value for min_id is 1; the default</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value Format</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>value for <code>max_id</code> is 0, which is unlimited.</td>
</tr>
<tr>
<td>Important</td>
<td></td>
<td>The default <code>min_id</code> value is the same for all types of identity provider. If LDAP directories are using UID numbers that start at one, it could cause conflicts with users in the local <code>/etc/passwd</code> file. To avoid these conflicts, set <code>min_id</code> to 1000 or higher as possible.</td>
</tr>
<tr>
<td>enumerate</td>
<td>Boolean</td>
<td>Optional. Specifies whether to list the users and groups of a domain. Enumeration means that the entire set of available users and groups on the remote source is cached on the local machine. When enumeration is disabled, users and groups are only cached as they are requested.</td>
</tr>
</tbody>
</table>
### Chapter 8. Configuring Authentication

#### Table of Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>Warning</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>When enumeration is enabled, reinitializing a client results in a complete refresh of the entire set of available users and groups from the remote source. Similarly, when SSSD is connected to a new server, the entire set of available users and groups from the remote source is pulled and cached on the local machine. In a domain with a large number of clients connected to a remote source, this refresh process can harm the network performance because of frequent queries from the clients. If the set of available users and groups is large enough, it degrades client performance as well.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default value for this parameter is <strong>false</strong>, which disables enumeration.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cache_credentials</td>
<td>Boolean</td>
<td><em>Optional.</em> Specifies whether to store user credentials in the local SSSD domain database cache. The default value for this parameter is <strong>false</strong>. Set this value to <strong>true</strong> for domains other than the LOCAL domain to enable offline authentication.</td>
</tr>
<tr>
<td>entry_cache_timeout</td>
<td>integer</td>
<td><em>Optional.</em> Specifies how long, in seconds, SSSD should cache positive cache hits. A positive cache hit is a successful query.</td>
</tr>
<tr>
<td>use_fully_qualified_names</td>
<td>Boolean</td>
<td><em>Optional.</em> Specifies whether requests to this domain require fully-qualified domain names. If set to <strong>true</strong>, all requests to this domain require fully-qualified domain names.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value Format</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>domain must use fully-qualified domain names. It also means that the output from the request displays the fully-qualified name. Restricting requests to fully-qualified user names allows SSSD to differentiate between domains with users with conflicting usernames. If <code>use_fully_qualified_names</code> is set to <code>false</code>, it is possible to use the fully-qualified name in the requests, but only the simplified version is displayed in the output. SSSD can only parse names based on the domain name, not the realm name. The same name can be used for both domains and realms, however.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 8.2.4.2. Configuring an LDAP Domain

An LDAP domain simply means that SSSD uses an LDAP directory as the identity provider (and, optionally, also as an authentication provider). SSSD supports several major directory services:

- Red Hat Directory Server
- OpenLDAP
- Microsoft Active Directory 2008, with Subsystem for UNIX-based Applications

**Note**

DNS service discovery allows the LDAP backend to find the appropriate DNS servers to connect to automatically using a special DNS query.

- [Section 8.2.4.2.1, “Parameters for Configuring an LDAP Domain”](#)
- [Section 8.2.4.2.2, “LDAP Domain Example”](#)
- [Section 8.2.4.2.3, “Active Directory Domain Example”](#)
- [Section 8.2.4.2.4, “Using IP Addresses in Certificate Subject Names”](#)
8.2.4.2.1. Parameters for Configuring an LDAP Domain

An LDAP directory can function as both an identity provider and an authentication provider. The configuration requires enough information to identify and connect to the user directory in the LDAP server, but the way that those connection parameters are defined is flexible.

Other options are available to provide more fine-grained control, like specifying a user account to use to connect to the LDAP server or using different LDAP servers for password operations. The most common options are listed in Table 8.5, "LDAP Domain Configuration Parameters". All of the options listed in Section 8.2.4.1, “General Rules and Options for Configuring a Domain” are also available for LDAP domains.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ldap_uri</strong></td>
<td>Gives a comma-separated list of the URIs of the LDAP servers to which SSSD will connect. The list is given in order of preference, so the first server in the list is tried first. Listing additional servers provides failover protection. This can be detected from the DNS SRV records if it is not given.</td>
</tr>
<tr>
<td><strong>ldap_search_base</strong></td>
<td>Gives the base DN to use for performing LDAP user operations.</td>
</tr>
</tbody>
</table>
| **ldap_tls_reqcert** | Specifies how to check for SSL server certificates in a TLS session. There are four options:  
  - *never* disables requests for certificates.  
  - *allow* requests a certificate, but proceeds normally even if no certificate is given or a bad certificate is given.  
  - *try* requests a certificate and proceeds normally if no certificate is given. If a bad certificate is given, the session terminates.  
  - *demand* and *hard* are the same option. This requires a valid certificate or the session is terminated.  
  The default is *hard*. |
| **ldap_tls_cacert** | Gives the full path and file name to the file that contains the CA certificates for all of the CAs that SSSD recognizes. SSSD will accept any certificate issued by these CAs. This uses the OpenLDAP system defaults if it is not given explicitly. |

Many other options are listed in the man page for LDAP domain configuration, **sssd-ldap(5)**.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ldap_referrals</td>
<td>Sets whether SSSD will use LDAP referrals, meaning forwarding queries from one LDAP database to another. SSSD supports database-level and subtree referrals. For referrals within the same LDAP server, SSSD will adjust the DN of the entry being queried. For referrals that go to different LDAP servers, SSSD does an exact match on the DN. Setting this value to <code>true</code> enables referrals; this is the default.</td>
</tr>
<tr>
<td>ldap_schema</td>
<td>Sets what version of schema to use when searching for user entries. This can be either <code>rfc2307</code> or <code>rfc2307bis</code>. The default is <code>rfc2307</code>.  In RFC 2307, group objects use a multi-valued attribute, <code>memberuid</code>, which lists the names of the users that belong to that group. In RFC 2307bis, group objects use the <code>member</code> attribute, which contains the full distinguished name (DN) of a user or group entry. RFC 2307bis allows nested groups using the <code>member</code> attribute. Because these different schema use different definitions for group membership, using the wrong LDAP schema with SSSD can affect both viewing and managing network resources, even if the appropriate permissions are in place. For example, with RFC 2307bis, all groups are returned when using nested groups or primary/secondary groups. <code>$ id uid=500(myserver) gid=500(myserver) groups=500(myserver),510(myothergroup)</code> If SSSD is using RFC 2307 schema, only the primary group is returned. This setting only affects how SSSD determines the group members. It does not change the actual user data.</td>
</tr>
<tr>
<td>ldap_search_timeout</td>
<td>Sets the time, in seconds, that LDAP searches are allowed to run before they are canceled and cached results are returned. This defaults to five when the <code>enumerate</code> value is false and defaults to 30 when <code>enumerate</code> is true. When an LDAP search times out, SSSD automatically switches to offline mode.</td>
</tr>
<tr>
<td>ldap_network_timeout</td>
<td>Sets the time, in seconds, SSSD attempts to poll an LDAP server after a connection attempt fails. The default is six seconds.</td>
</tr>
</tbody>
</table>
8.2.4.2.2. LDAP Domain Example

The LDAP configuration is very flexible, depending on your specific environment and the SSSD behavior. These are some common examples of an LDAP domain, but the SSSD configuration is not limited to these examples.

**Note**

Along with creating the domain entry, add the new domain to the list of domains for SSSD to query in the `sssd.conf` file. For example:

```
domains = LOCAL, LDAP1, AD, PROXYNIS
```

**Example 8.1. A Basic LDAP Domain Configuration**

An LDAP domain requires three things:

- An LDAP server
- The search base
- A way to establish a secure connection

The last item depends on the LDAP environment. SSSD requires a secure connection since it handles sensitive information. This connection can be a dedicated TLS/SSL connection or it can use Start TLS.

Using a dedicated TLS/SSL connection simply uses an LDAPS connection to connect to the server and is therefore set as part of the `ldap_uri` option:

```
# An LDAP domain
[domain/LDAP]
enumerate = false
cache_credentials = true

id_provider = ldap
auth_provider = ldap

ldap_uri = ldaps://ldap.example.com:636
ldap_search_base = dc=example,dc=com
```

Using Start TLS requires a way to input the certificate information to establish a secure connection dynamically over an insecure port. This is done using the `ldap_id_use_start_tls` option to use Start TLS and then `ldap_tls_cacert` to identify the CA certificate which issued the SSL server certificates.
# An LDAP domain

```bash
[domain/LDAP]
enumerate = false
cache_credentials = true

id_provider = ldap
auth_provider = ldap

ldap_uri = ldap://ldap.example.com
ldap_search_base = dc=example,dc=com
ldap_id_use_start_tls = true
ldap_tls_reqcert = demand
ldap_tls_cacert = /etc/pki/tls/certs/ca-bundle.crt
```

## 8.2.4.2.3. Active Directory Domain Example

For SSSD to work with an Active Directory domain, both the Active Directory domain and the local system have to be configured specially to communicate with one another.

**Note**

The Microsoft Active Directory documentation has complete procedures for configuring the Active Directory domain.

1. **Using authconfig**, set the Linux client to use Active Directory as its LDAP identity provider. For example:

   ```bash
   authconfig --enableldap --enableldapauth --ldapserver=ldap://ad.example.com:389
   --enablekrb5 --krb5realm AD-REALM.EXAMPLE.COM --krb5kdc ad-kdc.example.com:88 --
   krb5adminserver ad-kdc.example.com:749 --update
   ```

   The `authconfig` command is described in *Section 8.1, “Configuring System Authentication”*.

2. Create the Active Directory Domain Services role.

3. Add the Identity Management for UNIX service to the Active Directory Domain Services role. Use the Unix NIS domain as the domain name in the configuration.

4. On the Active Directory server, create a new **Computer** object with the name of the Linux client.
   a. In the **Administrative Tools** menu, select the **Active Directory Users and Computers** application.
   b. Expand the Active Directory root object, such as `ad.example.com`.
   c. Right-click **Computers**, and select the **New** and the **Computer** item.
   d. Enter the name for the Linux client, such as `rhel-server`, and click **OK**.
   e. Expand the **Computers** object.
   f. Right-click the `rhel-server` object, and select **Properties**.
Chapter 8. Configuring Authentication

5. From the command prompt on the Active Directory server, create a machine account, password, and UPN for the Linux host principal.

```cmd
C:\> setspn -A host/rhel-server.example.com@AD-REALM.EXAMPLE.COM rhel-server
Registering ServicePrincipalNames for CN=rhel server,CN=Computers,DC=ad,DC=example,DC=com
   host/rhel server.example.com@AD-REALM.EXAMPLE.COM
Updated object
C:\> setspn -L rhel-server
Registered ServicePrincipalNames for CN=rhel server,CN=Computers,DC=ad,DC=example,DC=com:
   host/rhel server.example.com@AD-REALM.EXAMPLE.COM
C:\> ktpass /princ host/rhel-server.example.com@AD-REALM.EXAMPLE.COM /out rhel-server.keytab /crypto all /ptype KRB5_NT_PRINCIPAL -desonly /mapuser AD\rhel-server$ +rndPass
Targeting domain controller: ad.example.com
Using legacy password setting method
Successfully mapped host/rhel server.redhat.com ...
```

6. Copy the keytab from the Active Directory server to the Linux client, and save it as `/etc/krb5.keytab`.

7. On the Linux system, reset the permissions and owner for the keytab file.

```bash
[root@rhel-server ~]# chown root:root /etc/krb5.keytab
[root@rhel-server ~]# chmod 0600 /etc/krb5.keytab
```

8. Restore the SELinux file permissions for the keytab.

```bash
[root@rhel-server ~]# restorecon /etc/krb5.keytab
```

9. Verify that the host can connect to the Active Directory domain.

```bash
[root@rhel-server ~]# kinit -k -t /etc/krb5.keytab host/rhel-server.example.com@AD-REALM.EXAMPLE.COM
```

10. On the Active Directory server, create a group for the Linux users.

   a. Create a new group named `unixusers`.

   b. Open the `unixusers` group and open the `Unix Attributes` tab.

   c. Configure the Unix settings:

      - The NIS domain
      - The UID

   g. In the **UNIX Attributes**, enter the name of the Linux NIS domain and the IP address of the Linux server.

      Click **OK**.
• The login shell, to `/bin/bash`
• The home directory, to `/home/aduser`
• The primary group name, to `unixusers`

11. Then, configure the SSSD domain on the Linux machine.

```
Example 8.2. An Active Directory 2008 Domain

[root@rhel-server ~]# vim /etc/sssd/sssd.conf

[sssd]
config_file_version = 2
domains = ad.example.com
services = nss, pam

[nss]

[pam]

[domain/ad.example.com]
cache_credentials = true
enumerate = false
id_provider = ldap
auth_provider = krb5
chpass_provider = krb5
access_provider = ldap
ldap_sasl_mech = GSSAPI
ldap_sasl_authid = host/rhel-server.example.com@AD-REALM.EXAMPLE.COM

ldap_schema = rfc2307bis
ldap_user_search_base = ou=user accounts,dc=ad,dc=example,dc=com
ldap_user_object_class = user
ldap_user_principal = userPrincipalName
ldap_user_name = sAMAccountName

ldap_group_search_base = ou=groups,dc=ad,dc=example,dc=com
ldap_group_object_class = group
ldap_access_order = expire
ldap_account_expire_policy = ad
ldap_force_upper_case_realm = true
ldap_disable_referrals = true

#krb5_server = server.ad.example.com
krb5_realm = AD-REALM.EXAMPLE.COM
```

These options are described in the man page for LDAP domain configuration, `sssd-ldap(5)`.

12. Restart SSSD.

```
[root@rhel-server ~]# service sssd restart
```
8.2.4.2.4. Using IP Addresses in Certificate Subject Names

Using an IP address in the `ldap_uri` option instead of the server name may cause the TLS/SSL connection to fail. TLS/SSL certificates contain the server name, not the IP address. However, the `subject alternative name` field in the certificate can be used to include the IP address of the server, which allows a successful secure connection using an IP address.

1. Convert an existing certificate into a certificate request. The signing key (`-signkey`) is the key of the issuer of whatever CA originally issued the certificate. If this is done by an external CA, it requires a separate PEM file; if the certificate is self-signed, then this is the certificate itself. For example:

```
openssl x509 -x509toreq -in old_cert.pem -out req.pem -signkey key.pem
```

With a self-signed certificate:

```
openssl x509 -x509toreq -in old_cert.pem -out req.pem -signkey old_cert.pem
```

2. Edit the `/etc/pki/tls/openssl.cnf` configuration file to include the server’s IP address under the `[ v3_ca ]` section:

```
subjectAltName = IP:10.0.0.10
```

3. Use the generated certificate request to generate a new self-signed certificate with the specified IP address:

```
openssl x509 -req -in req.pem -out new_cert.pem -extfile ./openssl.cnf -extensions v3_ca -signkey old_cert.pem
```

The `-extensions` option sets which extensions to use with the certificate. For this, it should be `v3_ca` to load the appropriate section.

4. Copy the private key block from the `old_cert.pem` file into the `new_cert.pem` file to keep all relevant information in one file.

When creating a certificate through the `certutil` utility provided by the `nss-utils` package, note that `certutil` supports DNS subject alternative names for certificate creation only.

8.2.4.3. Configuring Kerberos Authentication with a Domain

Both LDAP and proxy identity providers can use a separate Kerberos domain to supply authentication. Configuring a Kerberos authentication provider requires the key distribution center (KDC) and the Kerberos domain. All of the principal names must be available in the specified identity provider; if they are not, SSSD constructs the principals using the format `username@REALM`.

**Note**

Kerberos can only provide authentication; it cannot provide an identity database.
SSSD assumes that the Kerberos KDC is also a Kerberos kadmin server. However, production environments commonly have multiple, read-only replicas of the KDC and only a single kadmin server. Use the `krb5_kpasswd` option to specify where the password changing service is running or if it is running on a non-default port. If the `krb5_kpasswd` option is not defined, SSSD tries to use the Kerberos KDC to change the password.

The basic Kerberos configuration options are listed in Table 8.6, "Kerberos Authentication Configuration Parameters". The `sssd-krb5(5)` man page has more information about Kerberos configuration options.

Example 8.3. Basic Kerberos Authentication

```
# A domain with identities provided by LDAP and authentication by Kerberos
[domain/KRBDOMAIN]
enumerate = false
id_provider = ldap
chpass_provider = krb5
ldap_uri = ldap://ldap.example.com
ldap_search_base = dc=example,dc=com
ldap-tls_reqcert = demand
ldap_tls_cacert = /etc/pki/tls/certs/ca-bundle.crt

auth_provider = krb5
krb5_server = 192.168.1.1, kerberos.example.com
krb5_realm = EXAMPLE.COM
krb5_kpasswd = kerberos.admin.example.com
krb5_auth_timeout = 15
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chpass_provider</td>
<td>Specifies which service to use for password change operations. This is assumed to be the same as the authentication provider. To use Kerberos, set this to <code>krb5</code>.</td>
</tr>
<tr>
<td>krb5_server</td>
<td>Gives a comma-separated list of IP addresses or hostnames of Kerberos servers to which SSSD will connect. The list is given in order of preference, so the first server in the list is tried first. Listing additional servers provides failover protection. When using service discovery for KDC or kpasswd servers, SSSD first searches for DNS entries that specify UDP as the connection protocol, and then falls back to TCP.</td>
</tr>
<tr>
<td>krb5_realm</td>
<td>Identifies the Kerberos realm served by the KDC.</td>
</tr>
<tr>
<td>krb5_lifetime</td>
<td>Requests a Kerberos ticket with the specified lifetime in seconds (s), minutes (m), hours (h) or days (d).</td>
</tr>
<tr>
<td>krb5_renewable_lifetime</td>
<td>Requests a renewable Kerberos ticket with a total lifetime that is specified in seconds (s), minutes (m), hours (h) or days (d).</td>
</tr>
<tr>
<td>krb5_renew_interval</td>
<td>Sets the time, in seconds, for SSSD to check if tickets should be renewed. Tickets are renewed</td>
</tr>
</tbody>
</table>
### Chapter 8. Configuring Authentication

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>automatically once they exceed half their lifetime. If this option is missing or set to zero, then automatic ticket renewal is disabled.</td>
<td></td>
</tr>
<tr>
<td>krb5_store_password_if_offline</td>
<td>Sets whether to store user passwords if the Kerberos authentication provider is offline, and then to use that cache to request tickets when the provider is back online. The default is <code>false</code>, which does not store passwords.</td>
</tr>
<tr>
<td>krb5_kpasswd</td>
<td>Lists alternate Kerberos kadmin servers to use if the change password service is not running on the KDC.</td>
</tr>
</tbody>
</table>
| krb5_ccname_template             | Gives the directory to use to store the user’s credential cache. This can be templatized, and the following tokens are supported:  
  • `%u`, the user's login name  
  • `%U`, the user's login UID  
  • `%p`, the user's principal name  
  • `%r`, the realm name  
  • `%h`, the user's home directory  
  • `%d`, the value of the `krb5ccache_dir` parameter  
  • `%P`, the process ID of the SSSD client.  
  • `%%%`, a literal percent sign (%)  
  • `XXXXXX`, a string at the end of the template which instructs SSSD to create a unique filename safely  
  For example:  
  
  ```bash  
  krb5_ccname_template = FILE:%d/krb5cc_%u_XXXXXX  
  ``` |
| krb5_ccachedir                   | Specifies the directory to store credential caches. This can be templatized, using the same tokens as `krb5_ccname_template`, except for `%d` and `%P`. If `%u`, `%U`, `%p`, or `%h` are used, then SSSD creates a private directory for each user; otherwise, it creates a public directory. |
| krb5_auth_timeout                | Gives the time, in seconds, before an online authentication or change password request is aborted. If possible, the authentication request is continued offline. The default is 15 seconds. |
8.2.4.4. Configuring a Proxy Domain

A proxy with SSSD is just a relay, an intermediary configuration. SSSD connects to its proxy service, and then that proxy loads the specified libraries. This allows SSSD to use some resources that it otherwise would not be able to use. For example, SSSD only supports LDAP and Kerberos as authentication providers, but using a proxy allows SSSD to use alternative authentication methods like a fingerprint scanner or smart card.

Table 8.7. Proxy Domain Configuration Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>proxy_pam_target</td>
<td>Specifies the target to which PAM must proxy as an authentication provider. The PAM target is a file containing PAM stack information in the default PAM directory, /etc/pam.d/. This is used to proxy an authentication provider.</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong> Ensure that the proxy PAM stack does not recursively include pam_sss.so.</td>
</tr>
<tr>
<td>proxy_lib_name</td>
<td>Specifies which existing NSS library to proxy identity requests through. This is used to proxy an identity provider.</td>
</tr>
</tbody>
</table>

Example 8.4. Proxy Identity and Kerberos Authentication

The proxy library is loaded using the **proxy_lib_name** parameter. This library can be anything as long as it is compatible with the given authentication service. For a Kerberos authentication provider, it must be a Kerberos-compatible library, like NIS.

```
[domain/PROXY_KRB5]
auth_provider = krb5
krb5_server = 192.168.1.1
krb5_realmd = EXAMPLE.COM

id_provider = proxy
proxy_lib_name = nis
enumerate = true
cache_credentials = true
```

Example 8.5. LDAP Identity and Proxy Authentication

The proxy library is loaded using the **proxy_pam_target** parameter. This library must be a PAM module that is compatible with the given identity provider. For example, this uses a PAM fingerprint module with LDAP:

```
[domain/LDAP_PROXY]
id_provider = ldap
ldap_uri = ldap://example.com
```
ldap_search_base = dc=example,dc=com

auth_provider = proxy
proxy_pam_target = sssdpamproxy
enumerate = true
cache_credentials = true

After the SSSD domain is configured, make sure that the specified PAM files are configured. In this example, the target is sssdpamproxy, so create a /etc/pam.d/sssdpamproxy file and load the PAM/LDAP modules:

<table>
<thead>
<tr>
<th>Service</th>
<th>Requirement</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>auth</td>
<td>required</td>
<td>pam_frprint.so</td>
</tr>
<tr>
<td>account</td>
<td>required</td>
<td>pam_frprint.so</td>
</tr>
<tr>
<td>password</td>
<td>required</td>
<td>pam_frprint.so</td>
</tr>
<tr>
<td>session</td>
<td>required</td>
<td>pam_frprint.so</td>
</tr>
</tbody>
</table>

Example 8.6. Proxy Identity and Authentication
SSSD can have a domain with both identity and authentication proxies. The only configuration given then are the proxy settings, proxy_pam_target for the authentication PAM module and proxy_lib_name for the service, like NIS or LDAP.

This example illustrates a possible configuration, but this is not a realistic configuration. If LDAP is used for identity and authentication, then both the identity and authentication providers should be set to the LDAP configuration, not a proxy.

[domain/PROXY_PROXY]
auth_provider = proxy
id_provider = proxy
proxy_lib_name = ldap
proxy_pam_target = sssdproxyldap
enumerate = true
cache_credentials = true

Once the SSSD domain is added, then update the system settings to configure the proxy service:

1. Create a /etc/pam.d/sssldproxyldap file which requires the pam_ldap.so module:

<table>
<thead>
<tr>
<th>Service</th>
<th>Requirement</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>auth</td>
<td>required</td>
<td>pam_ldap.so</td>
</tr>
<tr>
<td>account</td>
<td>required</td>
<td>pam_ldap.so</td>
</tr>
<tr>
<td>password</td>
<td>required</td>
<td>pam_ldap.so</td>
</tr>
<tr>
<td>session</td>
<td>required</td>
<td>pam_ldap.so</td>
</tr>
</tbody>
</table>

2. Make sure the nss-pam-ldap package is installed.

   [root@server ~]# yum install nss-pam-ldap

3. Edit the /etc/nslcd.conf file, the configuration file for the LDAP name service daemon, to contain the information for the LDAP directory:

   uid nslcd
gid ldap
uri ldaps://ldap.example.com:636
base dc=example,dc=com
8.2.5. Configuring Access Control for SSSD Domains

SSSD provides a rudimentary access control for domain configuration, allowing either simple user allow/deny lists or using the LDAP backend itself.

8.2.5.1. Using the Simple Access Provider

The Simple Access Provider allows or denies access based on a list of usernames or groups.

The Simple Access Provider is a way to restrict access to certain, specific machines. For example, if a company uses laptops, the Simple Access Provider can be used to restrict access to only a specific user or a specific group, even if a different user authenticated successfully against the same authentication provider.

The most common options are `simple_allow_users` and `simple_allow_groups`, which grant access explicitly to specific users (either the given users or group members) and deny access to everyone else. It is also possible to create deny lists (which deny access only to explicit people and implicitly allow everyone else access).

The Simple Access Provider adheres to the following four rules to determine which users should or should not be granted access:

- If both the allow and deny lists are empty, access is granted.
- If any list is provided, allow rules are evaluated first, and then deny rules. Practically, this means that deny rules supersede allow rules.
- If an allowed list is provided, then all users are denied access unless they are in the list.
- If only deny lists are provided, then all users are allowed access unless they are in the list.

This example grants access to two users and anyone who belongs to the IT group; implicitly, all other users are denied:

```
[domain/example.com]
access_provider = simple
simple_allow_users = jsmith,bjensen
simple_allow_groups = itgroup
```

Note

The LOCAL domain in SSSD does not support `simple` as an access provider.

Other options are listed in the `sssd-simple` man page, but these are rarely used.
8.2.5.2. Using the LDAP Access Filter

The LDAP server itself can provide the access control rules. The associated filter option `ldap_access_filter` specifies which users are granted access to the specified host. The user filter must be used or all users are denied access.

For example:

```
[domain/example.com]
access_provider = ldap
ldap_access_filter = memberOf=cn=allowedusers,ou=Groups,dc=example,dc=com
```

**Note**

Offline caching for LDAP access providers is limited to determining whether the user's last online login attempt was successful. Users that were granted access during their last login will continue to be granted access while offline.

SSSD can also check results by the account expiration policy and the `authorizedService` attribute.

8.2.6. Configuring Domain Failover

SSSD attempts to connect to machines and to services separately.

When SSSD tries to connect to one of its domain backends, it first tries to resolve the hostname of a given machine. If this resolution attempt fails, the machine is considered offline, and SSSD no longer attempts to connect to this machine for any other service.

If the resolution attempt succeeds, the backend tries to connect to a service on this machine. If the service connection attempt fails, then only this particular service is considered offline and the backend automatically switches over to the next service. The machine is still considered online and might still be tried for another service.

SSSD only tries the first IP address given in the DNS A record. To find multiple servers with a single request, SSSD relies on SRV records.

Connections are retried to offline machines or services every 30 seconds, until SSSD can successfully connect to the backend.

8.2.6.1. Configuring Failover

Configuring failover allows SSSD to switch automatically to a different server if the primary server fails. These servers are entered as a case-insensitive, comma-separated list in the `[domain/Name]` sections of the `/etc/sssd/sssd.conf` file. The servers are listed in order of preference. This list can contain any number of servers.

For example, for a native LDAP domain:

```
ldap_uri = ldap://ldap0.example.com, ldap://ldap1.example.com, ldap://ldap2.example.com
```

The first entry, `ldap://ldap0.example.com`, is the primary server. If this server fails, SSSD first attempts to connect to `ldap1.example.com` and then `ldap2.example.com`. 
If the server parameter is not specified, then SSSD uses service discovery to try to find another server on the network.

**Important**

The failover servers must be entered as a comma-separated list of values for a single key. If there are multiple keys, SSSD only recognizes the last entry.

### 8.2.6.2. Using SRV Records with Failover

SSSD supports SRV records in its failover configuration. The SSSD configuration can specify a server that is later resolved into a list of specific servers using SRV requests.

For every service with which to use service discovery, add a special DNS record to the DNS server:

```
_service._protocol._domain TTL priority weight port hostname
```

The `priority` and `weight` attributes of SRV records provide fine-grained control over which servers to contact first if the primary server fails.

A typical configuration contains multiple such records, each with a different priority for failover and different weights for load balancing.

For more information on SRV records, see [RFC 2782](http://tools.ietf.org/html/rfc2782).

### 8.2.7. Managing the SSSD Cache

SSSD can define multiple domains of the same type and different types of domain. SSSD maintains a separate database file for each domain, meaning each domain has its own cache. These cache files are stored in the `/var/lib/sss/db/` directory.

#### 8.2.7.1. Purging the SSSD Cache

As LDAP updates are made to the identity provider for the domains, it can be necessary to clear the cache to reload the new information quickly.

The cache purge utility, `sss_cache`, invalidates records in the SSSD cache for a user, a domain, or a group. Invalidating the current records forces the cache to retrieve the updated records from the identity provider, so changes can be realized quickly.

Most commonly, this is used to clear the cache and update the records for an entire domain:

**Example 8.7. Purging Domain Records**

```
[root@server ~]# sss_cache -d LDAP1
```

If the administrator knows that a specific record (user, group, or netgroup) has been updated, then `sss_cache` can purge the records for that specific account, and leave the rest of the cache intact.

---

1. [RFC 2782](http://tools.ietf.org/html/rfc2782)
Example 8.8. Purging a User Record

```
[root@server ~]# sss_cache -u jsmith
```

Table 8.8. sss_cache Options

<table>
<thead>
<tr>
<th>Short Argument</th>
<th>Long Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-d name</td>
<td>--domain name</td>
<td>Invalidates cache entries for users, groups, and other entries only within the specified domain.</td>
</tr>
<tr>
<td>-G</td>
<td>--groups</td>
<td>Invalidates all group records. If -g is also used, -G takes precedence and -g is ignored.</td>
</tr>
<tr>
<td>-g name</td>
<td>--group name</td>
<td>Invalidates the cache entry for the specified group.</td>
</tr>
<tr>
<td>-N</td>
<td>--netgroups</td>
<td>Invalidates cache entries for all netgroup cache records. If -n is also used, -N takes precedence and -n is ignored.</td>
</tr>
<tr>
<td>-n name</td>
<td>--netgroup name</td>
<td>Invalidates the cache entry for the specified netgroup.</td>
</tr>
<tr>
<td>-U</td>
<td>--users</td>
<td>Invalidates cache entries for all user records. If the -u option is also used, -U takes precedence and -u is ignored.</td>
</tr>
<tr>
<td>-u name</td>
<td>--user name</td>
<td>Invalidates the cache entry for the specified user.</td>
</tr>
</tbody>
</table>

8.2.7.2. Deleting Domain Cache Files

All cache files are named for the domain. For example, for a domain named exampleldap, the cache file is named cache_exampleldap.ldb.

Be careful when you delete a cache file. This operation has significant effects:

- Deleting the cache file deletes all user data, both identification and cached credentials. Consequently, do not delete a cache file unless the system is online and can authenticate with a username against the domain's servers. Without a credentials cache, offline authentication will fail.

- If the configuration is changed to reference a different identity provider, SSSD will recognize users from both providers until the cached entries from the original provider time out.

It is possible to avoid this by purging the cache, but the better option is to use a different domain name for the new provider. When SSSD is restarted, it creates a new cache file with the new name and the old file is ignored.
8.2.8. Configuring OpenSSH to Check SSSD for Cached Keys (TECH PREVIEW)

OpenSSH creates secure, encrypted connections between two systems. One machine authenticates to another machine to allow access; the authentication can be of the machine itself for server connections or of a user on that machine. OpenSSH is described in more detail in Chapter 9, OpenSSH.

This authentication is performed through public-private key pairs that identify the authenticating user or machine. The remote machine or user attempting to access the machine presents a key pair. The local machine then elects whether to trust that remote entity; if it is trusted, the public key for that remote machine is stored in the known_hosts file or for the remote user in authorized_keys. Whenever that remote machine or user attempts to authenticate again, the local system simply checks the known_hosts or authorized_keys file first to see if that remote entity is recognized and trusted. If it is, then access is granted.

The first problem comes in verifying those identities reliably.

The known_hosts file is a triplet of the machine name, its IP address, and its public key:

```
server.example.com,255.255.255.255 ssh-rsa AbcdEfg1234ZYX098776/AbcdEfg1234ZYX098776/AbcdEfg1234ZYX098776=
```

The known_hosts file can quickly become outdated for a number of different reasons: systems using DHCP cycle through IP addresses, new keys can be re-issued periodically, or virtual machines or services can be brought online and removed. This changes the hostname, IP address, and key triplet.

Administrators have to clean and maintain a current known_hosts file to maintain security. (Or system users get in the habit of simply accepting any machine and key presented, which negates the security benefits of key-based security.)

Additionally, problem for both machines and users is distributing keys in a scalable way. Machines can send their keys are part of establishing an encrypted session, but users have to supply their keys in advance. Simply propagating and then updating keys consistently is a difficult administrative task.

Lastly, SSH key and machine information are only maintained locally. There may be machines or users on the network which are recognized and trusted by some systems and not by others because the known_hosts file has not been updated uniformly.

The goal of SSSD is to server as a credentials cache. This includes working as a credentials cache for SSH public keys for machines and users. OpenSSH is configured to reference SSSD to check for cached keys; SSSD uses Red Hat Linux’s Identity Management (IPA) domain as an identity, and IPA actually stores the public keys and host information.

**NOTE**

Only Linux machines enrolled, or joined, in the IPA domain can use SSSD as a key cache for OpenSSH. Other Unix machines and Windows machines must use the regular authentication mechanisms with the known_hosts file.

8.2.8.1. Configuring OpenSSH to Use SSSD for Host Keys

OpenSSH is configured in either a user-specific configuration file (~/.ssh/config) or a system-wide configuration file (/etc/ssh/ssh_config). The user file has precedence over the system settings.
and the first obtained value for a parameter is used. The formatting and conventions for this file are covered in Chapter 9, OpenSSH.

In order to manage host keys, SSSD has a tool, `sss_ssh_knownhostsproxy`, which performs three operations:

1. Retrieves the public host key from the enrolled Linux system.
2. Stores the host key in a custom hosts file, `.ssh/sss_known_hosts`.
3. Establishes a connection with the host machine, either a socket (the default) or a secure connection.

This tool has the format:

```
sss_ssh_knownhostsproxy [-d sssd_domain] [-p ssh_port] HOST [PROXY_COMMAND]
```

<table>
<thead>
<tr>
<th>Short Argument</th>
<th>Long Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOSTNAME</td>
<td></td>
<td>Gives the hostname of the host to check and connect to. In the OpenSSH configuration file, this can be a token, <code>%h</code>.</td>
</tr>
<tr>
<td>PROXY_COMMAND</td>
<td></td>
<td>Passes a proxy command to use to connect to the SSH client. This is similar to running <code>ssh -o ProxyCommand=value</code>. This option is used when running <code>sss_ssh_knownhostsproxy</code> from the command line or through another script, but is not necessary in the OpenSSH configuration file.</td>
</tr>
<tr>
<td>-d sssd_domain</td>
<td>--domain sssd_domain</td>
<td>Only searches for public keys in entries in the specified domain. If not given, SSSD searches for keys in all configured domains.</td>
</tr>
<tr>
<td>-p port</td>
<td>--port port</td>
<td>Uses this port to connect to the SSH client. By default, this is port 22.</td>
</tr>
</tbody>
</table>

To use this SSSD tool, add or edit two parameters to the `ssh_config` or `~/.ssh/config` file:

- Specify the command to use to connect to the SSH client (ProxyCommand). This is the `sss_ssh_knownhostsproxy`, with the desired arguments and hostname.

- Specify the location of the SSSD hosts file, rather than the default `known_hosts` file (UserKnownHostsFile). The SSSD hosts file is `.ssh/sss_known_hosts`.

For example, this looks for public keys in the IPA1 SSSD domain and connects over whatever port and host are supplied:

```
ProxyCommand /usr/bin/sss_ssh_knownhostsproxy -p %p -d IPA1 %h
```
8.2.8.2. Configuring OpenSSH to Use SSSD for User Keys

User keys are stored on a local system in the `authorized_keys` file for OpenSSH. As with hosts, SSSD can maintain and automatically update a separate cache of user public keys for OpenSSH to refer to. This is kept in the `.ssh/sss_authorised_keys` file.

OpenSSH is configured in either a user-specific configuration file (`~/.ssh/config`) or a system-wide configuration file (`/etc/ssh/ssh_config`). The user file has precedence over the system settings and the first obtained value for a parameter is used. The formatting and conventions for this file are covered in Chapter 9, OpenSSH.

In order to manage user keys, SSSD has a tool, `sss_ssh_authorisedkeys`, which performs two operations:

1. Retrieves the user's public key from the user entries in the Identity Management (IPA) domain.
2. Stores the user key in a custom file, `.ssh/sss_authorised_keys`, in the standard authorized keys format.

This tool has the format:

```
sss_ssh_authorisedkeys [-d sssd_domain] USER
```

<table>
<thead>
<tr>
<th>Short Argument</th>
<th>Long Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER</td>
<td></td>
<td>Gives the username or account name for which to obtain the public key. In the OpenSSH configuration file, this can be represented by a token, <code>%u</code>.</td>
</tr>
<tr>
<td>-d sssd_domain</td>
<td>--domain sssd_domain</td>
<td>Only searches for public keys in entries in the specified domain. If not given, SSSD searches for keys in all configured domains.</td>
</tr>
</tbody>
</table>

There are two possible options for how to configure OpenSSH to use SSSD for user keys, depending on the SSH deployment:

- Most commonly, SSH supports the authorized key command. In that case, it is necessary only to specify the command to run to retrieve user keys. For example:

  ```
  AuthorizedKeysCommand /usr/bin/sss_ssh_authorisedkeys
  ```

- SSH can also support a public key agent. In that case, give the command to use to retrieve agent keys, including tokens for required arguments (such as the username):

  ```
  PubKeyAgent /usr/bin/sss_ssh_authorisedkeys %u
  ```

8.2.9. Using NSCD with SSSD

---

UserKnownHostsFile2 .ssh/sss_known_hosts
SSSD is not designed to be used with the NSCD daemon. Even though SSSD does not directly conflict with NSCD, using both services can result in unexpected behavior, especially with how long entries are cached.

The most common evidence of a problem is conflicts with NFS. When using Network Manager to manage network connections, it may take several minutes for the network interface to come up. During this time, various services attempt to start. If these services start before the network is up and the DNS servers are available, these services fail to identify the forward or reverse DNS entries they need. These services will read an incorrect or possibly empty `resolv.conf` file. This file is typically only read once, and so any changes made to this file are not automatically applied. This can cause NFS locking to fail on the machine where the NSCD service is running, unless that service is manually restarted.

To avoid this problem, enable caching for hosts and services in the `/etc/nscd.conf` file and rely on the SSSD cache for the `passwd`, `group`, and `netgroup` entries.

Change the `/etc/nscd.conf` file:

```bash
enable-cache hosts yes
enable-cache passwd no
enable-cache group no
enable-cache netgroup no
```

With NSCD answering hosts requests, these entries will be cached by NSCD and returned by NSCD during the boot process. All other entries are handled by SSSD.

### 8.2.10. Troubleshooting SSSD

#### 8.2.10.1. Setting Debug Logs for SSSD Domains

Each domain sets its own debug log level. Increasing the log level can provide more information about problems with SSSD or with the domain configuration.

To change the log level, set the `debug_level` parameter for each section in the `sssd.conf` file for which to produce extra logs. For example:

```bash
[domain/LDAP]
enumerate = false
cache_credentials = true
default_log_level = 9
```

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Fatal failures. Anything that would prevent SSSD from starting up or causes it to cease running.</td>
</tr>
<tr>
<td>1</td>
<td>Critical failures. An error that doesn't kill the SSSD, but one that indicates that at least one major feature is not going to work properly.</td>
</tr>
<tr>
<td>2</td>
<td>Serious failures. An error announcing that a particular request or operation has failed.</td>
</tr>
<tr>
<td>3</td>
<td>Minor failures. These are the errors that would percolate down to cause the operation failure of 2.</td>
</tr>
<tr>
<td>4</td>
<td>Configuration settings.</td>
</tr>
</tbody>
</table>
### Level Description

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Function data.</td>
</tr>
<tr>
<td>6</td>
<td>Trace messages for operation functions.</td>
</tr>
<tr>
<td>7</td>
<td>Trace messages for internal control functions.</td>
</tr>
<tr>
<td>8</td>
<td>Contents of function-internal variables that may be interesting.</td>
</tr>
<tr>
<td>9</td>
<td>Extremely low-level tracing information.</td>
</tr>
</tbody>
</table>

**NOTE**

In versions of SSSD older than 1.8, debug log levels could be set globally in the `[sssd]` section. Now, each domain and service must configure its own debug log level.

To copy the global SSSD debug log levels into each configuration area in the SSSD configuration file, use the `sssd_update_debug_levels.py` script.

```python
python /usr/lib/python2.6/site-packages/sssd_update_debug_levels.py
```

#### 8.2.10.2. Checking SSSD Log Files

SSSD uses a number of log files to report information about its operation, located in the `/var/log/` `sssd/` directory. SSSD produces a log file for each domain, as well as an `sssd_pam.log` and an `sssd_nss.log` file.

Additionally, the `/var/log/secure` file logs authentication failures and the reason for the failure.

#### 8.2.10.3. Problems with SSSD Configuration

**SSSD fails to start**

SSSD requires that the configuration file be properly set up, with all the required entries, before the daemon will start.

- SSSD requires at least one properly configured domain before the service will start. Without a domain, attempting to start SSSD returns an error that no domains are configured:

```bash
# sssd -d4
[sssd] [ldb] (3): server_sort:Unable to register control with rootdse!
[sssd] [confdb_get_domains] (0): No domains configured, fatal error!
[sssd] [get_monitor_config] (0): No domains configured.
```

Edit the `/etc/sssd/sssd.conf` file and create at least one domain.

- SSSD also requires at least one available service provider before it will start. If the problem is with the service provider configuration, the error message indicates that there are no services configured:

```bash
[sssd] [get_monitor_config] (0): No services configured!
```
Edit the `/etc/sssd/sssd.conf` file and configure at least one service provider.

**Important**

SSSD requires that service providers be configured as a comma-separated list in a single `services` entry in the `/etc/sssd/sssd.conf` file. If services are listed in multiple entries, only the last entry is recognized by SSSD.

I don't see any groups with 'id' or group members with 'getent group'.

This may be due to an incorrect `ldap_schema` setting in the `[domain/DOMAINNAME]` section of `sssd.conf`.

SSSD supports RFC 2307 and RFC 2307bis schema types. By default, SSSD uses the more common RFC 2307 schema.

The difference between RFC 2307 and RFC 2307bis is the way which group membership is stored in the LDAP server. In an RFC 2307 server, group members are stored as the multi-valued `memberuid` attribute, which contains the names of the users that are members. In an RFC2307bis server, group members are stored as the multi-valued `member` or `uniqueMember` attribute which contains the DN of the user or group that is a member of this group. RFC2307bis allows nested groups to be maintained as well.

If group lookups are not returning any information:

1. Set `ldap_schema` to `rfc2307bis`.
2. Delete `/var/lib/sss/db/cache_DOMAINNAME.ldb`.
3. Restarting SSSD.

If that doesn't work, add this line to `sssd.conf`:

```
ldap_group_name = uniqueMember
```

Then delete the cache and restart SSSD again.

**Authentication fails against LDAP.**

To perform authentication, SSSD requires that the communication channel be encrypted. This means that if `sssd.conf` is configured to connect over a standard protocol (ldap://), it attempts to encrypt the communication channel with Start TLS. If `sssd.conf` is configured to connect over a secure protocol (ldaps://), then SSSD uses SSL.

This means that the LDAP server must be configured to run in SSL or TLS. TLS must be enabled for the standard LDAP port (389) or SSL enabled on the secure LDAPS port (636). With either SSL or TLS, the LDAP server must also be configured with a valid certificate trust.

An invalid certificate trust is one of the most common issues with authenticating against LDAP. If the client does not have proper trust of the LDAP server certificate, it is unable to validate the connection, and SSSD refuses to send the password. The LDAP protocol requires that the password be sent in
plaintext to the LDAP server. Sending the password in plaintext over an unencrypted connection is a security problem.

If the certificate is not trusted, a syslog message is written, indicating that TLS encryption could not be started. The certificate configuration can be tested by checking if the LDAP server is accessible apart from SSSD. For example, this tests an anonymous bind over a TLS connection to test.example.com:

```bash
$ ldapsearch -x -ZZ -h test.example.com -b dc=example,dc=com
```

If the certificate trust is not properly configured, the test fails with this error:

```
ldap_start_tls: Connect error (-11) additional info: TLS error -8179:Unknown code ___f 13
```

To trust the certificate:

1. Obtain a copy of the public CA certificate for the certificate authority used to sign the LDAP server certificate and save it to the local system.
2. Add a line to the `sssd.conf` file that points to the CA certificate on the filesystem.

   ```
   ldap_tls_cacert = /path/to/cacert
   ```
3. If the LDAP server uses a self-signed certificate, remove the `ldap_tls_reqcert` line from the `sssd.conf` file.

   This parameter directs SSSD to trust any certificate issued by the CA certificate, which is a security risk with a self-signed CA certificate.

**Connecting to LDAP servers on non-standard ports fail.**

When running SELinux in enforcing mode, the client’s SELinux policy has to be modified to connect to the LDAP server over the non-standard port. For example:

```bash
# semanage port -a -t ldap_port_t -p tcp 1389
```

**NSS fails to return user information**

This usually means that SSSD cannot connect to the NSS service.

- Ensure that NSS is running:

  ```
  # service sssd status
  ```
- If NSS is running, make sure that the provider is properly configured in the `[nss]` section of the `/etc/sssd/sssd.conf` file. Especially check the `filter_users` and `filter_groups` attributes.
- Make sure that NSS is included in the list of services that SSSD uses.
- Check the configuration in the `/etc/nsswitch.conf` file.

**NSS returns incorrect user information**

If searches are returning the incorrect user information, check that there are not conflicting usernames in separate domains. When there are multiple domains, set the `use_fully_qualified_domains`
attribute to **true** in the `/etc/sssd/sssd.conf` file. This differentiates between different users in different domains with the same name.

**Setting the password for the local SSSD user prompts twice for the password**

When attempting to change a local SSSD user's password, it may prompt for the password twice:

```
[root@clientF11 tmp]# passwd user1000
Changing password for user user1000.
New password:
Retype new password:
New Password:
Reenter new Password:
pwd: all authentication tokens updated successfully.
```

This is the result of an incorrect PAM configuration. Ensure that the `use_authtok` option is correctly configured in your `/etc/pam.d/system-auth` file.
OpenSSH

SSH (Secure Shell) 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.

The ssh program 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.

Fedora includes the general OpenSSH package (openssh) as well as the OpenSSH server (openssh-server) and client (openssh-clients) packages. Note that the OpenSSH packages require the OpenSSL package (openssl), which installs several important cryptographic libraries, enabling OpenSSH to provide encrypted communications.

9.1. The SSH Protocol

9.1.1. Why Use SSH?

Potential intruders 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

The attacker can be somewhere on the network between the communicating parties, copying any information passed between them. He may intercept and keep the information, or alter the information and send it on to the intended recipient.

This attack is usually performed using a packet sniffer, a rather common network utility that captures each packet flowing through the network, and analyzes its content.

Impersonation of a particular host

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 performed using a technique known as DNS poisoning, or via so-called IP spoofing. In the first case, the intruder uses a cracked DNS server to point client systems to a maliciously duplicated host. In the second case, the intruder sends falsified network packets that appear to be from a trusted host.

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.
9.1.2. Main Features

The SSH protocol provides the following safeguards:

No one can pose as the intended server
   After an initial connection, the client can verify that it is connecting to the same server it had
   connected to previously.

No one can capture the authentication information
   The client transmits its authentication information to the server using strong, 128-bit encryption.

No one can intercept the communication
   All data sent and received during a session is transferred using 128-bit encryption, making
   intercepted transmissions extremely difficult to decrypt and read.

Additionally, it also offers the following options:

It provides secure means to use graphical applications over a network
   Using a technique called X11 forwarding, the client can forward X11 (X Window System)
   applications from the server.

It provides a way to secure otherwise insecure protocols
   The SSH protocol encrypts everything it sends and receives. 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.

It can be used to create a secure channel
   The OpenSSH server and client can be configured to create a tunnel similar to a virtual private
   network for traffic between server and client machines.

It supports the Kerberos authentication
   OpenSSH servers and clients can be configured to authenticate using the GSSAPI (Generic
   Security Services Application Program Interface) implementation of the Kerberos network
   authentication protocol.

9.1.3. Protocol Versions

Two varieties of SSH currently exist: version 1, and newer version 2. The OpenSSH suite under
Fedora uses SSH version 2, which has an enhanced key exchange algorithm not vulnerable to the
known exploit in version 1. However, for compatibility reasons, the OpenSSH suite does support
version 1 connections as well.

Avoid using SSH version 1

To ensure maximum security for your connection, it is recommended that only SSH version 2-
compatible servers and clients are used whenever possible.

9.1.4. 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.

### 9.1.4.1. 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.

Always verify the integrity of a new SSH server

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.

9.1.4.2. 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.

9.1.4.3. Channels

After a successful authentication over the SSH transport layer, multiple channels are opened via a technique called multiplexing. 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.

9.2. An OpenSSH Configuration

In order to perform tasks described in this section, you must have superuser privileges. To obtain them, log in as root by typing:

```
su -
```

9.2.1. Configuration Files

---

1 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.
There are two different sets of configuration files: those for client programs (that is, ssh, scp, and sftp), and those for the server (the sshd daemon).

System-wide SSH configuration information is stored in the /etc/ssh/ directory. See Table 9.1, “System-wide configuration files” for a description of its content.

Table 9.1. System-wide configuration files

<table>
<thead>
<tr>
<th>Configuration File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc/ssh/moduli</td>
<td>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.</td>
</tr>
<tr>
<td>/etc/ssh/ssh_config</td>
<td>The default SSH client configuration file. Note that it is overridden by ~/.ssh/config if it exists.</td>
</tr>
<tr>
<td>/etc/ssh/sshd_config</td>
<td>The configuration file for the sshd daemon.</td>
</tr>
<tr>
<td>/etc/ssh/ssh_host_dsa_key</td>
<td>The DSA private key used by the sshd daemon.</td>
</tr>
<tr>
<td>/etc/ssh/ssh_host_dsa_key.pub</td>
<td>The DSA public key used by the sshd daemon.</td>
</tr>
<tr>
<td>/etc/ssh/ssh_host_key</td>
<td>The RSA private key used by the sshd daemon for version 1 of the SSH protocol.</td>
</tr>
<tr>
<td>/etc/ssh/ssh_host_key.pub</td>
<td>The RSA public key used by the sshd daemon for version 1 of the SSH protocol.</td>
</tr>
<tr>
<td>~/.ssh/id_dsa</td>
<td>Contains the DSA private key of the user.</td>
</tr>
<tr>
<td>~/.ssh/id_dsa.pub</td>
<td>The DSA public key of the user.</td>
</tr>
<tr>
<td>~/.ssh/id_rsa</td>
<td>The RSA private key used by ssh for version 2 of the SSH protocol.</td>
</tr>
<tr>
<td>~/.ssh/id_rsa.pub</td>
<td>The RSA public key used by ssh for version 2 of the SSH protocol.</td>
</tr>
<tr>
<td>~/.ssh/identity</td>
<td>The RSA private key used by ssh for version 1 of the SSH protocol.</td>
</tr>
<tr>
<td>~/.ssh/identity.pub</td>
<td>The RSA public key used by ssh for version 1 of the SSH protocol.</td>
</tr>
</tbody>
</table>

User-specific SSH configuration information is stored in the user's home directory within the ~/.ssh/ directory. See Table 9.2, “User-specific configuration files” for a description of its content.

Table 9.2. User-specific configuration files

<table>
<thead>
<tr>
<th>Configuration File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>~/.ssh/authorized_keys</td>
<td>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.</td>
</tr>
<tr>
<td>~/.ssh/id_dsa</td>
<td>Contains the DSA private key of the user.</td>
</tr>
<tr>
<td>~/.ssh/id_dsa.pub</td>
<td>The DSA public key of the user.</td>
</tr>
<tr>
<td>~/.ssh/id_rsa</td>
<td>The RSA private key used by ssh for version 2 of the SSH protocol.</td>
</tr>
<tr>
<td>~/.ssh/id_rsa.pub</td>
<td>The RSA public key used by ssh for version 2 of the SSH protocol.</td>
</tr>
<tr>
<td>~/.ssh/identity</td>
<td>The RSA private key used by ssh for version 1 of the SSH protocol.</td>
</tr>
<tr>
<td>~/.ssh/identity.pub</td>
<td>The RSA public key used by ssh for version 1 of the SSH protocol.</td>
</tr>
</tbody>
</table>
## Chapter 9. OpenSSH

### 9.2.2. Starting an OpenSSH Server

<table>
<thead>
<tr>
<th>Configuration File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>~/.ssh/known_hosts</td>
<td>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.</td>
</tr>
</tbody>
</table>

Refer to the `ssh_config` and `sshd_config` man pages for information concerning the various directives available in the SSH configuration files.

To run an OpenSSH server, you must have the `openssh-server` and `openssh` packages installed. Refer to Section 5.2.4, "Installing Packages" for more information on how to install new packages in Fedora.

To start the `sshd` daemon, type the following at a shell prompt:

```
systemctl start sshd.service
```

To stop the running `sshd` daemon, use the following command:

```
systemctl stop sshd.service
```

If you want the daemon to start automatically at the boot time, type:

```
systemctl enable sshd.service
```

Refer to Chapter 7, Services and Daemons for more information on how to configure services in Fedora.

Note that if you reinstall the system, a new set of identification keys will be created. As a result, 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)!
It is also possible that the RSA host key has just been changed.

To prevent this, you can back up the relevant files from the `/etc/ssh/` directory (see Table 9.1, “System-wide configuration files” for a complete list), and restore them whenever you reinstall the system.
9.2.3. Requiring SSH for Remote Connections

For SSH to be truly effective, using insecure connection protocols 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**, and **vsftpd**.

To make sure these services are not running, type the following commands at a shell prompt:

```
systemctl stop telnet.service
systemctl stop rsh.service
systemctl stop rlogin.service
systemctl stop vsftpd.service
```

To disable running these services at startup, type:

```
systemctl disable telnet.service
systemctl disable rsh.service
systemctl disable rlogin.service
systemctl disable vsftpd.service
```

Refer to *Chapter 7, Services and Daemons* for more information on how to configure services in Fedora.

9.2.4. Using a Key-Based Authentication

To improve the system security even further, you can enforce the key-based authentication by disabling the standard password authentication. To do so, open the `/etc/ssh/sshd_config` configuration file in a text editor, and change the `PasswordAuthentication` option as follows:

```
PasswordAuthentication no
```

To be able to use **ssh**, **scp**, or **sftp** to connect to the server from a client machine, generate an authorization key pair by following the steps below. Note that keys must be generated for each user separately.

Fedora 20 uses SSH Protocol 2 and RSA keys by default (see *Section 9.1.3, “Protocol Versions”* for more information).

**Do not generate key pairs as root**

If you complete the steps as root, only root will be able to use the keys.

**Backup your ~/.ssh/ directory**

If you reinstall your system and want to keep previously generated key pair, backup the `~/.ssh/` directory. After reinstalling, copy it back to your home directory. This process can be done for all users on your system, including root.
9.2.4.1. Generating Key Pairs

To generate an RSA key pair for version 2 of the SSH protocol, follow these steps:

1. Generate an RSA key pair by typing the following at a shell prompt:

   
   ```
   ~]$
   ssh-keygen -t rsa
   Generating public/private rsa key pair.
   Enter file in which to save the key (/home/john/.ssh/id_rsa):
   
   ~]$
   
   ```

2. Press `Enter` to confirm the default location (that is, `~/.ssh/id_rsa`) for the newly created key.

3. Enter a passphrase, and confirm it by entering it again when prompted to do so. For security reasons, avoid using the same password as you use to log in to your account.

   After this, you will be presented with a message similar to this:

   ```
   Your identification has been saved in /home/john/.ssh/id_rsa.
   Your public key has been saved in /home/john/.ssh/id_rsa.pub.
   The key fingerprint is:
   The key's randomart image is:
   +---[ RSA 2048]----+
   |             E.  |
   |            . .  |
   |             o . |
   |              . .|
   |        S .    . |
   |         + o o ..|
   |          * * +oo|
   |           O +..=|
   |           o*  o.|
   +-----------------+
   
   ```

4. Change the permissions of the `~/.ssh/` directory:

   ```
   ~]$
   chmod 755 ~/.ssh
   
   ```

5. Copy the content of `~/.ssh/id_rsa.pub` into the `~/.ssh/authorized_keys` on the machine to which you want to connect, appending it to its end if the file already exists.

6. Change the permissions of the `~/.ssh/authorized_keys` file using the following command:

   ```
   ~]$
   chmod 644 ~/.ssh/authorized_keys
   
   ```

To generate a DSA key pair for version 2 of the SSH protocol, follow these steps:

1. Generate a DSA key pair by typing the following at a shell prompt:

   ```
   ~]$
   ssh-keygen -t dsa
   Generating public/private dsa key pair.
   ```
Using a Key-Based Authentication

Enter file in which to save the key (/home/john/.ssh/id_dsa):

2. Press **Enter** to confirm the default location (that is, ~/.ssh/id_dsa) for the newly created key.

3. Enter a passphrase, and confirm it by entering it again when prompted to do so. For security reasons, avoid using the same password as you use to log in to your account.

After this, you will be presented with a message similar to this:

```
Your identification has been saved in /home/john/.ssh/id_dsa.
Your public key has been saved in /home/john/.ssh/id_dsa.pub.
The key fingerprint is:
81:a1:91:a8:9f:e8:56:54:90:cc:bc:cc:27 john@penguin.example.com
The key's randomart image is:
|   . + o.        |
|   . + o        |
|   . o o        |
|   . o o        |
|   . o o        |
|   . o o        |
|   . o o        |
|   . o o        |
|   . o o        |
+-----------------+
```

4. Change the permissions of the ~/.ssh/ directory:

```bash
$ chmod 775 ~/.ssh
```

5. Copy the content of ~/.ssh/id_dsa.pub into the ~/.ssh/authorized_keys on the machine to which you want to connect, appending it to its end if the file already exists.

6. Change the permissions of the ~/.ssh/authorized_keys file using the following command:

```bash
$ chmod 644 ~/.ssh/authorized_keys
```

To generate an RSA key pair for version 1 of the SSH protocol, follow these steps:

1. Generate an RSA key pair by typing the following at a shell prompt:

```bash
$ ssh-keygen -t rsa
Generating public/private rsa1 key pair.
Enter file in which to save the key (/home/john/.ssh/identity):
```

2. Press **Enter** to confirm the default location (that is, ~/.ssh/identity) for the newly created key.

3. Enter a passphrase, and confirm it by entering it again when prompted to do so. For security reasons, avoid using the same password as you use to log into your account.

After this, you will be presented with a message similar to this:

```
Your identification has been saved in /home/john/.ssh/identity.
```
Chapter 9. OpenSSH

Your public key has been saved in /home/john/.ssh/identity.pub.
The key fingerprint is:
The key’s randomart image is:
+--[RSA1 2048]----+
|                 |
|     . .         |
|    o o          |
|     + o E       |
|    . o S        |
|       . = +   . |
|      . = . o .  |
|       . = o o .|
|       . o o =0 .|
+-----------------+

4. Change the permissions of the ~/.ssh/ directory:

```bash
~]$ chmod 755 ~/.ssh
```

5. Copy the content of ~/.ssh/identity.pub into the ~/.ssh/authorized_keys on the machine to which you want to connect, appending it to its end if the file already exists.

6. Change the permissions of the ~/.ssh/authorized_keys file using the following command:

```bash
~]$ chmod 644 ~/.ssh/authorized_keys
```

Refer to Section 9.2.4.2, “Configuring ssh-agent” for information on how to set up your system to remember the passphrase.

Never share your private key

The private key is for your personal use only, and it is important that you never give it to anyone.

9.2.4.2. Configuring ssh-agent

To store your passphrase so that you do not have to enter it each time you initiate a connection with a remote machine, you can use the ssh-agent authentication agent. To save your passphrase for a certain shell prompt, use the following command:

```bash
~]$ ssh-add
Enter passphrase for /home/john/.ssh/id_rsa:
```

Note that when you log out, your passphrase will be forgotten. You must execute the command each time you log in to a virtual console or a terminal window.

9.3. OpenSSH Clients
Make sure you have relevant packages installed

To connect to an OpenSSH server from a client machine, you must have the `openssh-clients` and `openssh` packages installed. Refer to Section 5.2.4, "Installing Packages" for more information on how to install new packages in Fedora.

9.3.1. Using the ssh Utility

`ssh` allows you to log in to a remote machine and execute commands there. It is a secure replacement for the `rlogin`, `rsh`, and `telnet` programs.

Similarly to `telnet`, to log in to a remote machine named `penguin.example.com`, type the following command at a shell prompt:

```
$ ssh penguin.example.com
```

This will log you in with the same username you are using on a local machine. If you want to specify a different one, use a command in the `ssh username@hostname` form. For example, to log in as `john`, type:

```
$ ssh john@penguin.example.com
```

The first time you initiate a connection, you will be presented with a message similar to this:

```
The authenticity of host 'penguin.example.com' can't be established.
Are you sure you want to continue connecting (yes/no)?
```

Type `yes` to confirm. You will see a notice that the server has been added to the list of known hosts, and a prompt asking for your password:

```
Warning: Permanently added 'penguin.example.com' (RSA) to the list of known hosts.
john@penguin.example.com's password:
```

Updating the host key of an SSH server

If the SSH server's host key changes, the client notifies the user that the connection cannot proceed until the server's host key is deleted from the `~/.ssh/known_hosts` file. To do so, open the file in a text editor, and remove a line containing the remote machine name at the beginning. Before doing this, however, contact the system administrator of the SSH server to verify the server is not compromised.

After entering the password, you will be provided with a shell prompt for the remote machine.

Alternatively, the `ssh` program can be used to execute a command on the remote machine without logging in to a shell prompt. The syntax for that is `ssh [username@]hostname command`. For example, if you want to execute the `whoami` command on `penguin.example.com`, type:
Chapter 9. OpenSSH

9.3.2. Using the scp Utility

scp can be used to transfer files between machines over a secure, encrypted connection. In its design, it is very similar to rcp.

To transfer a local file to a remote system, use a command in the following form:

```
scp localfile username@hostname:remotefile
```

For example, if you want to transfer `taglist.vim` to a remote machine named `penguin.example.com`, type the following at a shell prompt:

```
~$ scp taglist.vim john@penguin.example.com:.vim/plugin/taglist.vim
```

Multiple files can be specified at once. To transfer the contents of `.vim/plugin/` to the same directory on the remote machine `penguin.example.com`, type the following command:

```
~$ scp .vim/plugin/* john@penguin.example.com:.vim/plugin/
```

To transfer a remote file to the local system, use the following syntax:

```
scp username@hostname:remotefile localfile
```

For instance, to download the `.vimrc` configuration file from the remote machine, type:

```
~$ scp john@penguin.example.com:.vimrc .vimrc
```

9.3.3. Using the sftp Utility

The sftp utility can be used to open a secure, interactive FTP session. In its design, it is similar to ftp except that it uses a secure, encrypted connection.

To connect to a remote system, use a command in the following form:

```
sftp username@hostname
```
For example, to log in to a remote machine named `penguin.example.com` with `john` as a username, type:

```
~]$ sftp john@penguin.example.com
john@penguin.example.com's password:
Connected to penguin.example.com.
sftp>
```

After you enter the correct password, you will be presented with a prompt. The `sftp` utility accepts a set of commands similar to those used by `ftp` (see Table 9.3, "A selection of available sftp commands").

Table 9.3. A selection of available sftp commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ls</code> [directory]</td>
<td>List the content of a remote directory. If none is supplied, a current working directory is used by default.</td>
</tr>
<tr>
<td><code>cd</code> directory</td>
<td>Change the remote working directory to directory.</td>
</tr>
<tr>
<td><code>mkdir</code> directory</td>
<td>Create a remote directory.</td>
</tr>
<tr>
<td><code>rmdir</code> path</td>
<td>Remove a remote directory.</td>
</tr>
<tr>
<td><code>put</code> localfile[remotefile]</td>
<td>Transfer localfile to a remote machine.</td>
</tr>
<tr>
<td><code>get</code> remotefile[localfile]</td>
<td>Transfer remotefile from a remote machine.</td>
</tr>
</tbody>
</table>

For a complete list of available commands, refer to the `sftp` man page.

### 9.4. 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.

#### 9.4.1. X11 Forwarding

To open an X11 session over an SSH connection, use a command in the following form:

```
ssh -Y username@hostname
```

For example, to log in to a remote machine named `penguin.example.com` with `john` as a username, type:

```
~]$ ssh -Y john@penguin.example.com
john@penguin.example.com's password:
```

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 utility. To do this, connect to the server using `ssh` and type:
The `system-config-printer` & will appear, allowing the remote user to safely configure printing on
the remote system.

### 9.4.2. 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.

#### Using reserved port numbers

Setting up port forwarding to listen on ports below 1024 requires root level access.

To create a TCP/IP port forwarding channel which listens for connections on the localhost, use a
command in the following form:

```bash
ssh -L local-port:remote-hostname:remote-port username@hostname
```

For example, to check email on a server called `mail.example.com` using POP3 through an
encrypted connection, use the following command:

```bash
~]$ 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 email. Any requests sent to
port **1100** on the client system will be 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:

```bash
~]$ 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 email. Note that 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 (that is, 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.
A connection is only as secure as a client system

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.

9.5. 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.

9.5.1. Installed Documentation

man ssh
The manual page for ssh containing the full documentation on its usage.

man scp
The manual page for scp containing the full documentation on its usage.

man sftp
The manual page for sftp containing the full documentation on its usage.

man sshd
The manual page for sshd containing the full documentation on its usage.

man ssh-keygen
The manual page for ssh-keygen containing the full documentation on its usage.

man ssh_config
The manual page with full description of available SSH client configuration options.

man sshd_config
The manual page with full description of available SSH daemon configuration options.

9.5.2. Useful Websites

http://www.openssh.com/
The OpenSSH home page containing further documentation, frequently asked questions, links to the mailing lists, bug reports, and other useful resources.

http://www.openssl.org/
The OpenSSL home page containing further documentation, frequently asked questions, links to the mailing lists, and other useful resources.
http://www.freesshd.com/

Another implementation of an SSH server.
Part IV. Servers

This part discusses various topics related to servers such as how to set up a Web server or share files and directories over the network.
Web Servers

HTTP (Hypertext Transfer Protocol) server, or a web server, is a network service that serves content to a client over the web. This typically means web pages, but any other documents can be served as well.

10.1. The Apache HTTP Server

This section focuses on the Apache HTTP Server 2.2, a robust, full-featured open source web server developed by the Apache Software Foundation, that is included in Fedora 20. It describes the basic configuration of the httpd service, and covers advanced topics such as adding server modules, setting up virtual hosts, or configuring the secure HTTP server.

There are important differences between the Apache HTTP Server 2.2 and version 2.0, and if you are upgrading from a previous release of Fedora, you will need to update the httpd service configuration accordingly. This section reviews some of the newly added features, outlines important changes, and guides you through the update of older configuration files.

10.1.1. New Features

The Apache HTTP Server version 2.2 introduces the following enhancements:

- Improved caching modules, that is, mod_cache and mod_disk_cache.
- Support for proxy load balancing, that is, the mod_proxy_balancer module.
- Support for large files on 32-bit architectures, allowing the web server to handle files greater than 2GB.
- A new structure for authentication and authorization support, replacing the authentication modules provided in previous versions.

10.1.2. Notable Changes

Since version 2.0, few changes have been made to the default httpd service configuration:

- The following modules are no longer loaded by default: mod_cern_meta and mod_asis.
- The following module is newly loaded by default: mod_ext_filter.

10.1.3. Updating the Configuration

To update the configuration files from the Apache HTTP Server version 2.0, take the following steps:

1. Make sure all module names are correct, since they may have changed. Adjust the LoadModule directive for each module that has been renamed.

1 http://www.apache.org/
2. Recompile all third party modules before attempting to load them. This typically means authentication and authorization modules.

3. If you use the mod_userdir module, make sure the UserDir directive indicating a directory name (typically public_html) is provided.

4. If you use the Apache HTTP Secure Server, edit the /etc/httpd/conf.d/ssl.conf to enable the Secure Sockets Layer (SSL) protocol.

Note that you can check the configuration for possible errors by using the following command:

```
service httpd configtest
```

For more information on upgrading the Apache HTTP Server configuration from version 2.0 to 2.2, refer to http://httpd.apache.org/docs/2.2/upgrading.html.

### 10.1.4. Running the httpd Service

This section describes how to start, stop, restart, and check the current status of the Apache HTTP Server. To be able to use the httpd service, make sure you have the httpd installed. You can do so by using the following command as root:

```
yum install httpd
```

For more information on the concept of runlevels and how to manage system services in Fedora in general, refer to Chapter 7, Services and Daemons.

#### 10.1.4.1. Starting the Service

To run the httpd service, type the following at a shell prompt as root:

```
systemctl start httpd.service
```

If you want the service to start automatically at the boot time, use the following command:

```
systemctl enable httpd.service
```

Refer to Chapter 7, Services and Daemons for more information on how to configure services in Fedora.

#### Using the secure server

If running the Apache HTTP Server as a secure server, a password may be required after the machine boots if using an encrypted private SSL key.

#### 10.1.4.2. Stopping the Service

To stop the running httpd service, type the following at a shell prompt as root:

```
systemctl stop httpd.service
```
10.1.4.3. Restarting the Service

There are two different ways to restart the running httpd service:

1. To restart the service completely, type the following at a shell prompt as root:

   ```
systemctl restart httpd.service
   ```

   This will stop the running httpd service, and then start it again. Use this command after installing or removing a dynamically loaded module such as PHP.

2. To only reload the configuration, as root, type:

   ```
systemctl reload httpd.service
   ```

   This will cause the running httpd service to reload the configuration file. Note that any requests being currently processed will be interrupted, which may cause a client browser to display an error message or render a partial page.

3. To reload the configuration without affecting active requests, run the following command as root:

   ```
service httpd graceful
   ```

   This will cause the running httpd service to reload the configuration file. Note that any requests being currently processed will use the old configuration.

Refer to Chapter 7, Services and Daemons for more information on how to configure services in Fedora.

10.1.4.4. Checking the Service Status

To check whether the service is running, type the following at a shell prompt:

```
systemctl is-active httpd.service
``` 

Refer to Chapter 7, Services and Daemons for more information on how to configure services in Fedora.

10.1.5. Editing the Configuration Files

When the httpd service is started, by default, it reads the configuration from locations that are listed in Table 10.1, “The httpd service configuration files”.

```
Table 10.1. The httpd service configuration files

<table>
<thead>
<tr>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc/httpd/conf/</td>
<td>The main configuration file.</td>
</tr>
<tr>
<td>httpd.conf</td>
<td></td>
</tr>
<tr>
<td>/etc/httpd/conf.d/</td>
<td>An auxiliary directory for configuration files that are included in the</td>
</tr>
<tr>
<td></td>
<td>main configuration file.</td>
</tr>
</tbody>
</table>

Although the default configuration should be suitable for most situations, it is a good idea to become at least familiar with some of the more important configuration options. Note that for any changes to take effect, the web server has to be restarted first. Refer to Section 10.1.4.3, “Restarting the Service” for more information on how to restart the httpd service.

To check the configuration for possible errors, type the following at a shell prompt:

```
service httpd configtest
```

To make the recovery from mistakes easier, it is recommended that you make a copy of the original file before editing it.

### 10.1.5.1. Common httpd.conf Directives

The following directives are commonly used in the `/etc/httpd/conf/httpd.conf` configuration file:

#### `<Directory>`

The `<Directory>` directive allows you to apply certain directives to a particular directory only. It takes the following form:

```
<Directory directory>
  directive...
</Directory>
```

The `directory` can be either a full path to an existing directory in the local file system, or a wildcard expression.

This directive can be used to configure additional `cgi-bin` directories for server-side scripts located outside the directory that is specified by `ScriptAlias`. In this case, the `ExecCGI` and `AddHandler` directives must be supplied, and the permissions on the target directory must be set correctly (that is, `0755`).

**Example 10.1. Using the `<Directory>` directive**

```
<Directory /var/www/html>
  Options Indexes FollowSymLinks
  AllowOverride None
  Order allow, deny
  Allow from all
</Directory>
```

#### `<IfDefine>`

The `IfDefine` directive allows you to use certain directives only when a particular parameter is supplied on the command line. It takes the following form:
<IfDefine ![parameter]>
  directive
  ...
</IfDefine>

The *parameter* can be supplied at a shell prompt using the `-Dparameter` command line option (for example, `httpd -DEnableHome`). If the optional exclamation mark (that is, `!`) is present, the enclosed directives are used only when the parameter is *not* specified.

Example 10.2. Using the `<IfDefine>` directive

```
<IfDefine EnableHome>
  UserDir public_html
</IfDefine>
```

**<IfModule>**

The `<IfModule>` directive allows you to use certain directive only when a particular module is loaded. It takes the following form:

<IfModule ![module]>
  directive
  ...
</IfModule>

The *module* can be identified either by its name, or by the file name. If the optional exclamation mark (that is, `!`) is present, the enclosed directives are used only when the module is *not* loaded.

Example 10.3. Using the `<IfModule>` directive

```
<IfModule mod_disk_cache.c>
  CacheEnable disk /
  CacheRoot /var/cache/mod_proxy
</IfModule>
```

**<Location>**

The `<Location>` directive allows you to apply certain directives to a particular URL only. It takes the following form:

```
<Location url>
  directive
  ...
</Location>
```

The *url* can be either a path relative to the directory specified by the `DocumentRoot` directive (for example, `/server-info`), or an external URL such as `http://example.com/server-info`.

Example 10.4. Using the `<Location>` directive

```
<Location /server-info>
  SetHandler server-info
  Order deny,allow
  Deny from all
  Allow from .example.com
</Location>
```
Chapter 10. Web Servers

**<Proxy>**
The `<Proxy>` directive allows you to apply certain directives to the proxy server only. It takes the following form:

```
<Proxy pattern>
  directive
</Proxy>
```

The `pattern` can be an external URL, or a wildcard expression (for example, `http://example.com/*`).

**Example 10.5. Using the `<Proxy>` directive**

```
<Proxy *>
  Order deny,allow
  Deny from all
  Allow from .example.com
</Proxy>
```

**<VirtualHost>**
The `<VirtualHost>` directive allows you to apply certain directives to particular virtual hosts only. It takes the following form:

```
<VirtualHost address[:port]...>
  directive
  ...
</VirtualHost>
```

The `address` can be an IP address, a fully qualified domain name, or a special form as described in Table 10.2, “Available `<VirtualHost>` options”.

**Table 10.2. Available `<VirtualHost>` options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>Represents all IP addresses.</td>
</tr>
<tr>
<td><em>default</em></td>
<td>Represents unmatched IP addresses.</td>
</tr>
</tbody>
</table>

**Example 10.6. Using the `<VirtualHost>` directive**

```
<VirtualHost *:80>
  ServerAdmin webmaster@penguin.example.com
  DocumentRoot /www/docs/penguin.example.com
  ServerName penguin.example.com
  ErrorLog logs/penguin.example.com-error_log
  CustomLog logs/penguin.example.com-access_log common
</VirtualHost>
```

**AccessFileName**
The `AccessFileName` directive allows you to specify the file to be used to customize access control information for each directory. It takes the following form:
The *filename* is a name of the file to look for in the requested directory. By default, the server looks for `.htaccess`.

For security reasons, the directive is typically followed by the **Files** tag to prevent the files beginning with `.ht` from being accessed by web clients. This includes the `.htaccess` and `.htpasswd` files.

### Example 10.7. Using the AccessFileName directive

```html
AccessFileName .htaccess

<Files ~ "^\.ht">
  Order allow,deny
  Deny from all
  Satisfy All
</Files>
```

### Action

The **Action** directive allows you to specify a CGI script to be executed when a certain media type is requested. It takes the following form:

```html
Action content-type path
```

The *content-type* has to be a valid MIME type such as `text/html`, `image/png`, or `application/pdf`. The *path* refers to an existing CGI script, and must be relative to the directory specified by the **DocumentRoot** directive (for example, `/cgi-bin/process-image.cgi`).

### Example 10.8. Using the Action directive

```html
Action image/png /cgi-bin/process-image.cgi
```

### AddDescription

The **AddDescription** directive allows you to specify a short description to be displayed in server-generated directory listings for a given file. It takes the following form:

```html
AddDescription "description" filename...
```

The *description* should be a short text enclosed in double quotes (that is, `"`). The *filename* can be a full file name, a file extension, or a wildcard expression.

### Example 10.9. Using the AddDescription directive

```html
AddDescription "GZIP compressed tar archive" .tgz
```
AddEncoding

The **AddEncoding** directive allows you to specify an encoding type for a particular file extension. It takes the following form:

```
AddEncoding encoding extension...
```

The *encoding* has to be a valid MIME encoding such as `x-compress`, `x-gzip`, etc. The *extension* is a case sensitive file extension, and is conventionally written with a leading dot (for example, `.gz`).

This directive is typically used to instruct web browsers to decompress certain file types as they are downloaded.

**Example 10.10. Using the AddEncoding directive**

```
AddEncoding x-gzip .gz .tgz
```

AddHandler

The **AddHandler** directive allows you to map certain file extensions to a selected handler. It takes the following form:

```
AddHandler handler extension...
```

The *handler* has to be a name of previously defined handler. The *extension* is a case sensitive file extension, and is conventionally written with a leading dot (for example, `.cgi`).

This directive is typically used to treat files with the `.cgi` extension as CGI scripts regardless of the directory they are in. Additionally, it is also commonly used to process server-parsed HTML and image-map files.

**Example 10.11. Using the AddHandler option**

```
AddHandler cgi-script .cgi
```

AddIcon

The **AddIcon** directive allows you to specify an icon to be displayed for a particular file in server-generated directory listings. It takes the following form:

```
AddIcon path pattern...
```

The *path* refers to an existing icon file, and must be relative to the directory specified by the **DocumentRoot** directive (for example, `/icons/folder.png`). The *pattern* can be a file name, a file extension, a wildcard expression, or a special form as described in the following table:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>^^DIRECTORY^^</code></td>
<td>Represents a directory.</td>
</tr>
<tr>
<td><code>^^BLANKICON^^</code></td>
<td>Represents a blank line.</td>
</tr>
</tbody>
</table>
AddIconByEncoding

The `AddIconByEncoding` directive allows you to specify an icon to be displayed for a particular encoding type in server-generated directory listings. It takes the following form:

```plaintext
AddIconByEncoding path encoding...
```

The `path` refers to an existing icon file, and must be relative to the directory specified by the `DocumentRoot` directive (for example, `/icons/compressed.png`). The `encoding` has to be a valid MIME encoding such as `x-compress`, `x-gzip`, etc.

Example 10.13. Using the AddIconByEncoding directive

```plaintext
AddIconByEncoding /icons/compressed.png x-compress x-gzip
```

AddIconByType

The `AddIconByType` directive allows you to specify an icon to be displayed for a particular media type in server-generated directory listings. It takes the following form:

```plaintext
AddIconByType path content-type...
```

The `path` refers to an existing icon file, and must be relative to the directory specified by the `DocumentRoot` directive (for example, `/icons/text.png`). The `content-type` has to be either a valid MIME type (for example, `text/html` or `image/png`), or a wildcard expression such as `text/*`, `image/*`, etc.

Example 10.14. Using the AddIconByType directive

```plaintext
AddIconByType /icons/video.png video/*
```

AddLanguage

The `AddLanguage` directive allows you to associate a file extension with a specific language. It takes the following form:

```plaintext
AddLanguage language extension...
```

The `language` has to be a valid MIME language such as `cs`, `en`, or `fr`. The `extension` is a case sensitive file extension, and is conventionally written with a leading dot (for example, `.cs`).

This directive is especially useful for web servers that serve content in multiple languages based on the client’s language settings.

Example 10.15. Using the AddLanguage directive

```plaintext
AddLanguage cs .cs .cz
```
Chapter 10. Web Servers

AddType

The **AddType** directive allows you to define or override the media type for a particular file extension. It takes the following form:

```
AddType content-type extension...
```

The *content-type* has to be a valid MIME type such as `text/html`, `image/png`, etc. The *extension* is a case sensitive file extension, and is conventionally written with a leading dot (for example, `.cs`).

Example 10.16. Using the AddType directive

```
AddType application/x-gzip .gz .tgz
```

Alias

The **Alias** directive allows you to refer to files and directories outside the default directory specified by the **DocumentRoot** directive. It takes the following form:

```
Alias url-path real-path
```

The *url-path* must be relative to the directory specified by the **DocumentRoot** directive (for example, `/images/`). The *real-path* is a full path to a file or directory in the local file system.

This directive is typically followed by the **Directory** tag with additional permissions to access the target directory. By default, the `/icons/` alias is created so that the icons from `/var/www/icons/` are displayed in server-generated directory listings.

Example 10.17. Using the Alias directive

```
Alias /icons/ /var/www/icons/

<Directory "/var/www/icons"
  Options Indexes MultiViews FollowSymLinks
  AllowOverride None
  Order allow,deny
  Allow from all
  <Directory>
```

Allow

The **Allow** directive allows you to specify which clients have permission to access a given directory. It takes the following form:

```
Allow from client...
```

The *client* can be a domain name, an IP address (both full and partial), a *network/netmask* pair, or **all** for all clients.

Example 10.18. Using the Allow directive

```
Allow from 192.168.1.0/255.255.255.0
```
**AllowOverride**

The `AllowOverride` directive allows you to specify which directives in a `.htaccess` file can override the default configuration. It takes the following form:

```
AllowOverride type...
```

The `type` has to be one of the available grouping options as described in Table 10.4, “Available AllowOverride options”.

### Table 10.4. Available AllowOverride options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>All</strong></td>
<td>All directives in <code>.htaccess</code> are allowed to override earlier configuration settings.</td>
</tr>
<tr>
<td><strong>None</strong></td>
<td>No directive in <code>.htaccess</code> is allowed to override earlier configuration settings.</td>
</tr>
<tr>
<td><strong>AuthConfig</strong></td>
<td>Allows the use of authorization directives such as <code>AuthName</code>, <code>AuthType</code>, or <code>Require</code>.</td>
</tr>
<tr>
<td><strong>FileInfo</strong></td>
<td>Allows the use of file type, metadata, and mod_rewrite directives such as <code>DefaultType</code>, <code>RequestHeader</code>, or <code>RewriteEngine</code>, as well as the <code>Action</code> directive.</td>
</tr>
<tr>
<td><strong>Indexes</strong></td>
<td>Allows the use of directory indexing directives such as <code>AddDescription</code>, <code>AddIcon</code>, or <code>FancyIndexing</code>.</td>
</tr>
<tr>
<td><strong>Limit</strong></td>
<td>Allows the use of host access directives, that is, <code>Allow</code>, <code>Deny</code>, and <code>Order</code>.</td>
</tr>
<tr>
<td><strong>Options</strong></td>
<td>Allows the use of the <code>Options</code> directive. Additionally, you can provide a comma-separated list of options to customize which options can be set using this directive.</td>
</tr>
</tbody>
</table>

**Example 10.19. Using the AllowOverride directive**

```
AllowOverride FileInfo AuthConfig Limit
```

**BrowserMatch**

The `BrowserMatch` directive allows you to modify the server behavior based on the client’s web browser type. It takes the following form:

```
BrowserMatch pattern variable...
```

The `pattern` is a regular expression to match the User-Agent HTTP header field. The `variable` is an environment variable that is set when the header field matches the pattern.

By default, this directive is used to deny connections to specific browsers with known issues, and to disable keepalives and HTTP header flushes for browsers that are known to have problems with these actions.

**Example 10.20. Using the BrowserMatch directive**

```
BrowserMatch "Mozilla/2" nokeepalive
```
Chapter 10. Web Servers

CacheDefaultExpire

The CacheDefaultExpire option allows you to set how long to cache a document that does not have any expiration date or the date of its last modification specified. It takes the following form:

```
CacheDefaultExpire time
```

The time is specified in seconds. The default option is 3600 (that is, one hour).

Example 10.21. Using the CacheDefaultExpire directive

```
CacheDefaultExpire 3600
```

CacheDisable

The CacheDisable directive allows you to disable caching of certain URLs. It takes the following form:

```
CacheDisable path
```

The path must be relative to the directory specified by the DocumentRoot directive (for example, /files/).

Example 10.22. Using the CacheDisable directive

```
CacheDisable /temporary
```

CacheEnable

The CacheEnable directive allows you to specify a cache type to be used for certain URLs. It takes the following form:

```
CacheEnable type url
```

The type has to be a valid cache type as described in Table 10.5, “Available cache types”. The url can be a path relative to the directory specified by the DocumentRoot directive (for example, /images/), a protocol (for example, ftp:/), or an external URL such as http://example.com/.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mem</td>
<td>The memory-based storage manager.</td>
</tr>
<tr>
<td>disk</td>
<td>The disk-based storage manager.</td>
</tr>
<tr>
<td>fd</td>
<td>The file descriptor cache.</td>
</tr>
</tbody>
</table>

Example 10.23. Using the CacheEnable directive

```
CacheEnable disk /
```
**CacheLastModifiedFactor**

The `CacheLastModifiedFactor` directive allows you to customize how long to cache a document that does not have any expiration date specified, but that provides information about the date of its last modification. It takes the following form:

```
CacheLastModifiedFactor number
```

The `number` is a coefficient to be used to multiply the time that passed since the last modification of the document. The default option is `0.1` (that is, one tenth).

**Example 10.24. Using the CacheLastModifiedFactor directive**

```
CacheLastModifiedFactor 0.1
```

**CacheMaxExpire**

The `CacheMaxExpire` directive allows you to specify the maximum amount of time to cache a document. It takes the following form:

```
CacheMaxExpire time
```

The `time` is specified in seconds. The default option is `86400` (that is, one day).

**Example 10.25. Using the CacheMaxExpire directive**

```
CacheMaxExpire 86400
```

**CacheNegotiatedDocs**

The `CacheNegotiatedDocs` directive allows you to enable caching of the documents that were negotiated on the basis of content. It takes the following form:

```
CacheNegotiatedDocs option
```

The `option` has to be a valid keyword as described in Table 10.6, “Available CacheNegotiatedDocs options”. Since the content-negotiated documents may change over time or because of the input from the requester, the default option is `off`.

**Table 10.6. Available CacheNegotiatedDocs options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>On</td>
<td>Enables caching the content-negotiated documents.</td>
</tr>
<tr>
<td>Off</td>
<td>Disables caching the content-negotiated documents.</td>
</tr>
</tbody>
</table>

**Example 10.26. Using the CacheNegotiatedDocs directive**

```
CacheNegotiatedDocs On
```

**CacheRoot**

The `CacheRoot` directive allows you to specify the directory to store cache files in. It takes the following form:
The directory must be a full path to an existing directory in the local file system. The default option is \texttt{/var/cache/mod_proxy/}.

\textbf{Example 10.27. Using the CacheRoot directive}

\begin{verbatim}
CacheRoot /var/cache/mod_proxy
\end{verbatim}

\textbf{CustomLog}

The \texttt{CustomLog} directive allows you to specify the log file name and the log file format. It takes the following form:

\begin{verbatim}
CustomLog path format
\end{verbatim}

The \texttt{path} refers to a log file, and must be relative to the directory that is specified by the \texttt{ServerRoot} directive (that is, \texttt{/etc/httpd/} by default). The \texttt{format} has to be either an explicit format string, or a format name that was previously defined using the \texttt{LogFormat} directive.

\textbf{Example 10.28. Using the CustomLog directive}

\begin{verbatim}
CustomLog logs/access_log combined
\end{verbatim}

\textbf{DefaultIcon}

The \texttt{DefaultIcon} directive allows you to specify an icon to be displayed for a file in server-generated directory listings when no other icon is associated with it. It takes the following form:

\begin{verbatim}
DefaultIcon path
\end{verbatim}

The \texttt{path} refers to an existing icon file, and must be relative to the directory specified by the \texttt{DocumentRoot} directive (for example, \texttt{/icons/unknown.png}).

\textbf{Example 10.29. Using the DefaultIcon directive}

\begin{verbatim}
DefaultIcon /icons/unknown.png
\end{verbatim}

\textbf{DefaultType}

The \texttt{DefaultType} directive allows you to specify a media type to be used in case the proper MIME type cannot be determined by the server. It takes the following form:

\begin{verbatim}
DefaultType content-type
\end{verbatim}

The \texttt{content-type} has to be a valid MIME type such as \texttt{text/html}, \texttt{image/png}, \texttt{application/pdf}, etc.
Example 10.30. Using the DefaultType directive

DefaultType text/plain

Deny

The **Deny** directive allows you to specify which clients are denied access to a given directory. It takes the following form:

```
Deny from client...
```

The `client` can be a domain name, an IP address (both full and partial), a `network/netmask` pair, or `all` for all clients.

Example 10.31. Using the Deny directive

Deny from 192.168.1.1

DirectoryIndex

The **DirectoryIndex** directive allows you to specify a document to be served to a client when a directory is requested (that is, when the URL ends with the `/` character). It takes the following form:

```
DirectoryIndex filename...
```

The `filename` is a name of the file to look for in the requested directory. By default, the server looks for `index.html`, and `index.html.var`.

Example 10.32. Using the DirectoryIndex directive

DirectoryIndex index.html index.html.var

DocumentRoot

The **DocumentRoot** directive allows you to specify the main directory from which the content is served. It takes the following form:

```
DocumentRoot directory
```

The `directory` must be a full path to an existing directory in the local file system. The default option is `/var/www/html/`.

Example 10.33. Using the DocumentRoot directive

```
DocumentRoot /var/www/html
```

ErrorDocument

The **ErrorDocument** directive allows you to specify a document or a message to be displayed as a response to a particular error. It takes the following form:
ErrorDocument  error-code  action

The *error-code* has to be a valid code such as **403** (Forbidden), **404** (Not Found), or **500** (Internal Server Error). The *action* can be either a URL (both local and external), or a message string enclosed in double quotes (that is, "").

Example 10.34. Using the ErrorDocument directive

ErrorDocument 403 "Access Denied"
ErrorDocument 404 /404-not_found.html

ErrorLog

The *ErrorLog* directive allows you to specify a file to which the server errors are logged. It takes the following form:

ErrorLog  path

The *path* refers to a log file, and can be either absolute, or relative to the directory that is specified by the *ServerRoot* directive (that is, */etc/httpd/* by default). The default option is *logs/error_log*

Example 10.35. Using the ErrorLog directive

ErrorLog  logs/error_log

ExtendedStatus

The *ExtendedStatus* directive allows you to enable detailed server status information. It takes the following form:

ExtendedStatus  option

The *option* has to be a valid keyword as described in Table 10.7, “Available ExtendedStatus options”. The default option is **Off**.

Table 10.7. Available ExtendedStatus options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>On</td>
<td>Enables generating the detailed server status.</td>
</tr>
<tr>
<td>Off</td>
<td>Disables generating the detailed server status.</td>
</tr>
</tbody>
</table>

Example 10.36. Using the ExtendedStatus directive

ExtendedStatus  On

Group

The *Group* directive allows you to specify the group under which the *httpd* service will run. It takes the following form:
The `group` has to be an existing UNIX group. The default option is `apache`.

Note that `Group` is no longer supported inside `<VirtualHost>`, and has been replaced by the `SuexecUserGroup` directive.

```
Example 10.37. Using the Group directive
Group apache
```

**HeaderName**

The `HeaderName` directive allows you to specify a file to be prepended to the beginning of the server-generated directory listing. It takes the following form:

```
HeaderName filename
```

The `filename` is a name of the file to look for in the requested directory. By default, the server looks for `HEADER.html`.

```
Example 10.38. Using the HeaderName directive
HeaderName HEADER.html
```

**HostnameLookups**

The `HostnameLookups` directive allows you to enable automatic resolving of IP addresses. It takes the following form:

```
HostnameLookups option
```

The `option` has to be a valid keyword as described in Table 10.8, "Available HostnameLookups options". To conserve resources on the server, the default option is `Off`.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>On</td>
<td>Enables resolving the IP address for each connection so that the hostname can be logged. However, this also adds a significant processing overhead.</td>
</tr>
<tr>
<td>Double</td>
<td>Enables performing the double-reverse DNS lookup. In comparison to the above option, this adds even more processing overhead.</td>
</tr>
<tr>
<td>Off</td>
<td>Disables resolving the IP address for each connection.</td>
</tr>
</tbody>
</table>

Note that when the presence of hostnames is required in server log files, it is often possible to use one of the many log analyzer tools that perform the DNS lookups more efficiently.

```
Example 10.39. Using the HostnameLookups directive
HostnameLookups Off
```
Include

The Include directive allows you to include other configuration files. It takes the following form:

```
Include filename
```

The filename can be an absolute path, a path relative to the directory specified by the ServerRoot directive, or a wildcard expression. All configuration files from the /etc/httpd/conf.d/ directory are loaded by default.

```
Example 10.40. Using the Include directive

Include conf.d/*.conf
```

IndexIgnore

The IndexIgnore directive allows you to specify a list of file names to be omitted from the server-generated directory listings. It takes the following form:

```
IndexIgnore filename...
```

The filename option can be either a full file name, or a wildcard expression.

```
Example 10.41. Using the IndexIgnore directive

IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
```

IndexOptions

The IndexOptions directive allows you to customize the behavior of server-generated directory listings. It takes the following form:

```
IndexOptions option...
```

The option has to be a valid keyword as described in Table 10.9, "Available directory listing options". The default options are Charset=UTF-8, FancyIndexing, HTMLTable, NameWidth=*, and VersionSort.

```
Table 10.9. Available directory listing options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charset=encoding</td>
<td>Specifies the character set of a generated web page. The encoding has to be a valid character set such as UTF-8 or ISO-8859-2.</td>
</tr>
<tr>
<td>Type=content-type</td>
<td>Specifies the media type of a generated web page. The content-type has to be a valid MIME type such as text/html or text/plain.</td>
</tr>
<tr>
<td>DescriptionWidth=value</td>
<td>Specifies the width of the description column. The value can be either a number of characters, or an asterisk (that is, *) to adjust the width automatically.</td>
</tr>
</tbody>
</table>
```
## Editing the Configuration Files

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FancyIndexing</td>
<td>Enables advanced features such as different icons for certain files or possibility to re-sort a directory listing by clicking on a column header.</td>
</tr>
<tr>
<td>FolderFirst</td>
<td>Enables listing directories first, always placing them above files.</td>
</tr>
<tr>
<td>HTMLTable</td>
<td>Enables the use of HTML tables for directory listings.</td>
</tr>
<tr>
<td>IconsAreLinks</td>
<td>Enables using the icons as links.</td>
</tr>
<tr>
<td>IconHeight=value</td>
<td>Specifies an icon height. The value is a number of pixels.</td>
</tr>
<tr>
<td>IconWidth=value</td>
<td>Specifies an icon width. The value is a number of pixels.</td>
</tr>
<tr>
<td>IgnoreCase</td>
<td>Enables sorting files and directories in a case-sensitive manner.</td>
</tr>
<tr>
<td>IgnoreClient</td>
<td>Disables accepting query variables from a client.</td>
</tr>
<tr>
<td>NameWidth=value</td>
<td>Specifies the width of the file name column. The value can be either a number of characters, or an asterisk (that is, *) to adjust the width automatically.</td>
</tr>
<tr>
<td>ScanHTMLTitles</td>
<td>Enables parsing the file for a description (that is, the title element) in case it is not provided by the AddDescription directive.</td>
</tr>
<tr>
<td>ShowForbidden</td>
<td>Enables listing the files with otherwise restricted access.</td>
</tr>
<tr>
<td>SuppressColumnSorting</td>
<td>Disables re-sorting a directory listing by clicking on a column header.</td>
</tr>
<tr>
<td>SuppressDescription</td>
<td>Disables reserving a space for file descriptions.</td>
</tr>
<tr>
<td>SuppressHTMLPreamble</td>
<td>Disables the use of standard HTML preamble when a file specified by the HeaderName directive is present.</td>
</tr>
<tr>
<td>SuppressIcon</td>
<td>Disables the use of icons in directory listings.</td>
</tr>
<tr>
<td>SuppressLastModified</td>
<td>Disables displaying the date of the last modification field in directory listings.</td>
</tr>
<tr>
<td>SuppressRules</td>
<td>Disables the use of horizontal lines in directory listings.</td>
</tr>
<tr>
<td>SuppressSize</td>
<td>Disables displaying the file size field in directory listings.</td>
</tr>
<tr>
<td>TrackModified</td>
<td>Enables returning the Last-Modified and ETag values in the HTTP header.</td>
</tr>
<tr>
<td>VersionSort</td>
<td>Enables sorting files that contain a version number in the expected manner.</td>
</tr>
<tr>
<td>XHTML</td>
<td>Enables the use of XHTML 1.0 instead of the default HTML 3.2.</td>
</tr>
</tbody>
</table>

**Example 10.42. Using the IndexOptions directive**

```text
IndexOptions FancyIndexing VersionSort NameWidth=* HTMLTable Charset=UTF-8
```

**KeepAlive**

The KeepAlive directive allows you to enable persistent connections. It takes the following form:

```text
KeepAlive option
```

The `option` has to be a valid keyword as described in Table 10.10, "Available KeepAlive options". The default option is Off.
Table 10.10. Available KeepAlive options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>On</td>
<td>Enables the persistent connections. In this case, the server will accept more than one request per connection.</td>
</tr>
<tr>
<td>Off</td>
<td>Disables the keep-alive connections.</td>
</tr>
</tbody>
</table>

Note that when the persistent connections are enabled, on a busy server, the number of child processes can increase rapidly and eventually reach the maximum limit, slowing down the server significantly. To reduce the risk, it is recommended that you set `KeepAliveTimeout` to a low number, and monitor the `/var/log/httpd/logs/error_log` log file carefully.

Example 10.43. Using the KeepAlive directive

```
KeepAlive Off
```

**KeepAliveTimeout**

The `KeepAliveTimeout` directive allows you to specify the amount of time to wait for another request before closing the connection. It takes the following form:

```
KeepAliveTimeout time
```

The `time` is specified in seconds. The default option is 15.

Example 10.44. Using the KeepAliveTimeout directive

```
KeepAliveTimeout 15
```

**LanguagePriority**

The `LanguagePriority` directive allows you to customize the precedence of languages. It takes the following form:

```
LanguagePriority language...
```

The `language` has to be a valid MIME language such as `cs`, `en`, or `fr`.

This directive is especially useful for web servers that serve content in multiple languages based on the client's language settings.

Example 10.45. Using the LanguagePriority directive

```
LanguagePriority sk cs en
```

**Listen**

The `Listen` directive allows you to specify IP addresses or ports to listen to. It takes the following form:

```
Listen [ip-address:]port [protocol]
```
The \textit{ip-address} is optional and unless supplied, the server will accept incoming requests on a given \textit{port} from all IP addresses. Since the \textit{protocol} is determined automatically from the port number, it can be usually omitted. The default option is to listen to port 80.

Note that if the server is configured to listen to a port under 1024, only superuser will be able to start the \texttt{httpd} service.

\begin{verbatim}
Example 10.46. Using the Listen directive

Listen 80
\end{verbatim}

\textbf{LoadModule}

The \texttt{LoadModule} directive allows you to load a Dynamic Shared Object (DSO) module. It takes the following form:

\begin{verbatim}
LoadModule name path
\end{verbatim}

The \texttt{name} has to be a valid identifier of the required module. The \texttt{path} refers to an existing module file, and must be relative to the directory in which the libraries are placed (that is, /usr/lib/httpd/ on 32-bit and /usr/lib64/httpd/ on 64-bit systems by default).

Refer to Section 10.1.6, "Working with Modules" for more information on the Apache HTTP Server's DSO support.

\begin{verbatim}
Example 10.47. Using the LoadModule directive

LoadModule php5_module modules/libphp5.so
\end{verbatim}

\textbf{LogFormat}

The \texttt{LogFormat} directive allows you to specify a log file format. It takes the following form:

\begin{verbatim}
LogFormat format name
\end{verbatim}

The \texttt{format} is a string consisting of options as described in Table 10.11, "Common LogFormat options". The \texttt{name} can be used instead of the format string in the \texttt{CustomLog} directive.

\begin{table}[h]
\centering
\caption{Common LogFormat options}
\begin{tabular}{|l|l|}
\hline
Option & Description \\
\hline
%b & Represents the size of the response in bytes. \\
%h & Represents the IP address or hostname of a remote client. \\
%l & Represents the remote log name if supplied. If not, a hyphen (that is, -) is used instead. \\
%r & Represents the first line of the request string as it came from the browser or client. \\
%s & Represents the status code. \\
%s & Represents the date and time of the request. \\
%u & If the authentication is required, it represents the remote user. If not, a hyphen (that is, -) is used instead. \\
\hline
\end{tabular}
\end{table}
### Option

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| {%{field} | Represents the content of the HTTP header field. The common options include
| %Referer | the URL of the web page that referred the client to the server) and
| %User-Agent | the type of the web browser making the request). |

#### Example 10.48. Using the LogFormat directive

```
LogFormat "%h %l %u %t "%r" %>s %b" common
```

### LogLevel

The **LogLevel** directive allows you to customize the verbosity level of the error log. It takes the following form:

```
LogLevel option
```

The `option` has to be a valid keyword as described in Table 10.12, "Available LogLevel options". The default option is `warn`.

#### Table 10.12. Available LogLevel options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>emerg</td>
<td>Only the emergency situations when the server cannot perform its work are logged.</td>
</tr>
<tr>
<td>alert</td>
<td>All situations when an immediate action is required are logged.</td>
</tr>
<tr>
<td>crit</td>
<td>All critical conditions are logged.</td>
</tr>
<tr>
<td>error</td>
<td>All error messages are logged.</td>
</tr>
<tr>
<td>warn</td>
<td>All warning messages are logged.</td>
</tr>
<tr>
<td>notice</td>
<td>Even normal, but still significant situations are logged.</td>
</tr>
<tr>
<td>info</td>
<td>Various informational messages are logged.</td>
</tr>
<tr>
<td>debug</td>
<td>Various debugging messages are logged.</td>
</tr>
</tbody>
</table>

#### Example 10.49. Using the LogLevel directive

```
LogLevel warn
```

### MaxKeepAliveRequests

The **MaxKeepAliveRequests** directive allows you to specify the maximum number of requests for a persistent connection. It takes the following form:

```
MaxKeepAliveRequests number
```

A high `number` can improve the performance of the server. Note that using `0` allows unlimited number of requests. The default option is `100`.

#### Example 10.50. Using the MaxKeepAliveRequests option

```
MaxKeepAliveRequests 100
```
**NameVirtualHost**

The **NameVirtualHost** directive allows you to specify the IP address and port number for a name-based virtual host. It takes the following form:

```
NameVirtualHost ip-address[:port]
```

The *ip-address* can be either a full IP address, or an asterisk (that is, *) representing all interfaces. Note that IPv6 addresses have to be enclosed in square brackets (that is, [ and ]). The *port* is optional.

Name-based virtual hosting allows one Apache HTTP Server to serve different domains without using multiple IP addresses.

---

### Using secure HTTP connections

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.

---

**Example 10.51. Using the NameVirtualHost directive**

```
NameVirtualHost *:80
```

---

**Options**

The **Options** directive allows you to specify which server features are available in a particular directory. It takes the following form:

```
Options option...
```

The *option* has to be a valid keyword as described in **Table 10.13**, “Available server features”.

---

**Table 10.13. Available server features**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExecCGI</td>
<td>Enables the execution of CGI scripts.</td>
</tr>
<tr>
<td>FollowSymLinks</td>
<td>Enables following symbolic links in the directory.</td>
</tr>
<tr>
<td>Includes</td>
<td>Enables server-side includes.</td>
</tr>
<tr>
<td>IncludesNOEXEC</td>
<td>Enables server-side includes, but does not allow the execution of commands.</td>
</tr>
<tr>
<td>Indexes</td>
<td>Enables server-generated directory listings.</td>
</tr>
<tr>
<td>MultiViews</td>
<td>Enables content-negotiated &quot;MultiViews&quot;.</td>
</tr>
<tr>
<td>SymLinksIfOwnerMatch</td>
<td>Enables following symbolic links in the directory when both the link and</td>
</tr>
<tr>
<td></td>
<td>the target file have the same owner.</td>
</tr>
<tr>
<td>All</td>
<td>Enables all of the features above with the exception of <strong>MultiViews</strong>.</td>
</tr>
<tr>
<td>None</td>
<td>Disables all of the features above.</td>
</tr>
</tbody>
</table>
Example 10.52. Using the Options directive

Options Indexes FollowSymLinks

Order

The `Order` directive allows you to specify the order in which the `Allow` and `Deny` directives are evaluated. It takes the following form:

```
Order option
```

The `option` has to be a valid keyword as described in Table 10.14, "Available Order options". The default option is `allow,deny`.

Table 10.14. Available Order options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow,deny</td>
<td>Allow directives are evaluated first.</td>
</tr>
<tr>
<td>deny,allow</td>
<td>Deny directives are evaluated first.</td>
</tr>
</tbody>
</table>

Example 10.53. Using the Order directive

```
Order allow,deny
```

PidFile

The `PidFile` directive allows you to specify a file to which the process ID (PID) of the server is stored. It takes the following form:

```
PidFile path
```

The `path` refers to a pid file, and can be either absolute, or relative to the directory that is specified by the `ServerRoot` directive (that is, `/etc/httpd/` by default). The default option is `run/httpd.pid`.

Example 10.54. Using the PidFile directive

```
PidFile run/httpd.pid
```

ProxyRequests

The `ProxyRequests` directive allows you to enable forward proxy requests. It takes the following form:

```
ProxyRequests option
```

The `option` has to be a valid keyword as described in Table 10.15, "Available ProxyRequests options". The default option is `off`. 
**Table 10.15. Available ProxyRequests options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>On</td>
<td>Enables forward proxy requests.</td>
</tr>
<tr>
<td>Off</td>
<td>Disables forward proxy requests.</td>
</tr>
</tbody>
</table>

**Example 10.55. Using the ProxyRequests directive**

```
ProxyRequests On
```

**ReadmeName**

The **ReadmeName** directive allows you to specify a file to be appended to the end of the server-generated directory listing. It takes the following form:

```
ReadmeName filename
```

The `filename` is a name of the file to look for in the requested directory. By default, the server looks for `README.html`.

**Example 10.56. Using the ReadmeName directive**

```
ReadmeName README.html
```

**Redirect**

The **Redirect** directive allows you to redirect a client to another URL. It takes the following form:

```
Redirect [status] path url
```

The `status` is optional, and if provided, it has to be a valid keyword as described in **Table 10.16, “Available status options”**. The `path` refers to the old location, and must be relative to the directory specified by the **DocumentRoot** directive (for example, `/docs`). The `url` refers to the current location of the content (for example, `http://docs.example.com`).

**Table 10.16. Available status options**

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>permanent</td>
<td>Indicates that the requested resource has been moved permanently. The 301 (Moved Permanently) status code is returned to a client.</td>
</tr>
<tr>
<td>temp</td>
<td>Indicates that the requested resource has been moved only temporarily. The 302 (Found) status code is returned to a client.</td>
</tr>
<tr>
<td>seeother</td>
<td>Indicates that the requested resource has been replaced. The 303 (See Other) status code is returned to a client.</td>
</tr>
<tr>
<td>gone</td>
<td>Indicates that the requested resource has been removed permanently. The 410 (Gone) status is returned to a client.</td>
</tr>
</tbody>
</table>

Note that for more advanced redirection techniques, you can use the `mod_rewrite` module that is part of the Apache HTTP Server installation.
Example 10.57. Using the Redirect directive

```
Redirect permanent /docs http://docs.example.com
```

### ScriptAlias

The **ScriptAlias** directive allows you to specify the location of CGI scripts. It takes the following form:

```
ScriptAlias url-path real-path
```

The `url-path` must be relative to the directory specified by the `DocumentRoot` directive (for example, `/cgi-bin/`). The `real-path` is a full path to a file or directory in the local file system.

This directive is typically followed by the `Directory` tag with additional permissions to access the target directory. By default, the `/cgi-bin/` alias is created so that the scripts located in the `/var/www/cgi-bin/` are accessible.

The **ScriptAlias** directive is used for security reasons to prevent CGI scripts from being viewed as ordinary text documents.

Example 10.58. Using the ScriptAlias directive

```
ScriptAlias /cgi-bin/ /var/www/cgi-bin/

<Directory "/var/www/cgi-bin/
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>
```

### ServerAdmin

The **ServerAdmin** directive allows you to specify the email address of the server administrator to be displayed in server-generated web pages. It takes the following form:

```
ServerAdmin email
```

The default option is `root@localhost`.

This directive is commonly set to `webmaster@hostname`, where `hostname` is the address of the server. Once set, alias `webmaster` to the person responsible for the web server in `/etc/aliases`, and as superuser, run the `newaliases` command.

Example 10.59. Using the ServerAdmin directive

```
ServerAdmin webmaster@penguin.example.com
```

### ServerName

The **ServerName** directive allows you to specify the hostname and the port number of a web server. It takes the following form:
The `hostname` has to be a **fully qualified domain name** (FQDN) of the server. The `port` is optional, but when supplied, it has to match the number specified by the `Listen` directive.

When using this directive, make sure that the IP address and server name pair are included in the `/etc/hosts` file.

**Example 10.60. Using the ServerName directive**

```bash
ServerName penguin.example.com:80
```

**ServerRoot**

The `ServerRoot` directive allows you to specify the directory in which the server operates. It takes the following form:

**Example 10.61. Using the ServerRoot directive**

```bash
ServerRoot /etc/httpd
```

**ServerSignature**

The `ServerSignature` directive allows you to enable displaying information about the server on server-generated documents. It takes the following form:

**Example 10.62. Using the ServerSignature directive**

```bash
ServerSignature On
```

The **option** has to be a valid keyword as described in Table 10.17, “Available ServerSignature options”. The default option is **On**.

**Table 10.17. Available ServerSignature options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>On</td>
<td>Enables appending the server name and version to server-generated pages.</td>
</tr>
<tr>
<td>Off</td>
<td>Disables appending the server name and version to server-generated pages.</td>
</tr>
<tr>
<td>EMail</td>
<td>Enables appending the server name, version, and the email address of the system administrator as specified by the <code>ServerAdmin</code> directive to server-generated pages.</td>
</tr>
</tbody>
</table>
ServerTokens

The `ServerTokens` directive allows you to customize what information are included in the Server response header. It takes the following form:

```
ServerTokens option
```

The `option` has to be a valid keyword as described in Table 10.18, "Available ServerTokens options". The default option is `OS`.

Table 10.18. Available ServerTokens options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prod</td>
<td>Includes the product name only (that is, Apache).</td>
</tr>
<tr>
<td>Major</td>
<td>Includes the product name and the major version of the server (for example, 2).</td>
</tr>
<tr>
<td>Minor</td>
<td>Includes the product name and the minor version of the server (for example, 2.2).</td>
</tr>
<tr>
<td>Min</td>
<td>Includes the product name and the minimal version of the server (for example, 2.2.15).</td>
</tr>
<tr>
<td>OS</td>
<td>Includes the product name, the minimal version of the server, and the type of the operating system it is running on (for example, Red Hat).</td>
</tr>
<tr>
<td>Full</td>
<td>Includes all the information above along with the list of loaded modules.</td>
</tr>
</tbody>
</table>

Note that for security reasons, it is recommended to reveal as little information about the server as possible.

Example 10.63. Using the ServerTokens directive

```
ServerTokens Prod
```

SuexecUserGroup

The `SuexecUserGroup` directive allows you to specify the user and group under which the CGI scripts will be run. It takes the following form:

```
SuexecUserGroup user group
```

The `user` has to be an existing user, and the `group` must be a valid UNIX group.

For security reasons, the CGI scripts should not be run with root privileges. Note that in `<VirtualHost>`, `SuexecUserGroup` replaces the `User` and `Group` directives.

Example 10.64. Using the SuexecUserGroup directive

```
SuexecUserGroup apache apache
```

Timeout

The `Timeout` directive allows you to specify the amount of time to wait for an event before closing a connection. It takes the following form:

```
Timeout time
```
The `time` is specified in seconds. The default option is 60.

**Example 10.65. Using the Timeout directive**

```
Timeout 60
```

**TypesConfig**

The **TypesConfig** allows you to specify the location of the MIME types configuration file. It takes the following form:

```
TypesConfig path
```

The `path` refers to an existing MIME types configuration file, and can be either absolute, or relative to the directory that is specified by the `ServerRoot` directive (that is, `/etc/httpd/` by default). The default option is `/etc/mime.types`.

Note that instead of editing `/etc/mime.types`, the recommended way to add MIME type mapping to the Apache HTTP Server is to use the **AddType** directive.

**Example 10.66. Using the TypesConfig directive**

```
TypesConfig /etc/mime.types
```

**UseCanonicalName**

The **UseCanonicalName** allows you to specify the way the server refers to itself. It takes the following form:

```
UseCanonicalName option
```

The `option` has to be a valid keyword as described in **Table 10.19, “Available UseCanonicalName options”**. The default option is Off.

**Table 10.19. Available UseCanonicalName options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>On</strong></td>
<td>Enables the use of the name that is specified by the <code>ServerName</code> directive.</td>
</tr>
<tr>
<td><strong>Off</strong></td>
<td>Disables the use of the name that is specified by the <code>ServerName</code> directive. The hostname and port number provided by the requesting client are used instead.</td>
</tr>
<tr>
<td><strong>DNS</strong></td>
<td>Disables the use of the name that is specified by the <code>ServerName</code> directive. The hostname determined by a reverse DNS lookup is used instead.</td>
</tr>
</tbody>
</table>

**Example 10.67. Using the UseCanonicalName directive**

```
UseCanonicalName Off
```
Chapter 10. Web Servers

User
The `User` directive allows you to specify the user under which the `httpd` service will run. It takes the following form:

```
User user
```

The `user` has to be an existing UNIX user. The default option is `apache`.

For security reasons, the `httpd` service should not be run with `root` privileges. Note that `User` is no longer supported inside `<VirtualHost>`, and has been replaced by the `SuexecUserGroup` directive.

Example 10.68. Using the User directive

```
User apache
```

UserDir
The `UserDir` directive allows you to enable serving content from users' home directories. It takes the following form:

```
UserDir option
```

The `option` can be either a name of the directory to look for in user's home directory (typically `public_html`), or a valid keyword as described in Table 10.20, "Available UserDir options". The default option is `disabled`.

Table 10.20. Available UserDir options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled user...</td>
<td>Enables serving content from home directories of given users.</td>
</tr>
<tr>
<td>disabled [user...]</td>
<td>Disables serving content from home directories, either for all users, or, if a space separated list of users is supplied, for given users only.</td>
</tr>
</tbody>
</table>

Set the correct permissions

In order for the web server to access the content, the permissions on relevant directories and files must be set correctly. Make sure that all users are able to access the home directories, and that they can access and read the content of the directory specified by the `UserDir` directive. For example, to allow access to `public_html/` in the home directory of user `joe`, type the following at a shell prompt as `root`:

```
~$ chmod a+x /home/joe/
~$ chmod a+rx /home/joe/public_html/
```

All files in this directory must be set accordingly.
Example 10.69. Using the UserDir directive

```
UserDir public_html
```

### 10.1.5.2. Common ssl.conf Directives

The Secure Sockets Layer (SSL) directives allow you to customize the behavior of the Apache HTTP Secure Server, and in most cases, they are configured appropriately during the installation. Be careful when changing these settings, as incorrect configuration can lead to security vulnerabilities.

The following directive is commonly used in `/etc/httpd/conf.d/ssl.conf`:

**SetEnvIf**

The `SetEnvIf` directive allows you to set environment variables based on the headers of incoming connections. It takes the following form:

```
SetEnvIf option pattern [!]variable[=value]...
```

The `option` can be either a HTTP header field, a previously defined environment variable name, or a valid keyword as described in Table 10.21, “Available SetEnvIf options”. The `pattern` is a regular expression. The `variable` is an environment variable that is set when the option matches the pattern. If the optional exclamation mark (that is, `!`) is present, the variable is removed instead of being set.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote_Host</td>
<td>Refers to the client's hostname.</td>
</tr>
<tr>
<td>Remote_Addr</td>
<td>Refers to the client's IP address.</td>
</tr>
<tr>
<td>Server_Addr</td>
<td>Refers to the server's IP address.</td>
</tr>
<tr>
<td>Request_Method</td>
<td>Refers to the request method (for example, GET).</td>
</tr>
<tr>
<td>Request_Protocol</td>
<td>Refers to the protocol name and version (for example, HTTP/1.1).</td>
</tr>
<tr>
<td>Request_URI</td>
<td>Refers to the requested resource.</td>
</tr>
</tbody>
</table>

The `SetEnvIf` directive is used to disable HTTP keepalives, and to allow SSL to close the connection without a closing notification from the client browser. This is necessary for certain web browsers that do not reliably shut down the SSL connection.

Example 10.70. Using the SetEnvIf directive

```
SetEnvIf User-Agent ".*MSIE.*" \
nokeepalive ssl-unclean-shutdown \ 
downgrade-1.0 force-response-1.0
```

Note that for the `/etc/httpd/conf.d/ssl.conf` file to be present, the `mod_ssl` needs to be installed. Refer to Section 10.1.8, “Setting Up an SSL Server” for more information on how to install and configure an SSL server.
10.1.5.3. Common Multi-Processing Module Directives

The Multi-Processing Module (MPM) directives allow you to customize the behavior of a particular MPM specific server-pool. Since its characteristics differ depending on which MPM is used, the directives are embedded in `IfModule`. By default, the server-pool is defined for both the `prefork` and `worker` MPMS.

The following MPM directives are commonly used in `/etc/httpd/conf/httpd.conf`:

**MaxClients**

The `MaxClients` directive allows you to specify the maximum number of simultaneously connected clients to process at one time. It takes the following form:

```
MaxClients number
```

A high `number` can improve the performance of the server, although it is not recommended to exceed 256 when using the `prefork` MPM.

**Example 10.71. Using the MaxClients directive**

```
MaxClients 256
```

**MaxRequestsPerChild**

The `MaxRequestsPerChild` directive allows you to specify the maximum number of request a child process can serve before it dies. It takes the following form:

```
MaxRequestsPerChild number
```

Setting the `number` to 0 allows unlimited number of requests.

The `MaxRequestsPerChild` directive is used to prevent long-lived processes from causing memory leaks.

**Example 10.72. Using the MaxRequestsPerChild directive**

```
MaxRequestsPerChild 4000
```

**MaxSpareServers**

The `MaxSpareServers` directive allows you to specify the maximum number of spare child processes. It takes the following form:

```
MaxSpareServers number
```

This directive is used by the `prefork` MPM only.

**Example 10.73. Using the MaxSpareServers directive**

```
MaxSpareServers 20
```
MaxSpareThreads

The `MaxSpareThreads` directive allows you to specify the maximum number of spare server threads. It takes the following form:

```
MaxSpareThreads number
```

The `number` must be greater than or equal to the sum of `MinSpareThreads` and `ThreadsPerChild`. This directive is used by the `worker` MPM only.

Example 10.74. Using the MaxSpareThreads directive

```
MaxSpareThreads 75
```

MinSpareServers

The `MinSpareServers` directive allows you to specify the minimum number of spare child processes. It takes the following form:

```
MinSpareServers number
```

Note that a high `number` can create a heavy processing load on the server. This directive is used by the `prefork` MPM only.

Example 10.75. Using the MinSpareServers directive

```
MinSpareServers 5
```

MinSpareThreads

The `MinSpareThreads` directive allows you to specify the minimum number of spare server threads. It takes the following form:

```
MinSpareThreads number
```

This directive is used by the `worker` MPM only.

Example 10.76. Using the MinSpareThreads directive

```
MinSpareThreads 75
```

StartServers

The `StartServers` directive allows you to specify the number of child processes to create when the service is started. It takes the following form:

```
StartServers number
```

Since the child processes are dynamically created and terminated according to the current traffic load, it is usually not necessary to change this value.
Example 10.77. Using the StartServers directive

StartServers 8

**ThreadsPerChild**

The **ThreadsPerChild** directive allows you to specify the number of threads a child process can create. It takes the following form:

```
ThreadsPerChild number
```

This directive is used by the worker MPM only.

Example 10.78. Using the ThreadsPerChild directive

ThreadsPerChild 25

### 10.1.6. Working with Modules

Being a modular application, the httpd service is distributed along with a number of *Dynamic Shared Objects* (DSOs), which can be dynamically loaded or unloaded at runtime as necessary. By default, these modules are located in `/usr/lib/httpd/modules/` on 32-bit and in `/usr/lib64/httpd/modules/` on 64-bit systems.

#### 10.1.6.1. Loading a Module

To load a particular DSO module, use the **LoadModule** directive as described in *Section 10.1.5.1, “Common httpd.conf Directives”*. Note that modules provided by a separate package often have their own configuration file in the `/etc/httpd/conf.d/` directory.

Example 10.79. Loading the mod_ssl DSO

```
LoadModule ssl_module modules/mod_ssl.so
```

Once you are finished, restart the web server to reload the configuration. Refer to *Section 10.1.4.3, “Restarting the Service”* for more information on how to restart the httpd service.

#### 10.1.6.2. Writing a Module

If you intend to create a new DSO module, make sure you have the *httpd-devel* package installed. To do so, type the following at a shell prompt as root:

```
yum install httpd-devel
```

This package contains the include files, the header files, and the **APache eXtenSion (apxs)** utility required to compile a module.

Once written, you can build the module with the following command:
Draft

Setting Up Virtual Hosts

If the build was successful, you should be able to load the module the same way as any other module that is distributed with the Apache HTTP Server.

10.1.7. Setting Up 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.

To create a name-based virtual host, find the virtual host container provided in `/etc/httpd/conf/httpd.conf` as an example, remove the hash sign (that is, #) from the beginning of each line, and customize the options according to your requirements as shown in Example 10.80, "Sample virtual host configuration".

Example 10.80. Sample virtual host configuration

```
NameVirtualHost penguin.example.com:80

<VirtualHost penguin.example.com:80>
    ServerAdmin webmaster@penguin.example.com
    DocumentRoot /www/docs/penguin.example.com
    ServerName penguin.example.com:80
    ErrorLog logs/penguin.example.com-error_log
    CustomLog logs/penguin.example.com-access_log common
</VirtualHost>
```

Note that `ServerName` must be a valid DNS name assigned to the machine. The `<VirtualHost>` container is highly customizable, and accepts most of the directives available within the main server configuration. Directives that are not supported within this container include `User` and `Group`, which were replaced by `SuexecUserGroup`.

Changing the port number

If you configure a virtual host to listen on a non-default port, make sure you update the `Listen` directive in the global settings section of the `/etc/httpd/conf/httpd.conf` file accordingly.

To activate a newly created virtual host, the web server has to be restarted first. Refer to Section 10.1.4.3, "Restarting the Service" for more information on how to restart the `httpd` service.

10.1.8. Setting Up an SSL Server

Secure Sockets Layer (SSL) is a cryptographic protocol that allows a server and a client to communicate securely. Along with its extended and improved version called Transport Layer Security (TLS), it ensures both privacy and data integrity. The Apache HTTP Server in combination with `mod_ssl`, a module that uses the OpenSSL toolkit to provide the SSL/TLS support, is commonly referred to as the SSL server.

Unlike a regular HTTP connection that can be read and possibly modified by anybody who is able to intercept it, the use of `mod_ssl` prevents any inspection or modification of the transmitted content.
This section provides basic information on how to enable this module in the Apache HTTP Server configuration, and guides you through the process of generating private keys and self-signed certificates.

**10.1.8.1. An Overview of Certificates and Security**

Secure communication is based on the use of keys. In conventional or *symmetric cryptography*, both ends of the transaction have the same key they can use to decode each other’s transmissions. On the other hand, in public or *asymmetric cryptography*, two keys co-exist: a *private key* that is kept a secret, and a *public key* that is usually shared with the public. While the data encoded with the public key can only be decoded with the private key, data encoded with the private key can in turn only be decoded with the public key.

To provide secure communications using SSL, an SSL server must use a digital certificate signed by a *Certificate Authority* (CA). The certificate lists various attributes of the server (that is, the server hostname, the name of the company, its location, etc.), and the signature produced using the CA’s private key. This signature ensures that a particular certificate authority has issued the certificate, and that the certificate has not been modified in any way.

When a web browser establishes a new SSL connection, it checks the certificate provided by the web server. If the certificate does not have a signature from a trusted CA, or if the hostname listed in the certificate does not match the hostname used to establish the connection, it refuses to communicate with the server and usually presents a user with an appropriate error message.

By default, most web browsers are configured to trust a set of widely used certificate authorities. Because of this, an appropriate CA should be chosen when setting up a secure server, so that target users can trust the connection, otherwise they will be presented with an error message, and will have to accept the certificate manually. Since encouraging users to override certificate errors can allow an attacker to intercept the connection, you should use a trusted CA whenever possible. For more information on this, see *Table 10.22, “CA lists for most common web browsers”*.

<table>
<thead>
<tr>
<th>Web Browser</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mozilla Firefox</td>
<td><a href="http://www.mozilla.org/projects/security/certs/included/">Mozilla root CA list</a>².</td>
</tr>
<tr>
<td>Opera</td>
<td><a href="http://my.opera.com/rootstore/blog/">The Opera Rootstore</a>³.</td>
</tr>
<tr>
<td>Internet Explorer</td>
<td><a href="http://support.microsoft.com/kb/931125">Windows root certificate program members</a>⁴.</td>
</tr>
</tbody>
</table>

When setting up an SSL server, you need to generate a certificate request and a private key, and then send the certificate request, proof of the company’s identity, and payment to a certificate authority. Once the CA verifies the certificate request and your identity, it will send you a signed certificate you can use with your server. Alternatively, you can create a self-signed certificate that does not contain a CA signature, and thus should be used for testing purposes only.

**10.1.8.2. Enabling the mod_ssl Module**

If you intend to set up an SSL server, make sure you have the *mod_ssl* (the *mod_ssl* module) and *openssl* (the OpenSSL toolkit) packages installed. To do so, type the following at a shell prompt as root:

⁴ [http://support.microsoft.com/kb/931125](http://support.microsoft.com/kb/931125)
yum install mod_ssl openssl

This will create the mod_ssl configuration file at `/etc/httpd/conf.d/ssl.conf`, which is included in the main Apache HTTP Server configuration file by default. For the module to be loaded, restart the httpd service as described in Section 10.1.4.3, "Restarting the Service".

10.1.8.3. Using an Existing Key and Certificate

If you have a previously created key and certificate, you can configure the SSL server to use these files instead of generating new ones. There are only two situations where this is not possible:

1. You are changing the IP address or domain name.
   Certificates are issued for a particular IP address and domain name pair. If one of these values changes, the certificate becomes invalid.

2. You have a certificate from VeriSign, and you are changing the server software.
   VeriSign, a widely used certificate authority, issues certificates for a particular software product, IP address, and domain name. Changing the software product renders the certificate invalid.

In either of the above cases, you will need to obtain a new certificate. For more information on this topic, refer to Section 10.1.8.4, "Generating a New Key and Certificate".

If you wish to use an existing key and certificate, move the relevant files to the `/etc/pki/tls/private/` and `/etc/pki/tls/certs/` directories respectively. You can do so by running the following commands as root:

- `mv key_file.key /etc/pki/tls/private/hostname.key`
- `mv certificate.crt /etc/pki/tls/certs/hostname.crt`

Then add the following lines to the `/etc/httpd/conf.d/ssl.conf` configuration file:

- `SSLCertificateFile /etc/pki/tls/certs/hostname.crt`
- `SSLCertificateKeyFile /etc/pki/tls/private/hostname.key`

To load the updated configuration, restart the httpd service as described in Section 10.1.4.3, "Restarting the Service".

Example 10.81. Using a key and certificate from the Red Hat Secure Web Server

- `mv /etc/httpd/conf/httpsd.key /etc/pki/tls/private/penguin.example.com.key`
- `mv /etc/httpd/conf/httpsd.crt /etc/pki/tls/certs/penguin.example.com.crt`

10.1.8.4. Generating a New Key and Certificate

In order to generate a new key and certificate pair, you must to have the `crypto-utils` package installed in your system. As root, you can install it by typing the following at a shell prompt:

- `yum install crypto-utils`
This package provides a set of tools to generate and manage SSL certificates and private keys, and includes `genkey`, the Red Hat Keypair Generation utility that will guide you through the key generation process.

## Replacing an existing certificate

If the server already has a valid certificate and you are replacing it with a new one, specify a different serial number. This ensures that client browsers are notified of this change, update to this new certificate as expected, and do not fail to access the page. To create a new certificate with a custom serial number, as root, use the following command instead of `genkey`:

```
openssl req -x509 -new -set_serial number -key hostname.key -out hostname.crt
```

## Remove a previously created key

If there already is a key file for a particular hostname in your system, `genkey` will refuse to start. In this case, remove the existing file using the following command as root:

```
rm /etc/pki/tls/private/hostname.key
```

To run the utility, as root, run the `genkey` command followed by the appropriate hostname (for example, `penguin.example.com`):

```
genkey hostname
```

To complete the key and certificate creation, take the following steps:

1. Review the target locations in which the key and certificate will be stored.
Use the **Tab** key to select the **Next** button, and press **Enter** to proceed to the next screen.

2. Using the **Up** and **down** arrow keys, select the suitable key size. Note that while the large key increases the security, it also increases the response time of your server. Because of this, the recommended option is **1024 bits**.

![Figure 10.2. Selecting the key size](image)

Once finished, use the **Tab** key to select the **Next** button, and press **Enter** to initiate the random bits generation process. Depending on the selected key size, this may take some time.

3. Decide whether you wish to send a certificate request to a certificate authority.

![Figure 10.3. Generating a certificate request](image)

Use the **Tab** key to select **Yes** to compose a certificate request, or **No** to generate a self-signed certificate. Then press **Enter** to confirm your choice.
4. Using the **Spacebar** key, enable ([*]) or disable ([ ])) the encryption of the private key.

![Figure 10.4. Encrypting the private key](image)

Use the **Tab** key to select the **Next** button, and press **Enter** to proceed to the next screen.

5. If you have enabled the private key encryption, enter an adequate passphrase. Note that for security reasons, it is not displayed as you type, and it must be at least five characters long.

![Figure 10.5. Entering a passphrase](image)

Use the **Tab** key to select the **Next** button, and press **Enter** to proceed to the next screen.

**Do not forget the passphrase**

Entering the correct passphrase is required in order for the server to start. If you lose it, you will need to generate a new key and certificate.
6. Customize the certificate details.

![Figure 10.6. Specifying certificate information](image)

Use the **Tab** key to select the **Next** button, and press **Enter** to finish the key generation.

7. If you have previously enabled the certificate request generation, you will be prompted to send it to a certificate authority.

![Figure 10.7. Instructions on how to send a certificate request](image)

Press **Enter** to return to a shell prompt.

Once generated, add the key and certificate locations to the `/etc/httpd/conf.d/ssl.conf` configuration file:

```
SSLCertificateFile /etc/pki/tls/certs/hostname.crt
SSLCertificateKeyFile /etc/pki/tls/private/hostname.key
```

Finally, restart the `httpd` service as described in Section 10.1.4.3, “Restarting the Service”, so that the updated configuration is loaded.
10.1.9. Additional Resources
To learn more about the Apache HTTP Server, refer to the following resources.

10.1.9.1. Installed Documentation

http://localhost/manual/
- The official documentation for the Apache HTTP Server with the full description of its directives and available modules. Note that in order to access this documentation, you must have the httpd-manual package installed, and the web server must be running.

man httpd
- The manual page for the httpd service containing the complete list of its command line options.

man genkey
- The manual page for genkey containing the full documentation on its usage.

10.1.9.2. 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 the mod_ssl module.

http://www.openssl.org/
- The OpenSSL home page containing further documentation, frequently asked questions, links to the mailing lists, and other useful resources.
Mail Servers

*Email* was born in the 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.

11.1. 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.

11.1.1. 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)*.

11.1.1.1. 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 and Sendmail SMTP programs.
11.1.2. 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).

11.1.2.1. POP

The default POP server under Fedora is Dovecot and is provided by the dovecot package.

**Installing the dovecot package**

In order to use Dovecot, first ensure the dovecot package is installed on your system by running, as root:

```
yum install dovecot
```

For more information on installing packages with Yum, refer to [Section 5.2.4, “Installing Packages”](#).

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 (Monash Directory Service) authentication. An encoded hash of the user's password is sent from the email client to the server rather than sending an unencrypted password.

- **KPOP** — POP3 with Kerberos authentication.

- **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 `pop3s` service, or by using the `/usr/sbin/stunnel` application. For more information on securing email communication, refer to [Section 11.5.1, “Securing Communication”](#).

11.1.2.2. IMAP
The default IMAP server under Fedora is **Dovecot** and is provided by the `dovecot` package. Refer to Section 11.1.2.1, “POP” for information on how to install **Dovecot**.

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 users 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. For more information on securing email communication, refer to Section 11.5.1, “Securing Communication”.

Other free, as well as commercial, IMAP clients and servers are available, many of which extend the IMAP protocol and provide additional functionality.

**11.1.2.3. Dovecot**

The `imap-login` and `pop3-login` processes which implement the IMAP and POP3 protocols are spawned by the master `dovecot` daemon included in the `dovecot` package. The use of IMAP and POP is configured through the `/etc/dovecot/dovecot.conf` configuration file; by default `dovecot` runs IMAP and POP3 together with their secure versions using SSL. To configure `dovecot` to use POP, complete the following steps:

1. Edit the `/etc/dovecot/dovecot.conf` configuration file to make sure the `protocols` variable is uncommented (remove the hash sign (`#`) at the beginning of the line) and contains the `pop3` argument. For example:

   ```
   protocols = imap imaps pop3 pop3s
   ```

   When the `protocols` variable is left commented out, `dovecot` will use the default values specified for this variable.

2. Make that change operational for the current session by running the following command as root:

   ```
   systemctl restart dovecot.service
   ```

3. Make that change operational after the next reboot by running the command:

   ```
   systemctl enable dovecot.service
   ```
The dovecot service starts the POP3 server

Please note that dovecot only reports that it started the IMAP server, but also starts the POP3 server.

Unlike SMTP, both IMAP and POP3 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 /etc/pki/dovecot/dovecot-openssl.conf configuration file 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/libexec/dovecot/mkcert.sh script which creates the dovecot self signed certificates. These certificates are copied in the /etc/pki/dovecot/certs and /etc/pki/dovecot/private directories. To implement the changes, restart dovecot by typing the following at a shell prompt as root:

    systemctl restart dovecot.service

More details on dovecot can be found online at http://www.dovecot.org.

11.2. 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.

11.2.1. 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 offers two MTAs—Postfix and Sendmail—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 Postfix, Sendmail, and Fetchmail, refer to Section 11.3, “Mail Transport Agents”.

### 11.2.2. 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.

### 11.2.3. 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.

### 11.3. Mail Transport Agents

Fedora offers two primary MTAs: Postfix and Sendmail. Postfix is configured as the default MTA, although it is easy to switch the default MTA to Sendmail. To switch the default MTA to Sendmail, as root, you can either uninstall Postfix or use the following command to switch to Sendmail:

```
alternatives --config mta
```

You can also use the following command to enable/disable the desired service:

```
systemctl enable|disable service.service
```

### 11.3.1. 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 changed root 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.

### 11.3.1.1. 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.
- **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.

The **aliases** file can be found in the `/etc/` directory. This file is shared between Postfix and Sendmail. It is a configurable list required by the mail protocol that describes user ID aliases.

#### Configuring Postfix as a server for other clients

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 11.3.1.2, "Basic Postfix Configuration".

Restart the `postfix` service after changing any options in the configuration files under the `/etc/postfix/` directory in order for those changes to take effect. To do so, run the following command as root:

```bash
systemctl restart postfix.service
```

### 11.3.1.2. 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 sign (`#`), 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 the `/etc/postfix/main.cf` configuration file. Additional resources including information about Postfix configuration, SpamAssassin integration, or detailed descriptions of the `/etc/postfix/main.cf` parameters are available online at http://www.postfix.org/.

### 11.3.1.3. Using Postfix with LDAP

Postfix can use an LDAP directory as a source for various lookup tables (e.g.: `aliases`, `virtual`, `canonical`, etc.). This allows LDAP to store hierarchical user information and Postfix to only be given the result of LDAP queries when needed. By not storing this information locally, administrators can easily maintain it.

#### 11.3.1.3.1. The `/etc/aliases` lookup example

The following is a basic example for using LDAP to look up the `/etc/aliases` file. Make sure your `/etc/postfix/main.cf` contains the following:

```plaintext
alias_maps = hash:/etc/aliases, ldap:/etc/postfix/ldap-aliases.cf
```

Create a `/etc/postfix/ldap-aliases.cf` file if you do not have one created already and make sure it contains the following:

```plaintext
server_host = ldap.example.com
search_base = dc=example, dc=com
```

where `ldap.example.com`, `example`, and `com` are parameters that need to be replaced with specification of an existing available LDAP server.

#### The `/etc/postfix/ldap-aliases.cf` file

The `/etc/postfix/ldap-aliases.cf` file can specify various parameters, including parameters that enable LDAP SSL and STARTTLS. For more information, refer to the `ldap_table(5)` man page.

For more information on LDAP, refer to Section 12.1, “OpenLDAP”.

### 11.3.2. 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.

11.3.2.1. 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 11.6, “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).

11.3.2.2. The Default Sendmail Installation

In order to use Sendmail, first ensure the sendmail package is installed on your system by running, as root:

```
yum install sendmail
```

In order to configure Sendmail, ensure the sendmail-cf package is installed on your system by running, as root:

```
yum install sendmail-cf
```

For more information on installing packages with Yum, refer to Section 5.2.4, “Installing Packages”.

Before using Sendmail, the default MTA has to be switched from Postfix. For more information how to switch the default MTA refer to Section 11.3, “Mail Transport Agents”.

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 `systemctl start|restart|reload sendmail.service`.

- Alternatively you may use the `m4` macro processor to create a new `/etc/mail/sendmail.cf`. The `m4` macro processor is not installed by default. Before using it to create `/etc/mail/sendmail.cf`, install the `m4` package as root:

  
  ```
  yum install m4
  ```

  More information on configuring Sendmail can be found in Section 11.3.2.3, “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, as root:

  ```
  makemap hash /etc/mail/name < /etc/mail/name
  ```

  where `name` represents the name of the configuration file to be updated. You may also restart the sendmail service for the changes to take effect by running:

  ```
  systemctl restart sendmail.service
  ```

  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:

  ```
  makemap hash /etc/mail/virtusertable < /etc/mail/virtusertable
  ```

  Sendmail will create an updated `virtusertable.db` file containing the new configuration.

### 11.3.2.3. 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.
Backup the sendmail.cf file before changing its content

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 root. Once you are finished, restart the sendmail service and, if the m4 package is installed, the m4 macro processor will automatically generate a new sendmail.cf configuration file:

```
systemctl restart sendmail.service
```

Configuring Sendmail as a server for other clients

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 restarting the service:

```
systemctl restart sendmail.service
```

The default configuration which ships with Fedora works for most SMTP-only sites. However, it does not work for UUCP (UNIX-to-UNIX Copy Protocol) sites. If using UUCP mail transfers, the /etc/mail/sendmail.mc file must be reconfigured and a new /etc/mail/sendmail.cf file 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 the /etc/mail/sendmail.cf file.

11.3.2.4. 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
```
After generating a new `sendmail.cf` using the `m4` macro processor, this configuration makes all mail from inside the network appear as if it were sent from `bigcorp.com`.

11.3.2.5. 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:

```
systemctl restart sendmail.service
```

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:

```
badsheimer.com ERROR:550 "Go away and do not spam us" tux.badsheimer.com OK 10.0 RELAY
```

This example shows that any email sent from `badsheimer.com` is blocked with a 550 RFC-821 compliant error code, with a message sent back to the spammer. Email sent from the `tux.badsheimer.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 the `/etc/mail/access.db` file is a database, use the `makemap` command to update 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 email’s 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 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 11.4.2.6, “Spam Filters” for more information about using SpamAssassin.

11.3.2.6. Using Sendmail with LDAP

Using 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 /etc/aliases and /etc/mail/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')
FEATURE('ldap_routing')
```

Advanced configuration

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 the m4 macro processor and again restarting Sendmail. Refer to Section 11.3.2.3, “Common Sendmail Configuration Changes” for instructions.

For more information on LDAP, refer to Section 12.1, “OpenLDAP”.

11.3.3. 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.
Installing the fetchmail package

In order to use Fetchmail, first ensure the fetchmail package is installed on your system by running, as root:

```
yum install fetchmail
```

For more information on installing packages with Yum, refer to Section 5.2.4, “Installing Packages”.

Fetchmail is configured for each user through the use of a .fetchmailrc file in the user's home directory. If it does not already exist, create the .fetchmailrc file in your 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.

11.3.3.1. 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 the .fetchmailrc file.
because it only checks skipped servers when specifically invoked, and does not affect any currently working configurations.

The following is a sample example of a `.fetchmailrc` file:

```
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.

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 in the following three sections.

### 11.3.3.2. 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`.

### 11.3.3.3. 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.

### 11.3.3.4. 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.**

### 11.3.3.5. 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.

### 11.3.3.6. 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.

### 11.3.3.7. 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.

### 11.3.4. 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 20 provides two MTAs: Postfix and Sendmail. If both are installed, Postfix is the default MTA.
11.4. 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 (man mail).

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.

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.

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 a /etc/procmailrc file 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.

11.4.1. 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 the ~/.procmailrc file 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.
Chapter 11. Mail Servers

11.4.2. 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 11.4.2.5, "Recipe Examples".

Procmail recipes take the following form:

- **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
  ```

  To turn off Procmail filtering of email lists but leaving spam control in place, comment out the first INCLUDERC line with a hash sign (#).

- **LOCKSLEEP** — Sets the amount of time, in seconds, between attempts by Procmail to use a particular lockfile. The default is 8 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.
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 asterisk character (*) can further control the condition.

The action-to-perform argument 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 11.4.2.4, “Special Conditions and Actions” for more information.

11.4.2.1. 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.

11.4.2.2. 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 is the default behavior.

• 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.

11.4.2.3. 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.

11.4.2.4. 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 asterisk 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.

### 11.4.2.5. 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](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.
### Sending messages to `/dev/null`

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.

```plaintext
: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 11.6, “Additional Resources” for more detailed and powerful recipes.

### 11.4.2.6. 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.

### Installing the `spamassassin` package

In order to use SpamAssassin, first ensure the `spamassassin` package is installed on your system by running, as `root`:

```bash
yum install spamassassin
```

For more information on installing packages with Yum, refer to Section 5.2.4, “Installing Packages”.

The easiest way for a local user to use SpamAssassin is to place the following line near the top of the `~/.procmailrc` file:

```plaintext
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 the client application (`spamc`). Configuring SpamAssassin this way, however, requires root access to the host.

To start the `spamd` daemon, type the following command:

```
systemctl start spamassassin.service
```

To start the SpamAssassin daemon when the system is booted, run:

```
systemctl enable spamassassin.service
```

Refer to Chapter 7, Services and Daemons for more information on how to configure services in Fedora.

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
```

### 11.5. Mail User Agents

Fedora offers a variety of email programs, both, graphical email client programs, such as Evolution, and text-based email programs such as mutt.

The remainder of this section focuses on securing communication between a client and a server.

#### 11.5.1. Securing Communication

Popular MUAs included with Fedora, such as 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.

11.5.1.1. 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 the 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.

11.5.1.2. 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 in two ways: by applying to a Certificate Authority (CA) for an SSL certificate or by creating a self-signed certificate.

Avoid using self-signed certificates

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 or POP, change to the /etc/pki/dovecot/ directory, edit the certificate parameters in the /etc/pki/dovecot/dovecot-openssl.conf configuration file as you prefer, and type the following commands, as root:

dovecot]$ rm -f certs/dovecot.pem private/dovecot.pem
dovecot]$ /usr/libexec/dovecot/mkcert.sh

Once finished, make sure you have the following configurations in your /etc/dovecot/conf.d/10-ssl.conf file:

```
ssl_cert = </etc/pki/dovecot/certs/dovecot.pem
ssl_key = </etc/pki/dovecot/private/dovecot.pem
```

Execute the systemctl restart dovecot.service command to restart the dovecot daemon.

Alternatively, the stunnel command can be used as an SSL encryption wrapper around the standard, non-secure connections to IMAP or POP services.

The stunnel utility uses external OpenSSL libraries included with Fedora to provide strong cryptography and to protect the network connections. It is recommended to apply to a CA to obtain an SSL certificate, but it is also possible to create a self-signed certificate.
Installing the stunnel package

In order to use stunnel, first ensure the stunnel package is installed on your system by running, as root:

```
  yum install stunnel
```

For more information on installing packages with Yum, refer to Section 5.2.4, “Installing Packages”.

To create a self-signed SSL certificate, change to the `/etc/pki/tls/certs/` directory, and type the following command:

```
  certs]# make stunnel.pem
```

Answer all of the questions to complete the process.

Once the certificate is generated, create an stunnel configuration file, for example `/etc/stunnel/mail.conf`, with the following content:

```
cert = /etc/pki/tls/certs/stunnel.pem
[pop3s]
  accept = 995
  connect = 110

[imaps]
  accept = 993
  connect = 143
```

Once you start stunnel with the created configuration file using the `/usr/bin/stunnel /etc/stunnel/mail.conf` command, it will be possible to use an IMAP or a POP email client and connect to the email server using SSL encryption.

For more information on stunnel, refer to the stunnel man page or the documents in the `/usr/share/doc/stunnel-version-number/` directory, where `version-number` is the version number of stunnel.

11.6. Additional Resources

The following is a list of additional documentation about email applications.

11.6.1. Installed Documentation

- Information on configuring Sendmail is included with the sendmail and sendmail-cf packages.
- `/usr/share/sendmail-cf/README` — Contains information on the m4 macro processor, 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/proclmail-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.

### 11.6.2. Useful Websites


- `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://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.

### 11.6.3. 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 customize 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.
Directory Servers

12.1. OpenLDAP

LDAP (Lightweight Directory Access Protocol) 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”.

Like X.500, LDAP organizes information in a hierarchical manner using directories. These directories can store a variety of information such as names, addresses, or phone numbers, 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.

LDAP is commonly used for centrally managed users and groups, user authentication, or system configuration. It can also serve as a virtual phone directory, allowing users to easily access contact information for other users. Additionally, it can refer a user to other LDAP servers throughout the world, and thus provide an ad-hoc global repository of information. However, it is most frequently used within individual organizations such as universities, government departments, and private companies.

This section covers the installation and configuration of OpenLDAP 2.4, an open source implementation of the LDAPv2 and LDAPv3 protocols.

12.1.1. Introduction to LDAP

Using a client/server architecture, LDAP provides reliable means to create a central information directory accessible from the network. When a client attempts to modify information within this directory, the server verifies the user has permission to make the change, and then adds or updates the entry as requested. To ensure the communication is secure, the Secure Sockets Layer (SSL) or Transport Layer Security (TLS) cryptographic protocols can be used to prevent an attacker from intercepting the transmission.

Using Mozilla NSS

The OpenLDAP suite in Fedora 20 no longer uses OpenSSL. Instead, it uses the Mozilla implementation of Network Security Services (NSS). OpenLDAP continues to work with existing certificates, keys, and other TLS configuration. For more information on how to configure it to use Mozilla certificate and key database, refer to How do I use TLS/SSL with Mozilla NSS.

The LDAP server supports several database systems, which gives administrators the flexibility to choose the best suited solution for the type of information they are planning to serve. Because of a well-defined client Application Programming Interface (API), the number of applications able to communicate with an LDAP server is numerous, and increasing in both quantity and quality.

12.1.1.1. LDAP Terminology

The following is a list of LDAP-specific terms that are used within this chapter:

1 http://www.openldap.org/faq/index.cgi?file=1514
entry
A single unit within an LDAP directory. Each entry is identified by its unique Distinguished Name (DN).

attribute
Information directly associated with an entry. For example, if an organization is represented as an LDAP entry, attributes associated with this organization might include an address, a fax number, etc. Similarly, people can be represented as entries with common attributes such as personal telephone number or email address.

An attribute can either have a single value, or an unordered space-separated list of values. While certain attributes are optional, other are required. Required attributes are specified using the objectClass definition, and can be found in schema files located in the /etc/openldap/slapd.d/cn=config/cn=schema/ directory.

The assertion of an attribute and its corresponding value is also referred to as a Relative Distinguished Name (RDN). Unlike distinguished names that are unique globally, a relative distinguished name is only unique per entry.

LDIF
The LDAP Data Interchange Format (LDIF) is a plain text representation of an LDAP entry. It takes the following form:

```
[id] dn: distinguished_name
attribute_type: attribute_value...
attribute_type: attribute_value...
...
```

The optional id is a number determined by the application that is used to edit the entry. Each entry can contain as many attribute_type and attribute_value pairs as needed, as long as they are all defined in a corresponding schema file. A blank line indicates the end of an entry.

12.1.1.2. OpenLDAP Features

OpenLDAP suite provides a number of important features:

• **LDAPv3 Support** — Many of the changes in the protocol since LDAP version 2 are designed to make LDAP more secure. Among other improvements, this includes the support for Simple Authentication and Security Layer (SASL), Transport Layer Security (TLS), and Secure Sockets Layer (SSL) protocols.

• **LDAP Over IPC** — The use of inter-process communication (IPC) enhances security by eliminating the need to communicate over a network.

• **IPv6 Support** — OpenLDAP is compliant with Internet Protocol version 6 (IPv6), the next generation of the Internet Protocol.

• **LDIFv1 Support** — OpenLDAP is fully compliant with LDIF version 1.

• **Updated C API** — The current C API improves the way programmers can connect to and use LDAP directory servers.

• **Enhanced Standalone LDAP Server** — This includes an updated access control system, thread pooling, better tools, and much more.
12.1.1.3. OpenLDAP Server Setup

The typical steps to set up an LDAP server on Fedora are as follows:

1. Install the OpenLDAP suite. Refer to Section 12.1.2, “Installing the OpenLDAP Suite” for more information on required packages.

2. Customize the configuration as described in Section 12.1.3, “Configuring an OpenLDAP Server”.

3. Start the slapd service as described in Section 12.1.4, “Running an OpenLDAP Server”.

4. Use the ldapadd utility to add entries to the LDAP directory.

5. Use the ldapsearch utility to verify that the slapd service is accessing the information correctly.

12.1.2. Installing the OpenLDAP Suite

The suite of OpenLDAP libraries and tools is provided by the following packages:

Table 12.1. List of OpenLDAP packages

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>openldap</td>
<td>A package containing the libraries necessary to run the OpenLDAP server and client applications.</td>
</tr>
<tr>
<td>openldap-clients</td>
<td>A package containing the command line utilities for viewing and modifying directories on an LDAP server.</td>
</tr>
<tr>
<td>openldap-servers</td>
<td>A package containing both the services and utilities to configure and run an LDAP server. This includes the Standalone LDAP Daemon, slapd.</td>
</tr>
<tr>
<td>openldap-servers-sql</td>
<td>A package containing the SQL support module.</td>
</tr>
</tbody>
</table>

Additionally, the following packages are commonly used along with the LDAP server:

Table 12.2. List of commonly installed additional LDAP packages

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nss-pam-ldapd</td>
<td>A package containing ns1cd, a local LDAP name service that allows a user to perform local LDAP queries.</td>
</tr>
<tr>
<td>mod_authz_ldap</td>
<td>A package containing mod_authz_ldap, the LDAP authorization 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. Note that the mod_ssl module is required when using the mod_authz_ldap module.</td>
</tr>
</tbody>
</table>

To install these packages, use the yum command in the following form:

```
yum install package...
```
Chapter 12. Directory Servers

For example, to perform the basic LDAP server installation, type the following at a shell prompt as root:

```
yum install openldap openldap-clients openldap-servers
```

Note that you must have superuser privileges (that is, you must be logged in as root) to run this command. For more information on how to install new packages in Fedora, refer to Section 5.2.4, “Installing Packages”.

12.1.2.1. Overview of OpenLDAP Server Utilities

To perform administrative tasks, the openldap-servers package installs the following utilities along with the slapd service:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>slapacl</td>
<td>Allows you to check the access to a list of attributes.</td>
</tr>
<tr>
<td>slapadd</td>
<td>Allows you to add entries from an LDIF file to an LDAP directory.</td>
</tr>
<tr>
<td>slapauth</td>
<td>Allows you to check a list of IDs for authentication and authorization permissions.</td>
</tr>
<tr>
<td>slapcat</td>
<td>Allows you to pull entries from an LDAP directory in the default format and save them in an LDIF file.</td>
</tr>
<tr>
<td>slapdn</td>
<td>Allows you to check a list of Distinguished Names (DNs) based on available schema syntax.</td>
</tr>
<tr>
<td>slapindex</td>
<td>Allows you to re-index the slapd directory based on the current content. Run this utility whenever you change indexing options in the configuration file.</td>
</tr>
<tr>
<td>slappasswd</td>
<td>Allows you to create an encrypted user password to be used with the ldapmodify utility, or in the slapd configuration file.</td>
</tr>
<tr>
<td>slapschema</td>
<td>Allows you to check the compliance of a database with the corresponding schema.</td>
</tr>
<tr>
<td>slaptest</td>
<td>Allows you to check the LDAP server configuration.</td>
</tr>
</tbody>
</table>

For a detailed description of these utilities and their usage, refer to the corresponding manual pages as referred to in Section 12.1.6.1, “Installed Documentation”.

Make sure the files have correct owner

Although only root can run slapadd, the slapd service runs as the ldap user. Because of this, the directory server is unable to modify any files created by slapadd. To correct this issue, after running the slapadd utility, type the following at a shell prompt:

```
chown -R ldap:ldap /var/lib/ldap
```
To preserve the data integrity, stop the \texttt{slapd} service before using \texttt{slapadd}, \texttt{slapcat}, or \texttt{slapindex}. You can do so by typing the following at a shell prompt as root:

\begin{verbatim}
    systemctl stop slapd.service
\end{verbatim}

For more information on how to start, stop, restart, and check the current status of the \texttt{slapd} service, refer to \autoref{sec:running-ldap}, “Running an OpenLDAP Server”.

### 12.1.2.2. Overview of OpenLDAP Client Utilities

The \texttt{openldap-clients} package installs the following utilities which can be used to add, modify, and delete entries in an LDAP directory:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{ldapadd}</td>
<td>Allows you to add entries to an LDAP directory, either from a file, or from standard input. It is a symbolic link to \texttt{ldapmodify -a}.</td>
</tr>
<tr>
<td>\texttt{ldapcompare}</td>
<td>Allows you to compare given attribute with an LDAP directory entry.</td>
</tr>
<tr>
<td>\texttt{ldapdelete}</td>
<td>Allows you to delete entries from an LDAP directory.</td>
</tr>
<tr>
<td>\texttt{ldapexop}</td>
<td>Allows you to perform extended LDAP operations.</td>
</tr>
<tr>
<td>\texttt{ldapmodify}</td>
<td>Allows you to modify entries in an LDAP directory, either from a file, or from standard input.</td>
</tr>
<tr>
<td>\texttt{ldapmodrdn}</td>
<td>Allows you to modify the RDN value of an LDAP directory entry.</td>
</tr>
<tr>
<td>\texttt{ldappasswd}</td>
<td>Allows you to set or change the password for an LDAP user.</td>
</tr>
<tr>
<td>\texttt{ldapsearch}</td>
<td>Allows you to search LDAP directory entries.</td>
</tr>
<tr>
<td>\texttt{ldapurl}</td>
<td>Allows you to compose or decompose LDAP URLs.</td>
</tr>
<tr>
<td>\texttt{ldapwhoami}</td>
<td>Allows you to perform a \texttt{whoami} operation on an LDAP server.</td>
</tr>
</tbody>
</table>

With the exception of \texttt{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.

### 12.1.2.3. Overview of Common LDAP Client Applications

Although there are various graphical LDAP clients capable of creating and modifying directories on the server, none of them is included in Fedora. Popular applications that can access directories in a read-only mode include Mozilla Thunderbird, Evolution, or Ekiga.

### 12.1.3. Configuring an OpenLDAP Server

By default, the OpenLDAP configuration is stored in the \texttt{/etc/openldap/} directory. The following table highlights the most important directories and files within this directory:
Table 12.5. List of OpenLDAP configuration files and directories

<table>
<thead>
<tr>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc/openldap/ldap.conf</td>
<td>The configuration file for client applications that use the OpenLDAP libraries. This includes ldapadd, ldapsearch, Evolution, etc.</td>
</tr>
<tr>
<td>/etc/openldap/slapd.d/</td>
<td>The directory containing the slapd configuration.</td>
</tr>
</tbody>
</table>

Note that OpenLDAP no longer reads its configuration from the /etc/openldap/slapd.conf file. Instead, it uses a configuration database located in the /etc/openldap/slapd.d/ directory. If you have an existing slapd.conf file from a previous installation, you can convert it to the new format by running the following command as root:

```
slaptest -f /etc/openldap/slapd.conf -F /etc/openldap/slapd.d/
```

The slapd configuration consists of LDIF entries organized in a hierarchical directory structure, and the recommended way to edit these entries is to use the server utilities described in Section 12.2.1, “Overview of OpenLDAP Server Utilities”.

Do not edit LDIF files directly

An error in an LDIF file can render the slapd service unable to start. Because of this, it is strongly advised that you avoid editing the LDIF files within the /etc/openldap/slapd.d/ directly.

12.1.3.1. Changing the Global Configuration

Global configuration options for the LDAP server are stored in the /etc/openldap/slapd.d/cn=config.ldif file. The following directives are commonly used:

**olcAllows**

The `olcAllows` directive allows you to specify which features to enable. It takes the following form:

```
olcAllows: feature...
```

It accepts a space-separated list of features as described in Table 12.6, “Available olcAllows options”. The default option is bind_v2.

Table 12.6. Available olcAllows options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bind_v2</td>
<td>Enables the acceptance of LDAP version 2 bind requests.</td>
</tr>
<tr>
<td>bind_anon_cred</td>
<td>Enables an anonymous bind when the Distinguished Name (DN) is empty.</td>
</tr>
<tr>
<td>bind_anon_dn</td>
<td>Enables an anonymous bind when the Distinguished Name (DN) is not empty.</td>
</tr>
<tr>
<td>update_anon</td>
<td>Enables processing of anonymous update operations.</td>
</tr>
</tbody>
</table>
### proxy_authz_anon

Enables processing of anonymous proxy authorization control.

#### Example 12.1. Using the olcAllows directive

```
olcAllows: bind_v2 update_anon
```

### olcConnMaxPending

The `olcConnMaxPending` directive allows you to specify the maximum number of pending requests for an anonymous session. It takes the following form:

```
olcConnMaxPending: number
```

The default option is **100**.

#### Example 12.2. Using the olcConnMaxPending directive

```
olcConnMaxPending: 100
```

### olcConnMaxPendingAuth

The `olcConnMaxPendingAuth` directive allows you to specify the maximum number of pending requests for an authenticated session. It takes the following form:

```
olcConnMaxPendingAuth: number
```

The default option is **1000**.

#### Example 12.3. Using the olcConnMaxPendingAuth directive

```
olcConnMaxPendingAuth: 1000
```

### olcDisallows

The `olcDisallows` directive allows you to specify which features to disable. It takes the following form:

```
olcDisallows: feature...
```

It accepts a space-separated list of features as described in Table 12.7, "Available olcDisallows options". No features are disabled by default.

#### Table 12.7. Available olcDisallows options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bind_anon</td>
<td>Disables the acceptance of anonymous bind requests.</td>
</tr>
<tr>
<td>bind_simple</td>
<td>Disables the simple bind authentication mechanism.</td>
</tr>
<tr>
<td>tls_2_anon</td>
<td>Disables the enforcing of an anonymous session when the STARTTLS command is received.</td>
</tr>
<tr>
<td>tls_authc</td>
<td>Disallows the STARTTLS command when authenticated.</td>
</tr>
</tbody>
</table>
Example 12.4. Using the olcDisallows directive

```
olcDisallows: bind_anon
```

**olcIdleTimeout**

The **olcIdleTimeout** directive allows you to specify how many seconds to wait before closing an idle connection. It takes the following form:

```
olcIdleTimeout: number
```

This option is disabled by default (that is, set to 0).

Example 12.5. Using the olcIdleTimeout directive

```
olcIdleTimeout: 180
```

**olcLogFile**

The **olcLogFile** directive allows you to specify a file in which to write log messages. It takes the following form:

```
olcLogFile: file_name
```

The log messages are written to standard error by default.

Example 12.6. Using the olcLogFile directive

```
olcLogFile: /var/log/slapd.log
```

**olcReferral**

The **olcReferral** option allows you to specify a URL of a server to process the request in case the server is not able to handle it. It takes the following form:

```
olcReferral: URL
```

This option is disabled by default.

Example 12.7. Using the olcReferral directive

```
olcReferral: ldap://root.openldap.org
```

**olcWriteTimeout**

The **olcWriteTimeout** option allows you to specify how many seconds to wait before closing a connection with an outstanding write request. It takes the following form:

```
olcWriteTimeout
```

This option is disabled by default (that is, set to 0).
12.1.3.2. Changing the Database-Specific Configuration

By default, the OpenLDAP server uses Berkeley DB (BDB) as a database back end. The configuration for this database is stored in the `/etc/openldap/slapd.d/cn=config/olcDatabase={1}bdb.ldif` file. The following directives are commonly used in a database-specific configuration:

**olcReadOnly**

The `olcReadOnly` directive allows you to use the database in a read-only mode. It takes the following form:

```
olcReadOnly: boolean
```

It accepts either `TRUE` (enable the read-only mode), or `FALSE` (enable modifications of the database). The default option is `FALSE`.

**Example 12.9. Using the olcReadOnly directive**

```
olcReadOnly: TRUE
```

**olcRootDN**

The `olcRootDN` directive allows you to specify the user that is unrestricted by access controls or administrative limit parameters set for operations on the LDAP directory. It takes the following form:

```
olcRootDN: distinguished_name
```

It accepts a *Distinguished Name* (DN). The default option is `cn=Manager,dn=my-domain,dc=com`.

**Example 12.10. Using the olcRootDN directive**

```
olcRootDN: cn=root,dn=example,dc=com
```

**olcRootPW**

The `olcRootPW` directive allows you to set a password for the user that is specified using the `olcRootDN` directive. It takes the following form:

```
olcRootPW: password
```

It accepts either a plain text string, or a hash. To generate a hash, use the `slappaswd` utility, for example:

```
~:]$ slappaswd
New password:
```
Chapter 12. Directory Servers

Re-enter new password: {SSHA}WczWsyPEnMchFf1GRTweq2q7XJcvmSxD

Example 12.11. Using the `olcRootPW` directive

```
olcRootPW: {SSHA}WczWsyPEnMchFf1GRTweq2q7XJcvmSxD
```

`olcSuffix`

The `olcSuffix` directive allows you to specify the domain for which to provide information. It takes the following form:

```
olcSuffix: domain_name
```

It accepts a fully qualified domain name (FQDN). The default option is `dc=my-domain,dc=com`.

Example 12.12. Using the `olcSuffix` directive

```
olcSuffix: dc=example,dc=com
```

12.1.3.3. Extending Schema

Since OpenLDAP 2.3, the `/etc/openldap/slapd.d/` directory also contains LDAP definitions that were previously located in `/etc/openldap/schema/`. 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. However, this task is beyond the scope of this chapter. For more information on this topic, refer to [http://www.openldap.org/doc/admin/schema.html](http://www.openldap.org/doc/admin/schema.html).

12.1.4. Running an OpenLDAP Server

This section describes how to start, stop, restart, and check the current status of the Standalone LDAP Daemon. For more information on how to manage system services in general, refer to Chapter 7, Services and Daemons.

12.1.4.1. Starting the Service

To run the `slapd` service, type the following at a shell prompt as root:

```
systemctl start slapd.service
```

If you want the service to start automatically at the boot time, use the following command:

```
systemctl enable slapd.service
```

Refer to Chapter 7, Services and Daemons for more information on how to configure services in Fedora.

12.1.4.2. Stopping the Service
To stop the running `slapd` service, type the following at a shell prompt as `root`:

```
systemctl stop slapd.service
```

To prevent the service from starting automatically at the boot time, type:

```
systemctl disable slapd.service
```

Refer to Chapter 7, Services and Daemons for more information on how to configure services in Fedora.

### 12.1.4.3. Restarting the Service

To restart the running `slapd` service, type the following at a shell prompt as `root`:

```
systemctl restart slapd.service
```

This stops the service, and then starts it again. Use this command to reload the configuration.

### 12.1.4.4. Checking the Service Status

To check whether the service is running, type the following at a shell prompt:

```
systemctl is-active slapd.service
```

### 12.1.5. Configuring a System to Authenticate Using OpenLDAP

In order to configure a system to authenticate using OpenLDAP, make sure that the appropriate packages are installed on both LDAP server and client machines. For information on how to set up the server, follow the instructions in Section 12.1.2, “Installing the OpenLDAP Suite” and Section 12.1.3, “Configuring an OpenLDAP Server”. On a client, type the following at a shell prompt as `root`:

```
yum install openldap openldap-clients nss-pam-ldapd
```

Chapter 8, Configuring Authentication provides detailed instructions on how to configure applications to use LDAP for authentication.

#### 12.1.5.1. Migrating Old Authentication Information to LDAP Format

The `migrationtools` package provides a set of shell and Perl scripts to help you migrate authentication information into an LDAP format. To install this package, type the following at a shell prompt as `root`:

```
yum install migrationtools
```

This will install the scripts to the `/usr/share/migrationtools/` directory. Once installed, edit the `/usr/share/migrationtools/migrate_common.ph` file and change the following lines to reflect the correct domain, for example:

```bash
# Default DNS domain
```
$DEFAULT_MAIL_DOMAIN = "example.com";

#define $DEFAULT_BASE = "dc=example,dc=com"

Alternatively, you can specify the environment variables directly on the command line. For example, to run the migrate_all_online.sh script with the default base set to dc=example,dc=com, type:

export DEFAULT_BASE="dc=example,dc=com" \
/usr/share/migrationtools/migrate_all_online.sh

To decide which script to run in order to migrate the user database, refer to Table 12.8, “Commonly used LDAP migration scripts”.

Table 12.8. Commonly used LDAP migration scripts

<table>
<thead>
<tr>
<th>Existing Name Service</th>
<th>Is LDAP Running?</th>
<th>Script to Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc flat files</td>
<td>yes</td>
<td>migrate_all_online.sh</td>
</tr>
<tr>
<td>/etc flat files</td>
<td>no</td>
<td>migrate_all_offline.sh</td>
</tr>
<tr>
<td>NetInfo</td>
<td>yes</td>
<td>migrate_all_netinfo_online.sh</td>
</tr>
<tr>
<td>NetInfo</td>
<td>no</td>
<td>migrate_all_netinfo_offline.sh</td>
</tr>
<tr>
<td>NIS (YP)</td>
<td>yes</td>
<td>migrate_all_nis_online.sh</td>
</tr>
<tr>
<td>NIS (YP)</td>
<td>no</td>
<td>migrate_all_nis_offline.sh</td>
</tr>
</tbody>
</table>

For more information on how to use these scripts, refer to the README and the migration-tools.txt files in the /usr/share/doc/migrationtools-version/ directory.

12.1.6. Additional Resources

The following resources offer additional information on the Lightweight Directory Access Protocol. Before configuring LDAP on your system, it is highly recommended that you review these resources, especially the OpenLDAP Software Administrator’s Guide.

12.1.6.1. Installed Documentation

The following documentation is installed with the openldap-servers package:

/us/share/doc/openldap-servers-version/guide.html
A copy of the OpenLDAP Software Administrator’s Guide.

/us/share/doc/openldap-servers-version/README.schema
A README file containing the description of installed schema files.

Additionally, there is also a number of manual pages that are installed with the openldap, openldap-servers, and openldap-clients packages:

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.

**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-config** — Describes the format and options available within the configuration directory.

**12.1.6.2. Useful Websites**

http://www.openldap.org/doc/admin24/
The current version of the *OpenLDAP Software Administrator’s Guide*.

http://www.kingsmountain.com/ldapRoadmap.shtml
Jeff Hodges’ *LDAP Roadmap & FAQ* containing links to several useful resources and emerging news concerning the LDAP protocol.

http://www.ldapman.org/articles/
A collection of articles that offer a good introduction to LDAP, including methods to design a directory tree and customizing directory structures.

http://www.padl.com/
A website of developers of several useful LDAP tools.

**12.1.6.3. Related Books**

*OpenLDAP by Example* by John Terpstra and Benjamin Coles; Prentice Hall.
A collection of practical exercises in the OpenLDAP deployment.

*Implementing LDAP* by Mark Wilcox; Wrox Press, Inc.
A book covering LDAP from both the system administrator’s and software developer’s perspective.

A book covering LDAP design principles, as well as its deployment in a production environment.
File and Print Servers

This chapter guides you through the installation and configuration of Samba, an open source implementation of the Server Message Block (SMB) protocol, and vsftpd, the primary FTP server shipped with Fedora. Additionally, it explains how to use the Printer Configuration tool to configure printers.

13.1. 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.

Installing the samba package

In order to use Samba, first ensure the samba package is installed on your system by running, as root:

```
yum install samba
```

For more information on installing packages with Yum, refer to Section 5.2.4, "Installing Packages".

13.1.1. 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 the Lightweight Directory Access Protocol (LDAP) and Kerberos
- Built in Unicode support for internationalization
- Support for all recent Microsoft Windows server and client versions to connect 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 13.1.12.2, "Related Books".

13.1.1.1. 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/2000/2003/2008 PDC

What Samba cannot do:
• Act as a BDC for a Windows PDC (and vice versa)
• Act as an Active Directory domain controller

### 13.1.2. Samba Daemons and Related Services

The following is a brief introduction to the individual Samba daemons and services.

#### 13.1.2.1. Samba Daemons

Samba is comprised of three daemons (**smbd**, **nmbd**, and **winbindd**). Three services (**smb**, **nmb**, and **winbind***) control how the daemons are started, stopped, and other service-related features. These services act as different init scripts. Each daemon is listed in detail below, 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/Common Internet File System (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 **nmb** service.

**winbindd**

The **winbind** service resolves user and group information on a server running Windows NT, 2000, 2003 or Windows Server 2008. 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 chapter.

### Obtaining a list of utilities that are shipped with Samba

You may refer to Section 13.1.11, “Samba Distribution Programs” for a list of utilities included in the Samba distribution.

### 13.1.3. Connecting to a Samba Share

You can use *Nautilus* to view available Samba shares on your network. To view a list of Samba workgroups and domains on your network, select Applications → Accessories → Files from the Activities menu, and click Browse Network at the sidebar.

An icon appears for each available SMB workgroup or domain on the network. Double-click one of the workgroup/domain icons to view a list of computers within the workgroup/domain.
Each machine within the workgroup is represented by its own icon. 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
```

### 13.1.3.1. 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.

### 13.1.3.2. 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 a directory to mount it to (if it does not already exist),
and execute the following command as root:

```
mount -t cifs //servername/sharename /mnt/point/ -o username=username,password=password
```

This command mounts sharename from servername in the local directory /mnt/point/.
Installing cifs-utils package

The `mount.cifs` utility is a separate RPM (independent from Samba). In order to use `mount.cifs`, first ensure the `cifs-utils` package is installed on your system by running, as root:

```
yum install cifs-utils
```

For more information on installing packages with Yum, refer to Section 5.2.4, "Installing Packages".

Note that the `cifs-utils` package also contains the `cifs.upcall` binary called by the kernel in order to perform kerberized CIFS mounts. For more information on `cifs.upcall`, refer to `man cifs.upcall`.

For more information about mounting a samba share, refer to `man mount.cifs`.

CIFS servers that require plain text passwords

Some CIFS servers require plain text passwords for authentication. Support for plain text password authentication can be enabled using the following command as root:

```
echo 0x37 > /proc/fs/cifs/SecurityFlags
```

WARNING: This operation can expose passwords by removing password encryption.

13.1.4. 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.

13.1.4.1. Graphical Configuration

To configure Samba using a graphical interface, use one of the available Samba graphical user interfaces. A list of available GUIs can be found at [http://www.samba.org/samba/GUI/](http://www.samba.org/samba/GUI/).

13.1.4.2. 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 following command, as root:

```
systemctl restart smb.service
```
To specify the Windows workgroup and a brief description of the Samba server, edit the following lines in your `/etc/samba/smb.conf` file:

```plaintext
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 `/etc/samba/smb.conf` file (after modifying it to reflect your needs and your system):

```plaintext
[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.

### 13.1.4.3. 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`.

### 13.1.5. Starting and Stopping Samba

To start a Samba server, type the following command in a shell prompt, as `root`:

```
systemctl start smb.service
```

#### Setting up a domain member server

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, as `root`:

```
systemctl stop smb.service
```

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, as `root`:
systemctl restart smb.service

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.

**Applying the changes to the configuration**

When the `/etc/samba/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`:

```
systemctl condrestart smb.service
```

A manual reload of the `/etc/samba/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`:

```
systemctl reload smb.service
```

By default, the **smb** service does **not** start automatically at boot time. To configure Samba to start at boot time, use a service manager such as **systemctl**. Refer to **Chapter 7, Services and Daemons** for more information regarding this tool.

### 13.1.6. 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 `/etc/samba/smb.conf` file for a successful configuration.

#### 13.1.6.1. 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 13.1.7, “Samba Security Modes”**.

#### 13.1.6.1.1. Anonymous Read-Only

The following `/etc/samba/smb.conf` file shows a sample configuration needed to implement anonymous read-only file sharing. The `security = share` parameter makes a share anonymous.
Chapter 13. File and Print Servers

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 /etc/samba/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
```

### 13.1.6.1.2. Anonymous Read/Write

The following /etc/samba/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.

**Do not use anonymous read/write servers**

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 /etc/samba/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
```

### 13.1.6.1.3. Anonymous Print Server

The following /etc/samba/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
```
Samba Server Types and the `smb.conf` File

```
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
```

13.1.6.1.4. Secure Read/Write File and Print Server

The following `/etc/samba/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
```

13.1.6.2. 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.
13.1.6.2.1. Active Directory Domain Member Server

The following `/etc/samba/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 2000/2003/2008 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 `/etc/samba/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 2000/2003/2008 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.

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.
The security option

Since `security = ads` and not `security = user` is used, a local password back end 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.

13.1.6.2.2. Windows NT4-based Domain Member Server

The following `/etc/samba/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 `/etc/samba/smb.conf` file simpler. In this instance, the Samba member server functions as a pass through to the NT4-based domain server.

```
[globals]
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 `/etc/samba/smb.conf` file to convert the server to a Samba-based PDC. If Windows NT-based servers are upgraded to Windows 2000/2003/2008, the `/etc/samba/smb.conf` file is easily modifiable to incorporate the infrastructure change to Active Directory if needed.

Make sure you join the domain before starting Samba

After configuring the `/etc/samba/smb.conf` file, join the domain `before` starting Samba by typing the following command as root:

```
net rpc join -U administrator%password
```
Chapter 13. File and Print Servers

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 `/etc/samba/smb.conf` file instead of it being stated explicitly.

13.1.6.3. 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.

A mixed Samba/Windows domain controller environment

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.

13.1.6.3.1. Primary Domain Controller (PDC) using `tdbsam`

The simplest and most common implementation of a Samba PDC uses the new default `tdbsam` password database back end. Replacing the aging `smbpasswd` back end, `tdbsam` has numerous improvements that are explained in more detail in Section 13.1.8, “Samba Account Information Databases”. The `passdb backend` directive controls which back end is to be used for the PDC.

The following `/etc/samba/smb.conf` file shows a sample configuration needed to implement a `tdbsam` password database back end.

```
[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
    # This sets the default profile path.
    logon drive = H:
    domain logons = Yes
    os level = 35
    preferred master = Yes
    domain master = Yes
[homes]
    comment = Home Directories
```
To provide a functional PDC system which uses the *tdbsam* follow these steps:

1. Use a configuration of the *smb.conf* file as shown in the example above.

2. Add the root user to the Samba password database:

   ```
smbpasswd -a root
   ```

3. Start the *smb* service.

4. Make sure all profile, user, and netlogon directories are created.

5. Add groups that users can be members of:

   ```
groupadd -f users
   groupadd -f nobody
   groupadd -f ntadmins
   ```

6. Associate the UNIX groups with their respective Windows groups:

   ```
   net groupmap add ntgroup="Domain Users" unixgroup=users
   net groupmap add ntgroup="Domain Guests" unixgroup=nobody
   net groupmap add ntgroup="Domain Admins" unixgroup=ntadmins
   ```

7. Grant access rights to a user or a group. For example, to grant the right to add client machines to the domain on a Samba domain controller, to the members to the Domain Admins group, execute the following command:

   ```
   net rpc rights grant 'DOCS\Domain Admins' SetMachineAccountPrivilege -S PDC -U root
   ```

Keep in mind that Windows systems prefer to have a primary group which is mapped to a domain group such as Domain Users.

Windows groups and users use the same namespace thus not allowing the existence of a group and a user with the same name like in UNIX.
Limitations of the tdbsam authentication back end

If you need more than one domain controller or have more than 250 users, do not use a tdbsam authentication back end. LDAP is recommended in these cases.

13.1.6.3.2. 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.

13.1.7. 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.

13.1.7.1. User-Level Security

User-level security is the default setting for Samba. Even if the security = user directive is not listed in the /etc/samba/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 the /etc/samba/smb.conf file, the security = user directive that sets user-level security is:

```
[GLOBAL]
...
security = user
...
```

The following sections describe other implementations of user-level security.

13.1.7.1.1. 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 the /etc/samba/smb.conf file:

```
[GLOBAL]
...
security = domain
workgroup = MARKETING
...
```
13.1.7.1.2. 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 the `/etc/samba/smb.conf` file, the following directives make Samba an Active Directory member server:

```plaintext
[GLOBAL]
...
security = ADS
realm = EXAMPLE.COM
password server = kerberos.example.com
...
```

13.1.7.1.3. Server Security Mode (User-Level Security)

Server security mode was previously used when Samba was not capable of acting as a domain member server.

Avoid using the server security mode

It is highly recommended to not use this mode since there are numerous security drawbacks.

In the `/etc/samba/smb.conf`, the following directives enable Samba to operate in server security mode:

```plaintext
[GLOBAL]
...
encrypt passwords = Yes
security = server
password server = "NetBIOS_of_Domain_Controller"
...
```

13.1.7.2. 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 the `/etc/samba/smb.conf` file, the `security = share` directive that sets share-level security is:

```plaintext
[GLOBAL]
...
security = share
```
13.8. Samba Account Information Databases

The latest release of Samba offers many new features including new password database back ends not previously available. Samba version 3.0.0 fully supports all databases used in previous versions of Samba. However, although supported, many back ends may not be suitable for production use.

The following is a list different back ends you can use with Samba. Other back ends not listed here may also be available.

Plain Text
Plain text back ends are nothing more than the `/etc/passwd` type back ends. With a plain text back end, 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 back end used in previous Samba packages, the `smbpasswd` back end utilizes a plain ASCII text layout that includes the MS Windows LanMan and NT account, and encrypted password information. The `smbpasswd` back end lacks the storage of the Windows NT/2000/2003 SAM extended controls. The `smbpasswd` back end is not recommended because it does not scale well or hold any Windows information, such as RIDs for NT-based groups. The `tdbsam` back end solves these issues for use in a smaller database (250 users), but is still not an enterprise-class solution.

`ldapsam_compat`
The `ldapsam_compat` back end allows continued OpenLDAP support for use with upgraded versions of Samba. This option is normally used when migrating to Samba 3.0.

`tdbsam`
The new default `tdbsam` password back end provides an ideal database back end 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` back end 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/2000/2003/2008-based systems.

The `tdbsam` back end 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` back end 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 such as an OpenLDAP Server. LDAP databases are light-weight and scalable, and as such are preferred by large enterprises. For more information on LDAP, refer to Section 12.1, “OpenLDAP”.

If you are upgrading from a previous version of Samba to 3.0, note that the OpenLDAP schema file (`/usr/share/doc/samba-version/LDAP/samba.schema`) has changed. These files contain the attribute syntax definitions and objectclass definitions that the `ldapsam` back end needs in order to function properly.
As such, if you are using the ldapsam back end for your Samba server, you will need to configure slapd to include one of these schema file. Refer to Section 12.1.3.3, “Extending Schema” for directions on how to do this.

Make sure the openldap-server package is installed

You need to have the openldap-server package installed if you want to use the ldapsam back end.

13.1.9. 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.

13.1.9.1. 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 /etc/samba/smb.conf file for a local master browser (or no browsing at all) in a domain controller environment is the same as workgroup configuration.

13.1.9.2. WINS (Windows Internet 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/2000/2003/2008 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 /etc/samba/smb.conf file in which the Samba server is serving as a WINS server:

```
[global]
wins support = Yes
```
Using WINS

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.

13.1.10. 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.

13.1.10.1. Simple smb.conf Settings

The following example shows a very basic /etc/samba/smb.conf configuration for CUPS support:

```
[global]
load printers = Yes
printing = cups
printcap name = cups
[printers]
comment = All Printers
path = /var/spool/samba
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$ directive contains printer drivers for clients to access if not available locally. The print$ directive 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.

13.1.11. Samba Distribution Programs

findsmb
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 3.0.8-15]
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 3.0.6-14.3E]
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 NT4/2000/2003/2008 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 back ends are supported including `smbpasswd`, LDAP, 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

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

The `smbcacls` program modifies Windows ACLs on files and directories shared by a Samba server or a Windows server.

smbclient

The `smbclient` program is a versatile UNIX client which provides functionality similar to `ftp`.

smbcontrol

The `smbcontrol` program sends control messages to running `smbd`, `nmbd`, or `winbindd` daemons. Executing `smbcontrol -i` runs commands interactively until a blank line or a 'q' is entered.

smbpasswd
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.

### smbpasswd

```
smbpasswd options username password
```

The **smbpasswd** program is a CUPS-compatible printing interface to Samba. Although designed for use with CUPS printers, **smbpool** 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 `/etc/samba/smb.conf` file. If your `/etc/samba/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 `/etc/samba/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
    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.

13.1.12. Additional Resources

The following sections give you the means to explore Samba in greater detail.

13.1.12.1. 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.
Make sure you have the samba-doc package installed

In order to use the Samba documentation, first ensure the samba-doc package is installed on your system by running, as root:

```
yum install samba-doc
```

For more information on installing packages with Yum, refer to Section 5.2.4, “Installing Packages”.

Refer to the following manual pages for detailed information specific Samba features:

- `smb.conf`
- `samba`
- `smbd`
- `nmbd`
- `winbind`

### 13.1.12.2. 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.

### 13.1.12.3. Useful Websites

- [http://www.samba.org/](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](http://us1.samba.org/samba/archives.html) — Active email lists for the Samba community. Enabling digest mode is recommended due to high levels of list activity.

---

1 [http://us1.samba.org/samba/archives.html](http://us1.samba.org/samba/archives.html)
• 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.

13.2. 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 section outlines the basics of the FTP protocol, as well as configuration options for the primary FTP server shipped with Fedora, **vsftpd**.

13.2.1. 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.

13.2.1.1. 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 13.2.5.8, “Network Options”* for more information about limiting passive ports.
13.2.2. FTP Servers

Fedora ships with two different FTP servers:

- **proftpd** - A fast, stable, and highly configurable FTP server.
- **vsftpd** — A fast, secure FTP daemon which is the preferred FTP server for Fedora. The remainder of this section focuses on **vsftpd**.

13.2.2.1. 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.

13.2.3. 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 systemctl command to start, stop, or reload vsftpd. Refer to Section 13.2.4, “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 on PAM, refer to the Using Pluggable Authentication Modules (PAM) chapter of the Fedora 20 Managing Single Sign-On and Smart Cards guide.

• /etc/vsftpd/vsftpd.conf — The configuration file for vsftpd. Refer to Section 13.2.5, “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.

### 13.2.4. Starting and Stopping vsftpd

The vsftpd RPM installs the /etc/rc.d/init.d/vsftpd script, which can be accessed using the systemctl command.

To start the server, as root type:

```bash
systemctl start vsftpd.service
```

To stop the server, as root type:

```bash
systemctl stop vsftpd.service
```

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:

```bash
systemctl restart vsftpd.service
```

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:

```bash
systemctl condrestart vsftpd.service
```

By default, the vsftpd service does not start automatically at boot time. To configure the vsftpd service to start at boot time, use a service manager such as systemctl. Refer to Chapter 7, Services and Daemons for more information on how to configure services in Fedora.
13.2.4.1. 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. For more information about configuring network devices, device aliases, and additional information about network configuration scripts, refer to the Red Hat Enterprise Linux 7 Networking Guide.

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 the Red Hat Enterprise Linux 7 Networking Guide.

If there is more configuration files present in the /etc/vsftpd directory, calling systemctl start vsftpd.service results in the /etc/rc.d/init.d/vsftpd initscript starting the same number of processes as the number of configuration files. Each configuration file must have a unique name in the /etc/vsftpd/ directory and must be readable and writable only by root.

13.2.5. 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.

Do not use spaces

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 sign (#) and are ignored by the daemon.

For a complete list of all directives available, refer to the man page for vsftpd.conf.

Securing the vsftpd service

For an overview of ways to secure vsftpd, refer to the Fedora 20 Security Guide.
The following is a list of some of the more important directives within /etc/vsftpd/vsftpd.conf. All directives not explicitly found or commented out within vsftpd's configuration file are set to their default value.

### 13.2.5.1. 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). For more information, refer to the Using Pluggable Authentication Modules (PAM) chapter of the Red Hat Enterprise Linux 6 Managing Single Sign-On and Smart Cards and the PAM man pages. If session logging is not necessary, disabling this option allows vsftpd to run with less processes and lower privileges.

  The default value is YES.

### 13.2.5.2. 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 13.2.5.3, “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 ftphd_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.
Chapter 13. File and Print Servers

13.2.5. 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**.

### 13.2.5.4. 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**.
Avoid enabling the chroot_local_user option

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.

### 13.2.5.5. 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, text 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**.

### 13.2.5.6. 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**.

### 13.2.5.7. 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**.

---

**Maintaining compatibility with older log file formats**

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 sign (#), or delete the line entirely.
13.2.5.8. 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.

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 13.2.4.1, “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.

### Running multiple copies of vsftpd

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 13.2.4.1, “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.
Avoid enabling the pasv_promiscuous option

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.

### 13.2.6. Additional Resources

For more information about **vsftpd**, refer to the following resources.

#### 13.2.6.1. 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`.

#### 13.2.6.2. Useful Websites

- [http://vsftpd.beasts.org/](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](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](http://www.ietf.org/rfc/rfc0959.txt) — The original Request for Comments (RFC) of the FTP protocol from the IETF.
13.3. Printer Configuration

The **Printer Configuration** tool serves for printer configuring, maintenance of printer configuration files, print spool directories and print filters, and printer classes management.

The tool is based on the Common Unix Printing System (CUPS). If you upgraded the system from a previous Fedora version that used CUPS, the upgrade process preserved the configured printers.

Using the CUPS web application or command line tools

You can perform the same and additional operations on printers directly from the CUPS web application or command line. To access the application, in a web browser, go to `http://localhost:631/`. For CUPS manuals refer to the links on the **Home** tab of the web site.

13.3.1. Starting the Printer Configuration Tool

With the Printer Configuration tool you can perform various operations on existing printers and set up new printers.

On the upper panel, go to **Activities**, choose **Applications** and click **Printing**. Alternatively, run the `system-config-printer` command from the command line to start the tool.

The **Printer Configuration** window depicted in **Figure 13.2, “Printer Configuration window”** appears.

![Figure 13.2. Printer Configuration window](image)

13.3.2. Starting Printer Setup

Printer setup process varies depending on the printer queue type.

If you are setting up a local printer connected with USB, the printer is discovered and added automatically. You will be prompted to confirm the packages to be installed and provide the root password. Local printers connected with other port types and network printers need to be set up manually.
Follow this procedure to start a manual printer setup:

1. Start the Printer Configuration tool (refer to Section 13.3.1, “Starting the Printer Configuration Tool”).
2. Go to Server → New → Printer.
3. In the Authentication Required box, type the root user password and confirm.
4. Select the printer connection type and provide its details in the area on the right.

### 13.3.3. Adding a Local Printer

Follow this procedure to add a local printer connected with other than a serial port:

1. Open the New Printer dialog (refer to Section 13.3.2, “Starting Printer Setup”).
2. If the device does not appear automatically, select the port to which the printer is connected in the list on the left (such as Serial Port #1 or LPT #1).
3. On the right, enter the connection properties:
   - for **Enter URI**
     - URI (for example file:/dev/lp0)
   - for **Serial Port**
     - Baud Rate
     - Parity
     - Data Bits
     - Flow Control
4. Click \texttt{Forward}.

5. Select the printer model. Refer to Section 13.3.8, "Selecting the Printer Model and Finishing" for details.

\subsection*{13.3.4. Adding an AppSocket/HP JetDirect printer}

Follow this procedure to add an AppSocket/HP JetDirect printer:

1. Open the New Printer dialog (refer to \textit{Section 13.3.1, "Starting the Printer Configuration Tool"}).

2. In the list on the left, select \textbf{Network Printer} $\rightarrow$ \textbf{AppSocket/HP JetDirect}.

3. On the right, enter the connection settings:

   \begin{itemize}
   \item \textbf{Hostname} \hfill printer hostname or IP address
   \item \textbf{Port Number} \hfill printer port listening for print jobs (9100 by default)
   \end{itemize}
4. Click Forward.

5. Select the printer model. Refer to Section 13.3.8, “Selecting the Printer Model and Finishing” for details.

13.3.5. 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 must be configured to allow incoming TCP connections on port 631. Note that the CUPS browsing protocol allows client machines to discover shared CUPS queues automatically. To enable this, the firewall on the client machine must be configured to allow incoming UDP packets on port 631.

Follow this procedure to add an IPP printer:

1. Open the New Printer dialog (refer to Section 13.3.2, “Starting Printer Setup”).

2. In the list of devices on the left, select Network Printer and Internet Printing Protocol (ipp) or Internet Printing Protocol (https).
3. On the right, enter the connection settings:

   **Host**
   the hostname for the system that controls the printer

   **Queue**
   the queue name to be given to the new queue (if the box is left empty, a name based on the device node will be used)

4. Optionally, click **Verify** to detect the printer.

5. Click **Forward** to continue.

6. Select the printer model. Refer to **Section 13.3.8, “Selecting the Printer Model and Finishing”** for details.

### 13.3.6. Adding an LPD/LPR Host or Printer

Follow this procedure to add an LPD/LPR host or printer:

1. Open the New Printer dialog (refer to **Section 13.3.2, “Starting Printer Setup”**).

2. In the list of devices on the left, select **Network Printer → LPD/LPR Host or Printer**.
3. On the right, enter the connection settings:

   **Host**
   the hostname of the LPD/LPR printer or host

   Optionally, click **Probe** to find queues on the LPD host.

   **Queue**
   the queue name to be given to the new queue (if the box is left empty, a name based on the device node will be used)

![Select Device](image)

   Figure 13.6. Adding an LPD/LPR printer

4. Click **Forward** to continue.

5. Select the printer model. Refer to Section 13.3.8, “Selecting the Printer Model and Finishing” for details.

**13.3.7. Adding a Samba (SMB) printer**

Follow this procedure to add a Samba printer:
Installing the samba-client package

Note that in order to add a Samba printer, you need to have the samba-client package installed. You can do so by running, as root:

```
yum install samba-client
```

For more information on installing packages with Yum, refer to Section 5.2.4, “Installing Packages”.

1. Open the New Printer dialog (refer to Section 13.3.2, “Starting Printer Setup”).
2. In the list on the left, select Network Printer → Windows Printer via SAMBA.
3. Enter the SMB address in the smb:// field. Use the format computer name/printer share. In Figure 13.7, “Adding a SMB printer”, the computer name is dellbox and the printer share is r2.

![Figure 13.7. Adding a SMB printer](image)
4. Click **Browse** to see the available workgroups/domains. To display only queues of a particular host, type in the host name (NetBios name) and click **Browse**.

5. Select either of the options:

   **Prompt user if authentication is required**: username and password are collected from the user when printing a document.

   **Set authentication details now**: provide authentication information now so it is not required later. 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.

6. Enter the **Password** (if required) for the user specified in the **Username** field.

   ![](danger.png) **Be careful when choosing a password**

   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.

7. Click **Verify** to test the connection. Upon successful verification, a dialog box appears confirming printer share accessibility.

8. Click **Forward**.

9. Select the printer model. Refer to **Section 13.3.8, “Selecting the Printer Model and Finishing”** for details.

### 13.3.8. Selecting the Printer Model and Finishing

Once you have properly selected a printer connection type, the system attempts to acquire a driver. If the process fails, you can locate or search for the driver resources manually.

Follow this procedure to provide the printer driver and finish the installation:

1. In the window displayed after the automatic driver detection has failed, select one of the following options:

   **Select printer from database** — the system chooses a driver based on the selected make of your printer from the list of **Makes**. If your printer model is not listed, choose **Generic**.

   **Provide PPD file** — the system uses the provided PostScript Printer Description (PPD) file for installation. A PPD file may also be delivered with your printer as being normally provided by the manufacturer. If the PPD file is available, you can choose this option and use the browser bar below the option description to select the PPD file.
Search for a printer driver to download — enter the make and model of your printer into the **Make and model** field to search on OpenPrinting.org for the appropriate packages.

**Choose Driver**

- Select printer from database
- Provide PPD file
- Search for a printer driver to download

The foomatic printer database contains various manufacturer provided PostScript Printer Description (PPD) files and also can generate PPD files for a large number of (non PostScript) printers. But in general manufacturer provided PPD files provide better access to the specific features of the printer.

![Selecting a printer brand](image)

**Figure 13.8. Selecting a printer brand**

2. Depending on your previous choice provide details in the area displayed below:
   - Printer brand for the **Select printer from database** option
   - PPD file location for the **Provide PPD file** option
   - Printer make and model for the **Search for a printer driver to download** option

3. Click **Forward** to continue.
4. If applicable for your option, window shown in Figure 13.9, “Selecting a printer model” appears. Choose the corresponding model in the **Models** column on the left.

### Selecting a printer driver

On the right, the recommended printed driver is automatically selected; however, you can select another available driver. 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.

![Selecting a printer model](Figure 13.9. Selecting a printer model)

5. Click **Forward**.

6. Under the **Describe 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 add further printer information. Both fields are optional, and may contain spaces.

7. Click **Apply** to confirm your printer configuration and add the print queue if the settings are correct. Click **Back** to modify the printer configuration.

8. After the changes are applied, a dialog box appears allowing you to print a test page. Click **Print Test Page** to print a test page now. Alternatively, you can print a test page also later, refer to **Section 13.3.9, “Printing a test page”** for details.

### 13.3.9. Printing a test page

After you have set up a printer or changed a printer configuration, print a test page to make sure the printer is functioning properly:

1. Right-click the printer in the **Printing** window and click **Properties**.
2. In the Properties window, click **Settings** on the left.
3. On the displayed **Settings** tab, click the **Print Test Page** button.
13.3.10. Modifying Existing Printers
To delete an existing printer, in the Printer Configuration window, select the printer and go to Printer → Delete. Confirm the printer deletion. Alternatively, press the Delete key.

To set the default printer, right-click the printer in the printer list and click the Set As Default button in the context menu.

13.3.10.1. The Settings Page
To change printer driver configuration, double-click the corresponding name in the Printer list and click the Settings label on the left to display the Settings page.

You can modify printer settings such as make and model, print a test page, change the device location (URI), and more.

![Settings page](image)

Figure 13.11. Settings page

13.3.10.2. The Policies Page
Click the Policies button on the left to change settings in printer state and print output.

You can select the printer states, configure the Error Policy of the printer (you can decide to abort the print job, retry, or stop it if an error occurs).

You can also create a banner page (a page that describes aspects of the print job such as the originating printer, the username from 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).

### 13.3.10.2.1. Sharing Printers

On the Policies page, you can mark a printer as shared: if a printer is shared, users published on the network can use it. To allow the sharing function for printers, go to Server → Settings and select Publish shared printers connected to this system.

Finally, ensure that the firewall allows incoming TCP connections to port 631, which is Network Printing Server (IPP) in system-config-firewall.

![Figure 13.12. Policies page](image)

### 13.3.10.2.2. The Access Control Page

You can change user-level access to the configured printer on the Access Control page. Click the Access Control label on the left to display the page. Select either Allow printing for everyone except these users or Deny printing for everyone except these users and define the user set below: enter the user name in the text box and click the Add button to add the user to the user set.
13.3.10.2.3. The Printer Options Page

The **Printer Options** page contains various configuration options for the printer media and output, and its content may vary from printer to printer. It contains general printing, paper, quality, and printing size settings.
13.3.10.2.4. Job Options Page

On the Job Options page, you can detail the printer job options. Click the Job Options label on the left to display the page. Edit the default settings to apply custom job options, such as number of copies, orientation, pages per side, 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), detailed text options, and custom job options.
13.3.10.2.5. Ink/Toner Levels Page

The Ink/Toner Levels page contains details on toner status if available and printer status messages. Click the Ink/Toner Levels label on the left to display the page.
13.3.10.3. 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 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 job number, and more.

During the printing process, messages informing about the process appear in the notification area.

<table>
<thead>
<tr>
<th>Job</th>
<th>User</th>
<th>Document</th>
<th>Printer</th>
<th>Size</th>
<th>Time submitted</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>eslobo</td>
<td>Product Document</td>
<td>canon-printer</td>
<td>6k</td>
<td></td>
<td>Pending</td>
</tr>
<tr>
<td>21</td>
<td>eslobo</td>
<td>Red Hat</td>
<td>canon-printer</td>
<td>6k</td>
<td>a minute ago</td>
<td>Processing</td>
</tr>
</tbody>
</table>

To cancel, hold, release, reprint or authenticate a print job, select the job in the GNOME Print Status and on the Job menu, click the respective command.
To view the list of print jobs in the print spool from a shell prompt, type the command `lpstat -o`. The last few lines look similar to the following:

**Example 13.1. Example of `lpstat -o` output**

```
$ lpstat -o
Charlie-69          twaugh          1024   Tue 08 Feb 2011 16:42:11 GMT
Aaron-61            twaugh          1024   Tue 08 Feb 2011 16:42:44 GMT
Ben-62              root             1024   Tue 08 Feb 2011 16:45:42 GMT
```

If you want to cancel a print job, find the job number of the request with the command `lpstat -o` and then use the command `cancel job number`. For example, `cancel 60` would cancel the print job in _Example 13.1, “Example of `lpstat -o` output”_. You cannot cancel print jobs that were started by other users with the `cancel` command. However, you can enforce deletion of such job by issuing the `cancel -U root job_number` command. To prevent such canceling, change the printer operation policy to `Authenticated` to force root authentication.

You can also print a file directly from a shell prompt. For example, the command `lp 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.

### 13.3.11. Additional Resources

To learn more about printing on Fedora, refer to the following resources.

#### 13.3.11.1. Installed Documentation

- **man lp**
  
  The manual page for the `lpr` command that allows you to print files from the command line.

- **man cancel**
  
  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.

- **man lpstat**
  
  The manual page for the `lpstat` command, which displays status information about classes, jobs, and printers.

#### 13.3.11.2. Useful Websites

- [http://www.linuxprinting.org/](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.
Configuring NTP Using the chrony Suite

Accurate time keeping is important for a number of reasons in IT. In networking for example, accurate time stamps in packets and logs are required. In Linux systems, the NTP protocol is implemented by a daemon running in user space.

The user space daemon updates the system clock running in the kernel. The system clock can keep time by using various clock sources. Usually, the Time Stamp Counter (TSC) is used. The TSC is a CPU register which counts the number of cycles since it was last reset. It is very fast, has a high resolution, and there are no interrupts.

There is a choice between the daemons ntpd and chronyd, which are available from the repositories in the ntp and chrony packages respectively. This section describes the use of the chrony suite of utilities to update the system clock on systems that do not fit into the conventional permanently networked, always on, dedicated server category.

14.1. Introduction to the chrony Suite

Chrony consists of chronyd, a daemon that runs in user space, and chronyc, a command line program for making adjustments to chronyd. Systems which are not permanently connected, or not permanently powered up, take a relatively long time to adjust their system clocks with ntpd. This is because many small corrections are made based on observations of the clocks drift and offset. Temperature changes, which may be significant when powering up a system, affect the stability of hardware clocks. Although adjustments begin within a few milliseconds of booting a system, acceptable accuracy may take anything from ten seconds from a warm restart to a number of hours depending on your requirements, operating environment and hardware. chrony is a different implementation of the NTP protocol than ntpd, it can adjust the system clock more rapidly.

14.1.1. Differences Between ntpd and chronyd

One of the main differences between ntpd and chronyd is in the algorithms used to control the computer's clock. Things chronyd can do better than ntpd are:

• chronyd can work well when external time references are only intermittently accessible whereas ntpd needs regular polling of time reference to work well.

• chronyd can perform well even when the network is congested for longer periods of time.

• chronyd can usually synchronize the clock faster and with better time accuracy.

• chronyd quickly adapts to sudden changes in the rate of the clock, for example, due to changes in the temperature of the crystal oscillator, whereas ntpd may need a long time to settle down again.

• chronyd in the default configuration never steps the time after the clock has been synchronized at system start, in order not to upset other running programs. ntpd can be configured to never step the time too, but it has to use a different means of adjusting the clock, which has some disadvantages.

• chronyd can adjust the rate of the clock on a Linux system in a larger range, which allows it to operate even on machines with a broken or unstable clock. For example, on some virtual machines.

Things chronyd can do that ntpd cannot do:
Chapter 14. Configuring NTP Using the chrony Suite

- **chronyd** provides support for isolated networks where the only method of time correction is manual entry. For example, by the administrator looking at a clock. **chronyd** can look at the errors corrected at different updates to estimate the rate at which the computer gains or loses time, and use this estimate to trim the computer clock subsequently.

- **chronyd** provides support to work out the rate of gain or loss of the real-time clock, the hardware clock, that maintains the time when the computer is turned off. It can use this data when the system boots to set the system time using an adjusted value of the time taken from the real-time clock. This is, at time of writing, only available in Linux.

Things **ntpd** can do that **chronyd** cannot do:

- **ntpd** fully supports NTP version 4 (*RFC5905*), including broadcast, multicast, manycast clients and servers, and the orphan mode. It also supports extra authentication schemes based on public-key cryptography (*RFC5906*). **chronyd** uses NTP version 3 (*RFC1305*), which is compatible with version 4.

- **ntpd** includes drivers for many reference clocks whereas **chronyd** relies on other programs, for example **gpsd**, to access the data from the reference clocks.

### 14.1.2. Choosing Between NTP Daemons

- **Chrony** should be considered for all systems which are frequently suspended or otherwise intermittently disconnected and reconnected to a network. Mobile and virtual systems for example.

- The NTP daemon (**ntpd**) should be considered for systems which are normally kept permanently on. Systems which are required to use broadcast or multicast IP, or to perform authentication of packets with the Autokey protocol, should consider using **ntpd**. **Chrony** only supports symmetric key authentication using a message authentication code (MAC) with MD5, SHA1 or stronger hash functions, whereas **ntpd** also supports the Autokey authentication protocol which can make use of the PKI system. **Autokey** is described in *RFC5906*.

### 14.2. Understanding chrony and Its Configuration

#### 14.2.1. Understanding chronyd

The **chrony** daemon, **chronyd**, running in user space, makes adjustments to the system clock which is running in the kernel. It does this by consulting external time sources, using the NTP protocol, when ever network access allows it to do so. When external references are not available, **chronyd** will use the last calculated drift stored in the drift file. It can also be commanded manually to make corrections, by **chronyc**.

#### 14.2.2. Understanding chronyc

The **chrony** daemon, **chronyd**, can be controlled by the command line utility **chronyc**. This utility provides a command prompt which allows entering of a number of commands to make changes to **chronyd**. The default configuration is for **chronyd** to only accept commands from a local instance of **chronyc**, but **chronyc** can be used to alter the configuration so that **chronyd** will allow external control. That is to say, **chronyc** can be run remotely after first configuring **chronyd** to accept remote connections. The IP addresses allowed to connect to **chronyd** should be tightly controlled.
14.2.3. Understanding the chrony Configuration Commands

The default configuration file for chronyd is /etc/chrony.conf. The -f option can be used to specify an alternate configuration file path. Refer to the chronyd man page for further options. For a complete list of the directives that can be used see http://chrony.tuxfamily.org/manual.html#Configuration-file. We present here a selection of configuration options:

Comments
Comments should be preceded by #, %, ; or !

allow
Optionally specify a host, subnet, or network from which to allow NTP connections to a machine acting as NTP server. The default is not to allow connections.

Examples:
1. allow server1.example.com
   Use this form to specify a particular host, by its host name, to be allowed access.
2. allow 192.0.2.0/24
   Use this form to specify a particular network to be allowed access.
3. allow 2001:db8::/32
   Use this form to specify an IPv6 address to be allowed access.

cmdallow
This is similar to the allow directive (see section allow), except that it allows control access (rather than NTP client access) to a particular subnet or host. (By “control access” is meant that chronyc can be run on those hosts and successfully connect to chronyd on this computer.) The syntax is identical. There is also a cmddeny all directive with similar behavior to the cmdallow all directive.

dumpdir
Path to the directory to save the measurement history across restarts of chronyd (assuming no changes are made to the system clock behavior whilst it is not running). If this capability is to be used (via the dumponexit command in the configuration file, or the dump command in chronyc), the dumpdir command should be used to define the directory where the measurement histories are saved.

dumponexit
If this command is present, it indicates that chronyd should save the measurement history for each of its time sources recorded whenever the program exits. (See the dumpdir command above).

local
The local keyword is used to allow chronyd to appear synchronized to real time (from the viewpoint of clients polling it), even if it has no current synchronization source. This option is normally used on computers in an isolated network, where several computers are required to synchronize to one other, this being the “master” which is kept vaguely in line with real time by manual input.

An example of the command is:
A large value of 10 indicates that the clock is so many hops away from a reference clock that its
time is fairly unreliable. Put another way, if the computer ever has access to another computer
which is ultimately synchronized to a reference clock, it will almost certainly be at a stratum less
than 10. Therefore, the choice of a high value like 10 for the \texttt{local} command prevents the
machine’s own time from ever being confused with real time, were it ever to leak out to clients that
have visibility of real servers.

\textbf{log}

The \texttt{log} command indicates that certain information is to be logged. It accepts the following
options:

\textbf{measurements}

This option logs the raw NTP measurements and related information to a file called
\texttt{measurements.log}.

\textbf{statistics}

This option logs information about the regression processing to a file called
\texttt{statistics.log}.

\textbf{tracking}

This option logs changes to the estimate of the system’s gain or loss rate, and any slews
made, to a file called \texttt{tracking.log}.

\textbf{rtc}

This option logs information about the system’s real-time clock.

\textbf{refclocks}

This option logs the raw and filtered reference clock measurements to a file called
\texttt{refclocks.log}.

\textbf{tempcomp}

This option logs the temperature measurements and system rate compensations to a file
called \texttt{tempcomp.log}.

The log files are written to the directory specified by the \texttt{logdir} command. An example of the
command is:

\begin{verbatim}
log measurements statistics tracking
\end{verbatim}

\textbf{logdir}

This directive allows the directory where log files are written to be specified. An example of the use
of this directive is:

\begin{verbatim}
logdir /var/log/chrony
\end{verbatim}

\textbf{makestep}

Normally \texttt{chronyd} will cause the system to gradually correct any time offset, by slowing down or
speeding up the clock as required. In certain situations, the system clock may be so far adrift that
this slewing process would take a very long time to correct the system clock. This directive forces
\texttt{chronyd} to step system clock if the adjustment is larger than a threshold value, but only if there
were no more clock updates since \texttt{chronyd} was started than a specified limit (a negative value
can be used to disable the limit). This is particularly useful when using reference clocks, because
the \texttt{initstepslew} directive only works with NTP sources.
An example of the use of this directive is:

```
makestep 1000 10
```

This would step the system clock if the adjustment is larger than 1000 seconds, but only in the first ten clock updates.

**maxchange**

This directive sets the maximum allowed offset corrected on a clock update. The check is performed only after the specified number of updates to allow a large initial adjustment of the system clock. When an offset larger than the specified maximum occurs, it will be ignored for the specified number of times and then **chronyd** will give up and exit (a negative value can be used to never exit). In both cases a message is sent to syslog.

An example of the use of this directive is:

```
maxchange 1000 1 2
```

After the first clock update, **chronyd** will check the offset on every clock update, it will ignore two adjustments larger than 1000 seconds and exit on another one.

**maxupdateskew**

One of **chronyd**'s tasks is to work out how fast or slow the computer's clock runs relative to its reference sources. In addition, it computes an estimate of the error bounds around the estimated value. If the range of error is too large, it probably indicates that the measurements have not settled down yet, and that the estimated gain or loss rate is not very reliable. The **maxupdateskew** parameter allows the threshold for determining whether an estimate may be so unreliable that it should not be used. By default, the threshold is 1000 ppm. The format of the syntax is:

```
maxupdateskew skew-in-ppm
```

Typical values for `skew-in-ppm` might be 100 for a dial-up connection to servers over a telephone line, and 5 or 10 for a computer on a LAN. It should be noted that this is not the only means of protection against using unreliable estimates. At all times, **chronyd** keeps track of both the estimated gain or loss rate, and the error bound on the estimate. When a new estimate is generated following another measurement from one of the sources, a weighted combination algorithm is used to update the master estimate. So if **chronyd** has an existing highly-reliable master estimate and a new estimate is generated which has large error bounds, the existing master estimate will dominate in the new master estimate.

**noclientlog**

This directive, which takes no arguments, specifies that client accesses are not to be logged. Normally they are logged, allowing statistics to be reported using the clients command in **chronyc**.

**reselectdist**

When **chronyd** selects synchronization source from available sources, it will prefer the one with minimum synchronization distance. However, to avoid frequent reselecting when there are sources with similar distance, a fixed distance is added to the distance for sources that are currently not selected. This can be set with the **reselectdist** option. By default, the distance is 100 microseconds.

The format of the syntax is:

```
reselectdist dist-in-seconds
```
Chapter 14. Configuring NTP Using the chrony Suite

stratumweight

The **stratumweight** directive sets how much distance should be added per stratum to the synchronization distance when *chronyd* selects the synchronization source from available sources.

The format of the syntax is:

```
stratumweight dist-in-seconds
```

By default, *dist-in-seconds* is 1 second. This usually means that sources with lower stratum will be preferred to sources with higher stratum even when their distance is significantly worse. Setting **stratumweight** to 0 makes *chronyd* ignore stratum when selecting the source.

rtcfile

The **rtcfile** directive defines the name of the file in which *chronyd* can save parameters associated with tracking the accuracy of the system's real-time clock (RTC). The format of the syntax is:

```
rtcfile /var/lib/chrony/rtc
```

*chronyd* saves information in this file when it exits and when the `writertc` command is issued in *chronyc*. The information saved is the RTC's error at some epoch, that epoch (in seconds since January 1 1970), and the rate at which the RTC gains or loses time. Not all real-time clocks are supported as their code is system-specific. Note that if this directive is used then real-time clock should not be manually adjusted as this would interfere with *chrony*'s need to measure the rate at which the real-time clock drifts if it was adjusted at random intervals.

rtcsync

The **rtcsync** directive is present in the `/etc/chrony.conf` file by default. This will inform the kernel the system clock is kept synchronized and the kernel will update the real-time clock every 11 minutes.

### 14.2.4. Security with chronyc

As access to *chronyc* allows changing *chronyd* just as editing the configuration files would, access to *chronyc* should be limited. Passwords can be specified in the key file, written in ASCII or HEX, to restrict the use of *chronyc*. One of the entries is used to restrict the use of operational commands and is referred to as the command key. In the default configuration, a random command key is generated automatically on start. It should not be necessary to specify or alter it manually.

Other entries in the key file can be used as NTP keys to authenticate packets received from remote NTP servers or peers. The two sides need to share a key with identical ID, hash type and password in their key file. This requires manually creating the keys and copying them over a secure medium, such as SSH. If the key ID was, for example, 10 then the systems that act as clients must have a line in their configuration files in the following format:

```
server w.x.y.z key 10
peer w.x.y.z key 10
```

The location of the key file is specified in the `/etc/chrony.conf` file. The default entry in the configuration file is:

```
keyfile /etc/chrony.keys
```
The command key number is specified in `/etc/chrony.conf` using the `commandkey` directive, it is the key `chronyd` will use for authentication of user commands. The directive in the configuration file takes the following form:

```
commandkey 1
```

An example of the format of the default entry in the key file, `/etc/chrony.keys`, for the command key is:

```
1 SHA1 HEX:A6CFC56C93AB6E5A19754C246242FC5471BCDF
```

Where `1` is the key ID, SHA1 is the hash function to use, `HEX` is the format of the key, and `A6CFC50C9C93AB6E5A19754C246242FC5471BCDF` is the key randomly generated when `chronyd` was started for the first time. The key can be given in hexadecimal or ASCII format (the default).

A manual entry in the key file, used to authenticate packets from certain NTP servers or peers, can be as simple as the following:

```
20 foobar
```

Were `20` is the key ID and `foobar` is the secret authentication key. The default hash is MD5, and ASCII is the default format for the key.

By default, `chronyd` is configured to listen for commands only from `localhost` (127.0.0.1 and ::1) on port 323. To access `chronyd` remotely with `chronyc`, any `bindcmdaddress` directives in the `/etc/chrony.conf` file should be removed to enable listening on all interfaces and the `cmdallow` directive should be used to allow commands from the remote IP address, network, or subnet. In addition, port 323 has to be opened in the firewall in order to connect from a remote system. Note that the `allow` directive is for NTP access whereas the `cmdallow` directive is to enable the receiving of remote commands. It is possible to make these changes temporarily using `chronyc` running locally. Edit the configuration file to make persistent changes.

The communication between `chronyc` and `chronyd` is done over UDP, so it needs to be authorized before issuing operational commands. To authorize, use the `authhash` and `password` commands as follows:

```
chronyc> authhash SHA1
chronyc> password HEX:A6CFC56C93AB6E5A19754C246242FC5471BCDF
200 OK
```

If `chronyc` is used to configure the local `chronyd`, the `-a` option will run the `authhash` and `password` commands automatically.

Only the following commands can be used without providing a password: `activity`, `authhash`, `dns`, `exit`, `help`, `password`, `quit`, `rtcdata`, `sources`, `sourcestats`, `tracking`, `waitsync`.

# 14.3. Using chrony

## 14.3.1. Checking if chrony is Installed

To check if `chrony` is installed, run the following command as root:

```
-]# yum install chrony
```
The default location for the `chrony` daemon is `/usr/sbin/chronyd`. The command line utility will be installed to `/usr/bin/chronyc`.

### 14.3.2. Installing chrony

To install `chrony`, run the following command as root:

```
~# yum install chrony -y
```

### 14.3.3. Checking the Status of chronyd

To check the status of `chronyd`, issue the following command:

```
~$ systemctl status chronyd
chronyd.service - NTP client/server
   Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled)
   Active: active (running) since Wed 2013-06-12 22:23:16 CEST; 11h ago
```

### 14.3.4. Starting chronyd

To start `chronyd`, issue the following command as root:

```
~# systemctl start chronyd
```

To ensure `chronyd` starts automatically at system start, issue the following command as root:

```
~# systemctl enable chronyd
```

### 14.3.5. Stopping chronyd

To stop `chronyd`, issue the following command as root:

```
~# systemctl stop chronyd
```

To prevent `chronyd` from starting automatically at system start, issue the following command as root:

```
~# systemctl disable chronyd
```

### 14.3.6. Checking if chrony is Synchronized

To check if `chrony` is synchronized, make use of the `tracking`, `sources`, and `sourcestats` commands.

#### 14.3.6.1. Checking chrony Tracking

To check `chrony` tracking, issue the following command:

```
~$ chronyc tracking
Reference ID    : 1.2.3.4 (a.b.c)
Stratum         : 3
Ref time (UTC)  : Fri Feb  3 15:00:29 2012
```
Checking if chrony is Synchronized

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>System time</td>
<td>0.000001501 seconds slow of NTP time</td>
</tr>
<tr>
<td>Last offset</td>
<td>-0.000001632 seconds</td>
</tr>
<tr>
<td>RMS offset</td>
<td>0.000002360 seconds</td>
</tr>
<tr>
<td>Frequency</td>
<td>331.898 ppm fast</td>
</tr>
<tr>
<td>Residual freq</td>
<td>0.004 ppm</td>
</tr>
<tr>
<td>Skew</td>
<td>0.154 ppm</td>
</tr>
<tr>
<td>Root delay</td>
<td>0.373169 seconds</td>
</tr>
<tr>
<td>Root dispersion</td>
<td>0.024780 seconds</td>
</tr>
<tr>
<td>Update interval</td>
<td>64.2 seconds</td>
</tr>
<tr>
<td>Leap status</td>
<td>Normal</td>
</tr>
</tbody>
</table>

The fields are as follows:

Reference ID
This is the reference ID and name (or IP address) if available, of the server to which the computer is currently synchronized. If this is 127.127.1.1 it means the computer is not synchronized to any external source and that you have the “local” mode operating (via the local command in `chronyc`, or the `local` directive in the `/etc/chrony.conf` file (see section `local`).

Stratum
The stratum indicates how many hops away from a computer with an attached reference clock we are. Such a computer is a stratum-1 computer, so the computer in the example is two hops away (that is to say, a.b.c is a stratum-2 and is synchronized from a stratum-1).

Ref time
This is the time (UTC) at which the last measurement from the reference source was processed.

System time
In normal operation, `chrony` never steps the system clock, because any jump in the timescale can have adverse consequences for certain application programs. Instead, any error in the system clock is corrected by slightly speeding up or slowing down the system clock until the error has been removed, and then returning to the system clock’s normal speed. A consequence of this is that there will be a period when the system clock (as read by other programs using the `gettimeofday()` system call, or by the date command in the shell) will be different from `chrony`’s estimate of the current true time (which it reports to NTP clients when it is operating in server mode). The value reported on this line is the difference due to this effect.

Last offset
This is the estimated local offset on the last clock update.

RMS offset
This is a long-term average of the offset value.

Frequency
The “frequency” is the rate by which the system’s clock would be would be wrong if `chrony` was not correcting it. It is expressed in ppm (parts per million). For example, a value of 1ppm would mean that when the system’s clock thinks it has advanced 1 second, it has actually advanced by 1.000001 seconds relative to true time.

Residual freq
This shows the “residual frequency” for the currently selected reference source. This reflects any difference between what the measurements from the reference source indicate the frequency should be and the frequency currently being used. The reason this is not always zero is that a smoothing procedure is applied to the frequency. Each time a measurement from the reference source is obtained and a new residual frequency computed, the estimated accuracy of this residual is compared with the estimated accuracy (see `skew` next) of the existing frequency value. A weighted average is computed for the new frequency, with weights depending on these
accordances. If the measurements from the reference source follow a consistent trend, the residual will be driven to zero over time.

Skew
This is the estimated error bound on the frequency.

Root delay
This is the total of the network path delays to the stratum-1 computer from which the computer is ultimately synchronized. In certain extreme situations, this value can be negative. (This can arise in a symmetric peer arrangement where the computers’ frequencies are not tracking each other and the network delay is very short relative to the turn-around time at each computer.)

Root dispersion
This is the total dispersion accumulated through all the computers back to the stratum-1 computer from which the computer is ultimately synchronized. Dispersion is due to system clock resolution, statistical measurement variations etc.

Leap status
This is the leap status, which can be Normal, Insert second, Delete second or Not synchronized.

### 14.3.6.2. Checking chrony Sources

The `sources` command displays information about the current time sources that `chronyd` is accessing. The optional argument `-v` can be specified, meaning verbose. In this case, extra caption lines are shown as a reminder of the meanings of the columns.

```
$ chronyc sources
210 Number of sources = 3
MS Name/IP address Stratum Poll Reach LastRx Last sample
#* GPS0                          0   4   377    11   -479ns[ -621ns] +/- 134ns
^? a.b.c                         2   6   377    23   -923us[ -924us] +/-   43ms
^+ d.e.f                         1   6   377    21  -2629us[-2619us] +/-   86ms
```

The columns are as follows:

- **M**: This indicates the mode of the source. ^ means a server, = means a peer and # indicates a locally connected reference clock.

- **S**: This column indicates the state of the sources. “*” indicates the source to which `chronyd` is currently synchronized. “+” indicates acceptable sources which are combined with the selected source. “-” indicates acceptable sources which are excluded by the combining algorithm. “?” indicates sources to which connectivity has been lost or whose packets do not pass all tests. “x” indicates a clock which `chronyd` thinks is is a falseticker (that is to say, its time is inconsistent with a majority of other sources). “~” indicates a source whose time appears to have too much variability. The “?” condition is also shown at start-up, until at least 3 samples have been gathered from it.

- **Name/IP address**: This shows the name or the IP address of the source, or reference ID for reference clocks.

- **Stratum**: This shows the stratum of the source, as reported in its most recently received sample. Stratum 1 indicates a computer with a locally attached reference clock. A computer that is synchronized to a
Checking if chrony is Synchronized

stratum 1 computer is at stratum 2. A computer that is synchronized to a stratum 2 computer is at stratum 3, and so on.

Poll
This shows the rate at which the source is being polled, as a base-2 logarithm of the interval in seconds. Thus, a value of 6 would indicate that a measurement is being made every 64 seconds. chronyd automatically varies the polling rate in response to prevailing conditions.

Reach
This shows the source’s reachability register printed as octal number. The register has 8 bits and is updated on every received or missed packet from the source. A value of 377 indicates that a valid reply was received for all from the last eight transmissions.

LastRx
This column shows how long ago the last sample was received from the source. This is normally in seconds. The letters m, h, d or y indicate minutes, hours, days or years. A value of 10 years indicates there were no samples received from this source yet.

Last sample
This column shows the offset between the local clock and the source at the last measurement. The number in the square brackets shows the actual measured offset. This may be suffixed by ns (indicating nanoseconds), us (indicating microseconds), ms (indicating milliseconds), or s (indicating seconds). The number to the left of the square brackets shows the original measurement, adjusted to allow for any slews applied to the local clock since. The number following the +/- indicator shows the margin of error in the measurement. Positive offsets indicate that the local clock is fast of the source.

14.3.6.3. Checking chrony Source Statistics
The sourcestats command displays information about the drift rate and offset estimation process for each of the sources currently being examined by chrony. The optional argument -v can be specified, meaning verbose. In this case, extra caption lines are shown as a reminder of the meanings of the columns.

~]$ chronyc sourcestats

210 Number of sources = 1
Name/IP Address NP NR Span Frequency Freq Skew Offset Std Dev
===============================================================================
abc.def.ghi

The columns are as follows:
Name/IP address
This is the name or IP address of the NTP server (or peer) or reference ID of the reference clock to which the rest of the line relates.

NP
This is the number of sample points currently being retained for the server. The drift rate and current offset are estimated by performing a linear regression through these points.

NR
This is the number of runs of residuals having the same sign following the last regression. If this number starts to become too small relative to the number of samples, it indicates that a straight line is no longer a good fit to the data. If the number of runs is too low, chrony discards older samples and re-runs the regression until the number of runs becomes acceptable.
14.3.7. Manually Adjusting the System Clock

To update, or step, the system clock immediately, bypassing any adjustments in progress by slewing the clock, issue the following commands as root:

```bash
~]# chronyc password commandkey-password
200 OK
chrony> makestep
200 OK
```

Where `commandkey-password` is the command key or password stored in the key file.

The real-time clock should not be manually adjusted if the `rtcfile` directive is used as this would interfere with `chrony`'s need to measure the rate at which the real-time clock drifts if it was adjusted at random intervals.

If `chronyc` is used to configure the local `chronyd`, the `-a` will run the `authhash` and `password` commands automatically. This means that the interactive session illustrated above can be replaced by:

```
chronyc -a makestep
```

14.4. Setting Up chrony for Different Environments

14.4.1. Setting Up chrony for a System Which is Infrequently Connected

This example is intended for systems which use dial-on-demand connections. The normal configuration should be sufficient for mobile and virtual devices which connect intermittently. First, review and confirm that the default settings in the `/etc/chrony.conf` are similar to the following:

```
driftfile /var/lib/chrony/drift
commandkey 1
keyfile /etc/chrony.keys
```

The command key ID is generated at install time and should correspond with the `commandkey` value in the key file, `/etc/chrony.keys`. 
1. Using your editor running as root, add the addresses of four NTP servers as follows:

```
server 0.pool.ntp.org offline
server 1.pool.ntp.org offline
server 2.pool.ntp.org offline
server 3.pool.ntp.org offline
```

The **offline** option can be useful in preventing systems from trying to activate connections. The **chrony** daemon will wait for **chronyc** to inform it that the system is connected to the network or Internet.

### 14.4.2. Setting Up chrony for a System in an Isolated Network

For a network that is never connected to the Internet, one computer is selected to be the master timeserver. The other computers are either direct clients of the master, or clients of clients. On the master, the drift file must be manually set with the average rate of drift of the system clock. If the master is rebooted it will obtain the time from surrounding systems and take an average to set its system clock. Thereafter it resumes applying adjustments based on the drift file. The drift file will be updated automatically when the **settime** command is used.

On the system selected to be the master, using a text editor running as root, edit the `/etc/chrony.conf` as follows:

```
# driftfile /var/lib/chrony/drift
# commandkey 1
# keyfile /etc/chrony.keys
# initstepslew 10 client1 client3 client6
local stratum 8
manual
allow 192.0.2.0
```

Where **192.0.2.0** is the network or subnet address from which the clients are allowed to connect.

On the systems selected to be direct clients of the master, using a text editor running as root, edit the `/etc/chrony.conf` as follows:

```
server master
driftfile /var/lib/chrony/drift
logdir /var/log/chrony
log measurements statistics tracking
keyfile /etc/chrony.keys
commandkey 24
local stratum 10
initstepslew 20 master
allow 192.0.2.123
```

Where **192.0.2.123** is the address of the master, and **master** is the host name of the master. These client will resynchronize the master if it restarts.

On the client systems which are not to be direct clients of the master, the `/etc/chrony.conf` file should be the same except that the **local** and **allow** directives should be omitted.
14.5. Using chronyc

14.5.1. Using chronyc to Control chronyd
To make changes using the command line utility chronyc in interactive mode, enter the following command as root:

```
~ ]# chronyc
```

chronyc must run as root if some of the restricted commands are to be used.

The chronyc command prompt will be displayed as follows:

```
chronyc>
```

You can type help to list all of the commands.

The utility can also be invoked in non-interactive command mode if called together with a command as follows:

```
~ ]# chronyc command
```

14.5.2. Using chronyc for Remote Administration
To configure chrony to connect to a remote instance of chronyd, issue a command as root in the following format:

```
~ ]# chronyc -h hostname
```

Where hostname is the hostname of a system running chronyd to connect to in order to allow remote administration from that host. The default is to connect to the daemon on the localhost.

To configure chrony to connect to a remote instance of chronyd on a non-default port, issue a command as root in the following format:

```
~ ]# chronyc -h hostname -p port
```

Where port is the port in use for controlling and monitoring by the instance of chronyd to be connected to.

Note that commands issued at the chronyc command prompt are not persistent. Only commands in the configuration file are persistent.

From the remote systems, the system administrator can issue commands after first using the password command, preceded by the authhash command if the key used a hash different from MD5, at the chronyc command prompt as follows:

```
chronyc> password secretpasswordwithnospaces
200 OK
```

The password or hash associated with the command key for a remote system is best obtained by SSH. That is to say, an SSH connection should be established to the remote machine and the ID of the command key from /etc/chrony.conf and the command key in /etc/chrony.keys memorized or stored securely for the duration of the session.
14.6. Additional Resources

The following sources of information provide additional resources regarding chrony.

14.6.1. Installed Documentation

- **chrony(1)** man page — Introduces the chrony daemon and the command-line interface tool.

- **chronyc(1)** man page — Describes the chronyc command-line interface tool including commands and command options.

- **chronyd(1)** man page — Describes the chronyd daemon including commands and command options.

- **chrony.conf(5)** man page — Describes the chrony configuration file.

- **/usr/share/doc/chrony*/chrony.txt** — User guide for the chrony suite.

14.6.2. Useful Websites


The on-line user guide for chrony.
Configuring NTP Using ntpd

15.1. Introduction to NTP

The Network Time Protocol (NTP) enables the accurate dissemination of time and date information in order to keep the time clocks on networked computer systems synchronized to a common reference over the network or the Internet. Many standards bodies around the world have atomic clocks which may be made available as a reference. The satellites that make up the Global Position System contain more than one atomic clock, making their time signals potentially very accurate. Their signals can be deliberately degraded for military reasons. An ideal situation would be where each site has a server, with its own reference clock attached, to act as a site-wide time server. Many devices which obtain the time and date via low frequency radio transmissions or the Global Position System (GPS) exist. However for most situations, a range of publicly accessible time servers connected to the Internet at geographically dispersed locations can be used. These NTP servers provide “Coordinated Universal Time” (UTC). Information about these time servers can found at www.pool.ntp.org.

Accurate time keeping is important for a number of reasons in IT. In networking for example, accurate time stamps in packets and logs are required. Logs are used to investigate service and security issues and so timestamps made on different systems must be made by synchronized clocks to be of real value. As systems and networks become increasingly faster, there is a corresponding need for clocks with greater accuracy and resolution. In some countries there are legal obligations to keep accurately synchronized clocks. Please see www.ntp.org for more information. In Linux systems, NTP is implemented by a daemon running in user space. The default NTP user space daemon in Fedora 20 is chronyd. It must be disabled if you want to use the ntpd daemon. See Chapter 14, Configuring NTP Using the chrony Suite for information on chrony.

The user space daemon updates the system clock, which is a software clock running in the kernel. Linux uses a software clock as its system clock for better resolution than the typical embedded hardware clock referred to as the “Real Time Clock” (RTC). See the rtc(4) and hwclock(8) man pages for information on hardware clocks. The system clock can keep time by using various clock sources. Usually, the Time Stamp Counter (TSC) is used. The TSC is a CPU register which counts the number of cycles since it was last reset. It is very fast, has a high resolution, and there are no interrupts. On system start, the system clock reads the time and date from the RTC. The time kept by the RTC will drift away from actual time by up to 5 minutes per month due to temperature variations. Hence the need for the system clock to be constantly synchronized with external time references. When the system clock is being synchronized by ntpd, the kernel will in turn update the RTC every 11 minutes automatically.

15.2. NTP Strata

NTP servers are classified according to their synchronization distance from the atomic clocks which are the source of the time signals. The servers are thought of as being arranged in layers, or strata, from 1 at the top down to 15. Hence the word stratum is used when referring to a specific layer. Atomic clocks are referred to as Stratum 0 as this is the source, but no Stratum 0 packet is sent on the Internet, all stratum 0 atomic clocks are attached to a server which is referred to as stratum 1. These servers send out packets marked as Stratum 1. A server which is synchronized by means of packets marked stratum n belongs to the next, lower, stratum and will mark its packets as stratum n+1. Servers of the same stratum can exchange packets with each other but are still designated as belonging to just the one stratum, the stratum one below the best reference they are synchronized to. The designation Stratum 16 is used to indicate that the server is not currently synchronized to a reliable time source.

Note that by default NTP clients act as servers for those systems in the stratum below them.
Chapter 15. Configuring NTP Using ntpd

Here is a summary of the NTP Strata:

**Stratum 0:**
- Atomic Clocks and their signals broadcast over Radio and GPS
  - GPS (Global Positioning System)
  - Mobile Phone Systems
  - Low Frequency Radio Broadcasts WWVB (Colorado, USA.), JJY-40 and JJY-60 (Japan), DCF77 (Germany), and MSF (United Kingdom)

  These signals can be received by dedicated devices and are usually connected by RS-232 to a system used as an organizational or site-wide time server.

**Stratum 1:**
- Computer with radio clock, GPS clock, or atomic clock attached

**Stratum 2:**
- Reads from stratum 1; Serves to lower strata

**Stratum 3:**
- Reads from stratum 2; Serves to lower strata

**Stratum \( n+1 \):**
- Reads from stratum \( n \); Serves to lower strata

**Stratum 15:**
- Reads from stratum 14; This is the lowest stratum.

This process continues down to Stratum 15 which is the lowest valid stratum. The label Stratum 16 is used to indicated an unsynchronized state.

### 15.3. Understanding NTP


This implementation of NTP enables sub-second accuracy to be achieved. Over the Internet, accuracy to 10s of milliseconds is normal. On a Local Area Network (LAN), 1 ms accuracy is possible under ideal conditions. This is because clock drift is now accounted and corrected for, which was not done in earlier, simpler, time protocol systems. A resolution of 233 picoseconds is provided by using 64-bit timestamps: 32-bits for seconds, 32-bits for fractional seconds.

NTP represents the time as a count of the number of seconds since 00:00 (midnight) 1 January, 1900 GMT. As 32-bits is used to count the seconds, this means the time will “roll over” in 2036. However NTP works on the difference between timestamps so this does not present the same level of problem as other implementations of time protocols have done. If a hardware clock accurate to better than 68 years is available at boot time then NTP will correctly interpret the current date. The NTP4 specification provides for an “Era Number” and an “Era Offset” which can be used to make software more robust.
when dealing with time lengths of more than 68 years. Note, please do not confuse this with the Unix Year 2038 problem.

The NTP protocol provides additional information to improve accuracy. Four timestamps are used to allow the calculation of round-trip time and server response time. In order for a system in its role as NTP client to synchronize with a reference time server, a packet is sent with an “originate timestamp”. When the packet arrives, the time server adds a “receive timestamp”. After processing the request for time and date information and just before returning the packet, it adds a “transmit timestamp”. When the returning packet arrives at the NTP client, a “receive timestamp” is generated. The client can now calculate the total round trip time and by subtracting the processing time derive the actual traveling time. By assuming the outgoing and return trips take equal time, the single-trip delay in receiving the NTP data is calculated. The full NTP algorithm is much more complex then presented here.

Each packet containing time information received is not immediately acted upon, but is subject to validation checks and then used together with several other samples to arrive at a reasonably good estimate of the time. This is then compared to the system clock to determine the time offset, that is to say, the difference between the system clock’s time and what ntpd has determined the time should be. The system clock is adjusted slowly, at most at a rate of 0.5ms per second, to reduce this offset by changing the frequency of the counter being used. It will take at least 2000 seconds to adjust the clock by 1 second using this method. This slow change is referred to as slewing and cannot go backwards. If the time offset of the clock is more than 128ms (the default setting), ntpd can “step” the clock forwards or backwards. If the time offset at system start is greater than 1000 seconds then the user, or an installation script, should make a manual adjustment. See Chapter 3, Configuring the Date and Time. With the -g option to the ntpd command (used by default), any offset at system start will be corrected, but during normal operation only offsets of up to 1000 seconds will be corrected.

Some software may fail or produce an error if the time is changed backwards. For systems that are sensitive to step changes in the time, the threshold can be changed to 600s instead of 128ms using the -x option (unrelated to the -g option). Using the -x option to increase the stepping limit from 0.128s to 600s has a drawback because a different method of controlling the clock has to be used. It disables the kernel clock discipline and may have a negative impact on the clock accuracy. The -x option can be added to the /etc/sysconfig/ntpd configuration file.

15.4. Understanding the Drift File

The drift file is used to store the frequency offset between the system clock running at its nominal frequency and the frequency required to remain in synchronization with UTC. If present, the value contained in the drift file is read at system start and used to correct the clock source. Use of the drift file reduces the time required to achieve a stable and accurate time. The value is calculated, and the drift file replaced, once per hour by ntpd. The drift file is replaced, rather than just updated, and for this reason the drift file must be in a directory for which the ntpd has write permissions.

15.5. UTC, Timezones, and DST

As NTP is entirely in UTC (Universal Time, Coordinated), Timezones and DST (Daylight Saving Time) are applied locally by the system. The file /etc/localtime is a copy of, or symlink to, a zone information file from /usr/share/zoneinfo. The RTC may be in localtime or in UTC, as specified by the 3rd line of /etc/adjtime, which will be one of LOCAL or UTC to indicate how the RTC clock has been set. Users can easily change this setting using the checkbox System Clock Uses UTC in the Date and Time graphical configuration tool. See Chapter 3, Configuring the Date and Time for information on how to use that tool. Running the RTC in UTC is recommended to avoid various problems when daylight saving time is changed.

The operation of ntpd is explained in more detail in the man page ntpd(8). The resources section lists useful sources of information. See Section 15.19, “Additional Resources”. 

317
15.6. Authentication Options for NTP

NTPv4 added support for the Autokey Security Architecture, which is based on public asymmetric cryptography while retaining support for symmetric key cryptography. The Autokey Security Architecture is described in RFC5906 Network Time Protocol Version 4: Autokey Specification. The man page ntp_auth(5) describes the authentication options and commands for ntpd.

An attacker on the network can attempt to disrupt a service by sending NTP packets with incorrect time information. On systems using the public pool of NTP servers, this risk is mitigated by having more than three NTP servers in the list of public NTP servers in /etc/ntp.conf. If only one time source is compromised or spoofed, ntpd will ignore that source. You should conduct a risk assessment and consider the impact of incorrect time on your applications and organization. If you have internal time sources you should consider steps to protect the network over which the NTP packets are distributed.

If you conduct a risk assessment and conclude that the risk is acceptable, and the impact to your applications minimal, then you can choose not to use authentication.

The broadcast and multicast modes require authentication by default. If you have decided to trust the network then you can disable authentication by using disable auth directive in the ntp.conf file. Alternatively, authentication needs to be configured by using SHA1 or MD5 symmetric keys, or by public (asymmetric) key cryptography using the Autokey scheme. The Autokey scheme for asymmetric cryptography is explained in the ntp_auth(8) man page and the generation of keys is explained in ntp-keygen(8). To implement symmetric key cryptography, see Section 15.16.12, “Configuring Symmetric Authentication Using a Key” for an explanation of the key option.

15.7. Managing the Time on Virtual Machines

Virtual machines cannot access a real hardware clock and a virtual clock is not stable enough as the stability is dependent on the host systems work load. For this reason, para-virtualized clocks should be provided by the virtualization application in use. On Fedora with KVM the default clock source is kvm-clock. See the KVM guest timing management chapter of the Virtualization Host Configuration and Guest Installation Guide.

15.8. Understanding Leap Seconds

Greenwich Mean Time (GMT) was derived by measuring the solar day, which is dependent on the Earth’s rotation. When atomic clocks were first made, the potential for more accurate definitions of time became possible. In 1958, International Atomic Time (TAI) was introduced based on the more accurate and very stable atomic clocks. A more accurate astronomical time, Universal Time 1 (UT1), was also introduced to replace GMT. The atomic clocks are in fact far more stable than the rotation of the Earth and so the two times began to drift apart. For this reason UTC was introduced as a practical measure. It is kept within one second of UT1 but to avoid making many small trivial adjustments it was decided to introduce the concept of a leap second in order to reconcile the difference in a manageable way. The difference between UT1 and UTC is monitored until they drift apart by more than half a second. Then only is it deemed necessary to introduce a one second adjustment, forward or backward. Due to the erratic nature of the Earth’s rotational speed, the need for an adjustment cannot be predicted far into the future. The decision as to when to make an adjustment is made by the International Earth Rotation and Reference Systems Service (IERS). However, these announcements are important only to administrators of Stratum 1 servers because NTP transmits information about pending leap seconds and applies them automatically.

3 http://www.rfc-editor.org/info/rfc5906
5 http://www.iers.org
15.9. Understanding the ntpd Configuration File

The daemon, ntpd, reads the configuration file at system start or when the service is restarted. The default location for the file is \texttt{/etc/ntp.conf} and you can view the file by entering the following command:

```bash
\$ less /etc/ntp.conf
```

The configuration commands are explained briefly later in this chapter, see Section 15.16, "Configure NTP", and more verbosely in the ntp.conf(5) man page.

Here follows a brief explanation of the contents of the default configuration file:

### The driftfile entry

A path to the drift file is specified, the default entry on Fedora is:

```plaintext
driftfile /var/lib/ntp/drift
```

If you change this be certain that the directory is writable by ntpd. The file contains one value used to adjust the system clock frequency after every system or service start. See Understanding the Drift File for more information.

### The access control entries

The following lines setup the default access control restrictions:

```plaintext
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery
```

The \texttt{kod} option means a "Kiss-o'-death" packet is to be sent to reduce unwanted queries. The \texttt{nomodify} options prevents any changes to the configuration. The \texttt{notrap} option prevents ntpdc control message protocol traps. The \texttt{nopeer} option prevents a peer association being formed. The \texttt{noquery} option prevents ntpq and ntpdc queries, but not time queries, from being answered. The \texttt{-6} option is required before an IPv6 address.

Addresses within the range 127.0.0/8 range are sometimes required by various processes or applications. As the "restrict default" line above prevents access to everything not explicitly allowed, access to the standard loopback address for IPv4 and IPv6 is permitted by means of the following lines:

```plaintext
# the administrative functions.
restrict 127.0.0.1
restrict -6 ::1
```

Addresses can be added underneath if specifically required by another application. The \texttt{-6} option is required before an IPv6 address.

Hosts on the local network are not permitted because of the "restrict default" line above. To change this, for example to allow hosts from the 192.0.2.0/24 network to query the time and statistics but nothing more, a line in the following format is required:

```plaintext
restrict 192.0.2.0 mask 255.255.255.0 nomodify notrap nopeer
```

To allow unrestricted access from a specific host, for example 192.0.2.250/24, a line in the following format is required:

```plaintext
restrict 192.0.2.250
```
A mask of 255.255.255.255 is applied if none is specified.

The restrict commands are explained in the ntp_acc(5) man page.

The public servers entry

By default, the ntp.conf file contains four public server entries:

```
server 0.fedora.pool.ntp.org iburst
server 1.fedora.pool.ntp.org iburst
server 2.fedora.pool.ntp.org iburst
server 3.fedora.pool.ntp.org iburst
```

The broadcast multicast servers entry

By default, the ntp.conf file contains some commented out examples. These are largely self explanatory. Refer to the explanation of the specific commands Section 15.16, “Configure NTP”. If required, add your commands just below the examples.

**Note**

When the DHCP client program, dhclient, receives a list of NTP servers from the DHCP server, it adds them to ntp.conf and restarts the service. To disable that feature, add PEERNTP=no to /etc/sysconfig/network.

15.10. Understanding the ntpd Sysconfig File

The file will be read by the ntpd init script on service start. The default contents is as follows:

```
# Drop root to id 'ntp:ntp' by default.
OPTIONS="-u ntp:ntp -p /var/run/ntpd.pid -g"
```

The -g option enables ntpd to ignore the offset limit of 1000s and attempt to synchronize the time even if the offset is larger than 1000s, but only on system start. Without that option ntpd will exit if the time offset is greater than 1000s. It will also exit after system start if the service is restarted and the offset is greater than 1000s even with the -g option.

The -p option sets the path to the pid file and -u sets the user and group to which the daemon should drop the root privileges.

15.11. Disabling chrony

In order to use ntpd the default user space daemon, chronyd, must be stopped and disabled. Issue the following command as root:

```
~]$ systemctl stop chronyd
```

To prevent it restarting at system start, issue the following command as root:

```
~]$ systemctl disable chronyd
```

To check the status of chronyd, issue the following command:
15.12. Checking if the NTP Daemon is Installed

To check if `ntpd` is installed, enter the following command as `root`:

```
~]$ systemctl status chronyd
```

NTP is implemented by means of the daemon or service `ntpd`, which is contained within the `ntp` package.

15.13. Installing the NTP Daemon (`ntpd`)

To install `ntpd`, enter the following command as `root`:

```
~]$ yum install ntp
```

To enable `ntpd` at system start, enter the following command as `root`:

```
~]$ systemctl enable ntpd
```

15.14. Checking the Status of NTP

To check if `ntpd` is running and configured to run at system start, issue the following command:

```
~]$ systemctl status ntpd
```

To obtain a brief status report from `ntpd`, issue the following command:

```
~]$ ntpstat
unsynchronised
  time server re-starting
  polling server every 64 s
```

```
~]$ ntpstat
synchronised to NTP server (10.5.26.10) at stratum 2
  time correct to within 52 ms
  polling server every 1024 s
```

15.15. Configure the Firewall to Allow Incoming NTP Packets

The NTP traffic consists of UDP packets on port 123 and needs to be permitted through network and host-based firewalls in order for NTP to function.

Check if the firewall is configured to allow incoming NTP traffic for clients using the graphical **Firewall Configuration** tool.

To start the graphical `firewall-config` tool, press the **Super** key to enter the Activities Overview, type `firewall` and then press **Enter**. The `firewall-config` tool appears. You will be prompted for your user password.
To start the graphical firewall configuration tool using the command line, enter the following command as root user:

```
~]# firewall-config
```

The **Firewall Configuration** window opens. Note, this command can be run as normal user but you will then be prompted for the root password from time to time.

Look for the word “Connected” in the lower left corner. This indicates that the `firewall-config` tool is connected to the user space daemon, `firewalld`.

### 15.15.1. Change the Firewall Settings

To immediately change the current firewall settings, ensure the current view is set to **Runtime Configuration**. Alternatively, to edit the settings to be applied at the next system start, or firewall reload, select **Permanent Configuration** from the drop-down list.

**Note**

When making changes to the firewall settings in **Runtime Configuration** mode, your selection takes immediate effect when you set or clear the check box associated with the service. You should keep this in mind when working on a system that may be in use by other users.

When making changes to the firewall settings in **Permanent Configuration** mode, your selection will only take effect when you reload the firewall or the system restarts. You can use the reload icon below the **File** menu, or click the **Options** menu and select **Reload Firewall**.

### 15.15.2. Open Ports in the Firewall for NTP Packets

To permit traffic through the firewall to a certain port, start the `firewall-config` tool and select the network zone whose settings you want to change. Select the **Ports** tab and the click the **Add** button on the right hand side. The **Port and Protocol** window opens.

Enter the port number **123** and select **udp** from the drop-down list.

### 15.16. Configure NTP

To change the default configuration of the NTP service, use a text editor running as root user to edit the `/etc/ntp.conf` file. This file is installed together with `ntpd` and is configured to use time servers from the Fedora pool by default. The man page `ntp.conf(5)` describes the command options that can be used in the configuration file apart from the access and rate limiting commands which are explained in the `ntp_acc(5)` man page.

### 15.16.1. Configure Access Control to an NTP Service

To restrict or control access to the NTP service running on a system, make use of the **restrict** command in the `ntp.conf` file. See the commented out example:

```
# Hosts on local network are less restricted.
#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap
```
The `restrict` command takes the following form:

```
restrict option
```

where `option` is one or more of:

- **ignore** — All packets will be ignored, including `ntpq` and `ntpd` queries.
- **kod** — a “Kiss-o’-death” packet is to be sent to reduce unwanted queries.
- **limited** — do not respond to time service requests if the packet violates the rate limit default values or those specified by the `discard` command. `ntpq` and `ntpd` queries are not affected. For more information on the `discard` command and the default values, see Section 15.16.2, “Configure Rate Limiting Access to an NTP Service”.
- **lowpriotrap** — traps set by matching hosts to be low priority.
- **nomodify** — prevents any changes to the configuration.
- **noquery** — prevents `ntpq` and `ntpd` queries, but not time queries, from being answered.
- **nopeer** — prevents a peer association being formed.
- **noserve** — deny all packets except `ntpq` and `ntpd` queries.
- **notrap** — prevents `ntpd` control message protocol traps.
- **notrust** — deny packets that are not cryptographically authenticated.
- **ntpport** — modify the match algorithm to only apply the restriction if the source port is the standard NTP UDP port 123.
- **version** — deny packets that do not match the current NTP version.

To configure rate limit access to not respond at all to a query, the respective `restrict` command has to have the `limited` option. If `ntpd` should reply with a KoD packet, the `restrict` command needs to have both `limited` and `kod` options.

### 15.16.2. Configure Rate Limiting Access to an NTP Service

To enable rate limiting access to the NTP service running on a system, add the `limited` option to the `restrict` command as explained in Section 15.16.1, "Configure Access Control to an NTP Service". If you do not want to use the default discard parameters, then also use the `discard` command as explained here.

The `discard` command takes the following form:

```
discard [average value] [minimum value] [monitor value]
```

- **average** — specifies the minimum average packet spacing to be permitted, it accepts an argument in \( \log_2 \) seconds. The default value is 3 (\( 2^3 \) equates to 8 seconds).
- **minimum** — specifies the minimum packet spacing to be permitted, it accepts an argument in \( \log_2 \) seconds. The default value is 1 (\( 2^1 \) equates to 2 seconds).
- **monitor** — specifies the discard probability for packets once the permitted rate limits have been exceeded. The default value is 3000 seconds. This option is intended for servers that receive 1000 or more requests per second.
Examples of the `discard` command are as follows:

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>discard average 4</td>
</tr>
<tr>
<td>discard average 4 minimum 2</td>
</tr>
</tbody>
</table>

### 15.16.3. Adding a Peer Address

To add the address of a peer, that is to say, the address of a server running an NTP service of the same stratum, make use of the `peer` command in the `ntp.conf` file.

The `peer` command takes the following form:

```
peer address
```

where `address` is an IP unicast address or a DNS resolvable name. The address must only be that of a system known to be a member of the same stratum. Peers should have at least one time source that is different to each other. Peers are normally systems under the same administrative control.

### 15.16.4. Adding a Server Address

To add the address of a server, that is to say, the address of a server running an NTP service of a higher stratum, make use of the `server` command in the `ntp.conf` file.

The `server` command takes the following form:

```
server address
```

where `address` is an IP unicast address or a DNS resolvable name. The address of a remote reference server or local reference clock from which packets are to be received.

### 15.16.5. Adding a Broadcast or Multicast Server Address

To add a broadcast or multicast address for sending, that is to say, the address to broadcast or multicast NTP packets to, make use of the `broadcast` command in the `ntp.conf` file.

The broadcast and multicast modes require authentication by default. See Section 15.6, “Authentication Options for NTP”.

The `broadcast` command takes the following form:

```
broadcast address
```

where `address` is an IP broadcast or multicast address to which packets are sent.

This command configures a system to act as an NTP broadcast server. The address used must be a broadcast or a multicast address. Broadcast address implies the IPv4 address 255.255.255.255. By default, routers do not pass broadcast messages. The multicast address can be an IPv4 Class D address, or an IPv6 address. The IANA has assigned IPv4 multicast address 224.0.1.1 and IPv6 address FF05::101 (site local) to NTP. Administratively scoped IPv4 multicast addresses can also be used, as described in RFC 2365 Administratively Scoped IP Multicast.

---

6 http://www.rfc-editor.org/info/rfc2365
15.16.6. Adding a Manycast Client Address

To add a manycast client address, that is to say, to configure a multicast address to be used for NTP server discovery, make use of the `manycastclient` command in the `ntp.conf` file.

The `manycastclient` command takes the following form:

```
manycastclient address
```

where address is an IP multicast address from which packets are to be received. The client will send a request to the address and select the best servers from the responses and ignore other servers. NTP communication then uses unicast associations, as if the discovered NTP servers were listed in `ntp.conf`.

This command configures a system to act as an NTP client. Systems can be both client and server at the same time.

15.16.7. Adding a Broadcast Client Address

To add a broadcast client address, that is to say, to configure a broadcast address to be monitored for broadcast NTP packets, make use of the `broadcastclient` command in the `ntp.conf` file.

The `broadcastclient` command takes the following form:

```
broadcastclient
```

Enables the receiving of broadcast messages. Requires authentication by default. See Section 15.6, “Authentication Options for NTP”.

This command configures a system to act as an NTP client. Systems can be both client and server at the same time.

15.16.8. Adding a Manycast Server Address

To add a manycast server address, that is to say, to configure an address to allow the clients to discover the server by multicasting NTP packets, make use of the `manycastserver` command in the `ntp.conf` file.

The `manycastserver` command takes the following form:

```
manycastserver address
```

Enables the sending of multicast messages. Where address is the address to multicast to. This should be used together with authentication to prevent service disruption.

This command configures a system to act as an NTP server. Systems can be both client and server at the same time.

15.16.9. Adding a Multicast Client Address

To add a multicast client address, that is to say, to configure a multicast address to be monitored for multicast NTP packets, make use of the `multicastclient` command in the `ntp.conf` file.

The `multicastclient` command takes the following form:

```
multicastclient address
```
Chapter 15. Configuring NTP Using ntpd

Enables the receiving of multicast messages. Where address is the address to subscribe to. This should be used together with authentication to prevent service disruption.

This command configures a system to act as an NTP client. Systems can be both client and server at the same time.

15.16.10. Configuring the Burst Option

Using the burst option against a public server is considered abuse. Do not use this option with public NTP servers. Use it only for applications within your own organization.

To increase the average quality of time offset statistics, add the following option to the end of a server command:

```
burst
```

At every poll interval, send a burst of eight packets instead of one, when the server is responding. For use with the server command to improve the average quality of the time offset calculations.

15.16.11. Configuring the iburst Option

To improve the time taken for initial synchronization, add the following option to the end of a server command:

```
iburst
```

At every poll interval, send a burst of eight packets instead of one. When the server is not responding, packets are sent 16s apart. When the server responds, packets are sent every 2s. For use with the server command to improve the time taken for initial synchronization. This is now a default option in the configuration file.

15.16.12. Configuring Symmetric Authentication Using a Key

To configure symmetric authentication using a key, add the following option to the end of a server or peer command:

```
key number
```

where number is in the range 1 to 65534 inclusive. This option enables the use of a message authentication code (MAC) in packets. This option is for use with the peer, server, broadcast, and manycastclient commands.

The option can be used in the /etc/ntp.conf file as follows:

```
server 192.168.1.1 key 10
broadcast 192.168.1.255 key 20
manycastclient 239.255.254.254 key 30
```

See also Section 15.6, “Authentication Options for NTP”.

15.16.13. Configuring the Poll Interval

To change the default poll interval, add the following options to the end of a server or peer command:
Options to change the default poll interval, where the interval in seconds will be calculated by raising 2 to the power of \textit{value}, in other words, the interval is expressed in \textit{log}_2 seconds. The default \texttt{minpoll} value is 6, \textit{2}^6 equates to 64s. The default value for \texttt{maxpoll} is 10, which equates to 1024s. Allowed values are in the range 3 to 17 inclusive, which equates to 8s to 36.4h respectively. These options are for use with the \texttt{peer} or \texttt{server}. Setting a shorter \texttt{maxpoll} may improve clock accuracy.


To specify that a particular server should be preferred above others of similar statistical quality, add the following option to the end of a server or peer command:

\begin{verbatim}
prefer
\end{verbatim}

Use this server for synchronization in preference to other servers of similar statistical quality. This option is for use with the \texttt{peer} or \texttt{server} commands.

**15.16.15. Configuring the Time-to-Live for NTP Packets**

To specify that a particular \texttt{time-to-live} (TTL) value should be used in place of the default, add the following option to the end of a server or peer command:

\begin{verbatim}
ttl \textit{value}
\end{verbatim}

Specify the time-to-live value to be used in packets sent by broadcast servers and multicast NTP servers. Specify the maximum time-to-live value to use for the "expanding ring search" by a manycast client. The default value is \texttt{127}.

**15.16.16. Configuring the NTP Version to Use**

To specify that a particular version of NTP should be used in place of the default, add the following option to the end of a server or peer command:

\begin{verbatim}
version \textit{value}
\end{verbatim}

Specify the version of NTP set in created NTP packets. The value can be in the range \texttt{1} to \texttt{4}. The default is \texttt{4}.

**15.17. Configuring the Hardware Clock Update**

To configure the system clock to update the hardware clock, also known as the real-time clock (RTC), once after executing \texttt{ntpd}, add the following line to \texttt{/etc/sysconfig/ntpd}:

\begin{verbatim}
SYNC_HWCLOCK=yes
\end{verbatim}

To update the hardware clock from the system clock, issue the following command as \texttt{root}:

\begin{verbatim}
~\# hwclock --systohc
\end{verbatim}

When the system clock is being synchronized by \texttt{ntpd}, the kernel will in turn update the RTC every 11 minutes automatically.
15.18. Configuring Clock Sources
To list the available clock sources on your system, issue the following commands:

```
[-] $ cd /sys/devices/system/clocksource/clocksource0/
clocksource0$ cat available_clocksource
kvm-clock tsc hpet acpi_pm
clocksource0$ cat current_clocksource
kvm-clock
```

In the above example, the kernel is using `kvm-clock`. This was selected at boot time as this is a virtual machine.

To override the default clock source, add a line similar to the following in `grub.conf`:

```
clocksource=tsc
```

The available clock source is architecture dependent.

15.19. Additional Resources
The following sources of information provide additional resources regarding NTP and `ntpd`.

15.19.1. Installed Documentation
- `ntpd(8)` man page — Describes `ntpd` in detail, including the command line options.
- `ntp.conf(5)` man page — Contains information on how to configure associations with servers and peers.
- `ntpq(8)` man page — Describes the NTP query utility for monitoring and querying an NTP server.
- `ntpd(8)` man page — Describes the `ntpd` utility for querying and changing the state of `ntpd`.
- `ntp_auth(5)` man page — Describes authentication options, commands, and key management for `ntpd`.
- `ntp_keygen(8)` man page — Describes generating public and private keys for `ntpd`.
- `ntp_acc(5)` man page — Describes access control options using the `restrict` command.
- `ntp_mon(5)` man page — Describes monitoring options for the gathering of statistics.
- `ntp_clock(5)` man page — Describes commands for configuring reference clocks.
- `ntp_misc(5)` man page — Describes miscellaneous options.

15.19.2. Useful Websites
`http://doc.ntp.org/`  
The NTP Documentation Archive

`http://www.eecis.udel.edu/~mills/ntp.html`  
Network Time Synchronization Research Project.

Information on Automatic Server Discovery in NTPv4.
Configuring PTP Using ptp4l

16.1. Introduction to PTP

The Precision Time Protocol (PTP) is a protocol used to synchronize clocks in a network. When used in conjunction with hardware support, PTP is capable of sub-microsecond accuracy, which is far better than is normally obtainable with NTP. PTP support is divided between the kernel and user space. The kernel in Fedora includes support for PTP clocks, which are provided by network drivers. The actual implementation of the protocol is known as linuxptp, a PTPv2 implementation according to the IEEE standard 1588 for Linux.

The linuxptp package includes the ptp4l and phc2sys programs for clock synchronization. The ptp4l program implements the PTP boundary clock and ordinary clock. With hardware time stamping, it is used to synchronize the PTP hardware clock to the master clock, and with software time stamping it synchronizes the system clock to the master clock. The phc2sys program is needed only with hardware time stamping, for synchronizing the system clock to the PTP hardware clock on the network interface card (NIC).

16.1.1. Understanding PTP

The clocks synchronized by PTP are organized in a master-slave hierarchy. The slaves are synchronized to their masters which may be slaves to their own masters. The hierarchy is created and updated automatically by the best master clock (BMC) algorithm, which runs on every clock. When a clock has only one port, it can be master or slave, such a clock is called an ordinary clock (OC). A clock with multiple ports can be master on one port and slave on another, such a clock is called a boundary clock (BC). The top-level master is called the grandmaster clock, which can be synchronized by using a Global Positioning System (GPS) time source. By using a GPS-based time source, disparate networks can be synchronized with a high-degree of accuracy.
Figure 16.1. PTP grandmaster, boundary, and slave Clocks
16.1.2. Advantages of PTP

One of the main advantages that PTP has over the Network Time Protocol (NTP) is hardware support present in various network interface controllers (NIC) and network switches. This specialized hardware allows PTP to account for delays in message transfer, and greatly improves the accuracy of time synchronization. While it is possible to use non-PTP enabled hardware components within the network, this will often cause an increase in jitter or introduce an asymmetry in the delay resulting in synchronization inaccuracies, which add up with multiple non-PTP aware components used in the communication path. To achieve the best possible accuracy, it is recommended that all networking components between PTP clocks are PTP hardware enabled. Time synchronization in larger networks where not all of the networking hardware supports PTP might be better suited for NTP.

With hardware PTP support, the NIC has its own on-board clock, which is used to time stamp the received and transmitted PTP messages. It is this on-board clock that is synchronized to the PTP master, and the computer's system clock is synchronized to the PTP hardware clock on the NIC. With software PTP support, the system clock is used to time stamp the PTP messages and it is synchronized to the PTP master directly. Hardware PTP support provides better accuracy since the NIC can time stamp the PTP packets at the exact moment they are sent and received while software PTP support requires additional processing of the PTP packets by the operating system.

16.2. Using PTP

In order to use PTP, the kernel network driver for the intended interface has to support either software or hardware time stamping capabilities.

16.2.1. Checking for Driver and Hardware Support

In addition to hardware time stamping support being present in the driver, the NIC must also be capable of supporting this functionality in the physical hardware. The best way to verify the time stamping capabilities of a particular driver and NIC is to use the `ethtool` utility to query the interface as follows:

```bash
[-]# ethtool -T em3
Time stamping parameters for em3:
Capabilities:
  hardware-transmit (SOF_TIMESTAMPING_TX_HARDWARE)
  software-transmit (SOF_TIMESTAMPING_TX_SOFTWARE)
  hardware-receive (SOF_TIMESTAMPING_RX_HARDWARE)
  software-receive (SOF_TIMESTAMPING_RX_SOFTWARE)
  software-system-clock (SOF_TIMESTAMPING_SOFTWARE)
  hardware-raw-clock (SOF_TIMESTAMPING_RAW_HARDWARE)
PTP Hardware Clock: 0
Hardware Transmit Timestamp Modes:
  off (HWTSTAMP_TX_OFF)
  on (HWTSTAMP_TX_ON)
Hardware Receive Filter Modes:
  none (HWTSTAMP_FILTER_NONE)
  all (HWTSTAMP_FILTER_ALL)
```

Where `em3` is the interface you wish to check.

For software time stamping support, the parameters list should include:

- **SOF_TIMESTAMPING_SOFTWARE**
- **SOF_TIMESTAMPING_TX_SOFTWARE**
- **SOF_TIMESTAMPING_RX_SOFTWARE**

For hardware time stamping support, the parameters list should include:
• SOF_TIMESTAMPING_RAW_HARDWARE
• SOF_TIMESTAMPING_TX_HARDWARE
• SOF_TIMESTAMPING_RX_HARDWARE

16.2.2. Installing PTP
The kernel in Fedora includes support for PTP. User space support is provided by the tools in the linuxptp package. To install linuxptp, issue the following command as root:

    ~# yum install linuxptp

This will install ptp4l and phc2sys.

Do not run more than one service to set the system clock’s time at the same time. If you intend to serve PTP time using NTP, see Section 16.7, “Serving PTP Time with NTP”.

16.2.3. Starting ptp4l
The ptp4l program tries to use hardware time stamping by default. To use ptp4l with hardware time stamping capable drivers and NICs, you must provide the network interface to use with the -i option. Enter the following command as root:

    ~# ptp4l -i em3 -m

Where em3 is the interface you wish to configure. Below is example output from ptp4l when the PTP clock on the NIC is synchronized to a master:

    ~# ptp4l -i em3 -m
    selected em3 as PTP clock
    port 1: INITIALIZING to LISTENING on INITIALIZE
    port 0: INITIALIZING to LISTENING on INITIALIZE
    port 1: new foreign master 00a069.fffe.0b552d-1
    selected best master clock 00a069.fffe.0b552d
    port 1: LISTENING to UNCALIBRATED on RS_SLAVE
    master offset -23947 s0 freq +0 path delay 11350
    master offset -28867 s0 freq +0 path delay 11236
    master offset -32801 s0 freq +0 path delay 10841
    master offset -37203 s1 freq +0 path delay 10583
    master offset -4775 s2 freq -30575 path delay 10583
    port 1: UNCALIBRATED to SLAVE on MASTER_CLOCK_SELECTED
    master offset -4552 s2 freq -30035 path delay 10385

The master offset value is the measured offset from the master in nanoseconds. The s0, s1, s2 strings indicate the different clock servo states: s0 is unlocked, s1 is clock step and s2 is locked. Once the servo is in the locked state (s2), the clock will not be stepped (only slowly adjusted) unless the pi_offset_const option is set to a positive value in the configuration file (described in the ptp4l(8) man page). The adj value is the frequency adjustment of the clock in parts per billion (ppb). The path delay value is the estimated delay of the synchronization messages sent from the master in nanoseconds. Port 0 is a Unix domain socket used for local PTP management. Port 1 is the em3 interface (based on the example above.) INITIALIZING, LISTENING, UNCALIBRATED and SLAVE are some of possible port states which change on the INITIALIZE, RS_SLAVE, MASTER_CLOCK_SELECTED events. In the last state change message, the port state changed from UNCALIBRATED to SLAVE indicating successful synchronization with a PTP master clock.

The ptp4l program can also be started as a service by running:
Specifying a Configuration File

When running as a service, options are specified in the `/etc/sysconfig/ptp4l` file. More information on the different `ptp4l` options and the configuration file settings can be found in the `ptp4l(8)` man page.

By default, messages are sent to `/var/log/messages`. However, specifying the `-m` option enables logging to standard output which can be useful for debugging purposes.

To enable software time stamping, the `-S` option needs to be used as follows:

```
-]# systemctl start ptp4l
```

```
-]# ptp4l -i eth3 -m -S
```

### 16.2.3.1. Selecting a Delay Measurement Mechanism

There are two different delay measurement mechanisms and they can be selected by means of an option added to the `ptp4l` command as follows:

- `-P`
  - The `-P` selects the *peer-to-peer* (P2P) delay measurement mechanism.

  The P2P mechanism is preferred as it reacts to changes in the network topology faster, and may be more accurate in measuring the delay, than other mechanisms. The P2P mechanism can only be used in topologies where each port exchanges PTP messages with at most one other P2P port. It must be supported and used by all hardware, including transparent clocks, on the communication path.

- `-E`
  - The `-E` selects the *end-to-end* (E2E) delay measurement mechanism. This is the default.

  The E2E mechanism is also referred to as the delay “request-response” mechanism.

- `-A`
  - The `-A` enables automatic selection of the delay measurement mechanism.

  The automatic option starts `ptp4l` in E2E mode. It will change to P2P mode if a peer delay request is received.

**Note**

All clocks on a single PTP communication path must use the same mechanism to measure the delay. A warning will be printed when a peer delay request is received on a port using the E2E mechanism. A warning will be printed when a E2E delay request is received on a port using the P2P mechanism.

### 16.3. Specifying a Configuration File

The command line options and other options, which cannot be set on the command line, can be set in an optional configuration file.

No configuration file is read by default, so it needs to be specified at runtime with the `-f` option. For example:
A configuration file equivalent to the `-i em3 -m -S` options shown above would look as follows:

```bash
[global]
verbose               1
time_stamping         software
[em3]
```

### 16.4. Using the PTP Management Client

The PTP management client, `pmc`, can be used to obtain additional information from `ptp4l` as follows:

```bash
[global]
verbose               1
time_stamping         software
[em3]
```

Setting the `-b` option to zero limits the boundary to the locally running `ptp4l` instance. A larger boundary value will retrieve the information also from PTP nodes further from the local clock. The retrievable information includes:

- **stepsRemoved** is the number of communication paths to the grandmaster clock.
- **offsetFromMaster** and master_offset is the last measured offset of the clock from the master in nanoseconds.
- **meanPathDelay** is the estimated delay of the synchronization messages sent from the master in nanoseconds.
- If **gmPresent** is true, the PTP clock is synchronized to a master, the local clock is not the grandmaster clock.
- **gmIdentity** is the grandmaster's identity.

For a full list of `pmc` commands, type the following as root:

```bash
[global]
verbose               1
time_stamping         software
[em3]
```

Additional information is available in the `[pmc](8)` man page.
16.5. Synchronizing the Clocks

The phc2sys program is used to synchronize the system clock to the PTP hardware clock (PHC) on the NIC. To start phc2sys, where em3 is the interface with the PTP hardware clock, enter the following command as root:

```
~# phc2sys -s em3 -w
```

The -w option waits for the running ptp4l application to synchronize the PTP clock and then retrieves the TAI to UTC offset from ptp4l.

Normally, PTP operates in the *International Atomic Time* (TAI) timescale, while the system clock is kept in *Coordinated Universal Time* (UTC). The current offset between the TAI and UTC timescales is 35 seconds. The offset changes when leap seconds are inserted or deleted, which typically happens every few years. The -O option needs to be used to set this offset manually when the -w is not used, as follows:

```
~# phc2sys -s em3 -O -35
```

Once the phc2sys servo is in a locked state, the clock will not be stepped, unless the -S option is used. This means that the phc2sys program should be started after the ptp4l program has synchronized the PTP hardware clock. However, with -w, it is not necessary to start phc2sys after ptp4l as it will wait for it to synchronize the clock.

The phc2sys program can also be started as a service by running:

```
~# systemctl start phc2sys
```

When running as a service, options are specified in the `/etc/sysconfig/phc2sys` file. More information on the different phc2sys options can be found in the `phc2sys(8)` man page.

Note that the examples in this section assume the command is run on a slave system or slave port.

16.6. Verifying Time Synchronization

When PTP time synchronization is working properly, new messages with offsets and frequency adjustments will be printed periodically to the ptp4l and phc2sys (if hardware time stamping is used) outputs. These values will eventually converge after a short period of time. These messages can be seen in `/var/log/messages` file. An example of the output follows:

```
ptp4l[352.359]: selected /dev/ptp0 as PTP clock
ptp4l[352.361]: port 1: INITIALIZING to LISTENING on INITIALIZE
ptp4l[352.361]: port 0: INITIALIZING to LISTENING on INITIALIZE
ptp4l[353.210]: port 1: new foreign master 00a069.fffe.0b552d-1
ptp4l[357.214]: selected best master clock 00a069.fffe.0b552d
ptp4l[357.214]: port 1: LISTENING to UNCALIBRATED on RS_SLAVE
ptp4l[359.224]: master offset       3304 s0 freq      +0 path delay      9262
ptp4l[360.224]: master offset       3708 s1 freq  -29492 path delay      9262
ptp4l[361.224]: master offset      -3145 s2 freq  -32637 path delay      9262
ptp4l[361.224]: port 1: UNCALIBRATED to SLAVE on MASTER_CLOCK_SELECTED
ptp4l[362.223]: master offset        145 s2 freq  -29580 path delay      9282
ptp4l[363.223]: master offset        263 s2 freq  -29436 path delay      8972
ptp4l[364.223]: master offset        266 s2 freq  -29000 path delay      9153
ptp4l[365.223]: master offset        430 s2 freq  -29656 path delay      9153
ptp4l[366.223]: master offset        615 s2 freq  -29342 path delay      9169
```
Chapter 16. Configuring PTP Using ptp4l

An example of the `phc2sys` output follows:

```
phc2sys[526.527]: Waiting for ptp4l...
phc2sys[527.528]: phc offset 55341 s0 freq +0 delay 2729
phc2sys[528.528]: phc offset 54658 s1 freq -37690 delay 2725
phc2sys[530.528]: phc offset 888 s2 freq -36802 delay 2756
phc2sys[531.528]: phc offset 1156 s2 freq -36287 delay 2766
phc2sys[532.528]: phc offset 411 s2 freq -36666 delay 2738
phc2sys[533.528]: phc offset -73 s2 freq -37026 delay 2764
phc2sys[534.528]: phc offset 95 s2 freq -36869 delay 2733
phc2sys[535.529]: phc offset -359 s2 freq -37294 delay 2738
phc2sys[536.529]: phc offset -257 s2 freq -37300 delay 2753
phc2sys[537.529]: phc offset 119 s2 freq -37001 delay 2745
phc2sys[538.529]: phc offset 288 s2 freq -36796 delay 2766
phc2sys[539.529]: phc offset -149 s2 freq -37147 delay 2760
phc2sys[540.529]: phc offset -352 s2 freq -37395 delay 2771
phc2sys[541.529]: phc offset 166 s2 freq -36982 delay 2748
phc2sys[542.529]: phc offset 58 s2 freq -37048 delay 2756
phc2sys[543.529]: phc offset -31 s2 freq -37114 delay 2748
phc2sys[544.529]: phc offset -333 s2 freq -37426 delay 2747
phc2sys[545.529]: phc offset 194 s2 freq -36999 delay 2749
```

For `ptp4l` there is also a directive, `summary_interval`, to reduce the output and print only statistics, as normally it will print a message every second or so. For example, to reduce the output to every 1024 seconds, add the following line to the `/etc/ptp4l.conf` file:

```
summary_interval 10
```

An example of the `ptp4l` output, with `summary_interval` 6, follows:

```
ptp4l: [615.253] selected /dev/ptp0 as PTP clock
ptp4l: [615.255] port 1: INITIALIZING to LISTENING on INITIALIZE
ptp4l: [615.255] port 0: INITIALIZING to LISTENING on INITIALIZE
ptp4l: [615.564] port 1: new foreign master 00a069.fffe.0b552d-1
ptp4l: [619.574] selected best master clock 00a069.fffe.0b552d
ptp4l: [619.574] port 1: LISTENING to UNCALIBRATED on RS_SLAVE
ptp4l: [623.573] port 1: UNCALIBRATED to SLAVE on MASTER_CLOCK_SELECTED
ptp4l: [684.649] rms 669 max 3691 freq -29383 ± 3735 delay 9232 ± 122
ptp4l: [748.724] rms 253 max 588 freq -29787 ± 221 delay 9219 ± 158
ptp4l: [812.793] rms 287 max 673 freq -29802 ± 248 delay 9211 ± 183
ptp4l: [876.853] rms 226 max 534 freq -29795 ± 197 delay 9221 ± 138
ptp4l: [940.925] rms 250 max 562 freq -29801 ± 218 delay 9199 ± 148
ptp4l: [1004.988] rms 226 max 525 freq -29802 ± 196 delay 9228 ± 143
ptp4l: [1069.065] rms 300 max 646 freq -29802 ± 259 delay 9214 ± 176
ptp4l: [1133.125] rms 226 max 505 freq -29792 ± 197 delay 9225 ± 159
ptp4l: [1197.185] rms 244 max 688 freq -29790 ± 211 delay 9201 ± 162
```

To reduce the output from the `phc2sys`, it can be called it with the `-u` option as follows:

```
~\]# phc2sys -u summary-updates
```
Where `summary-updates` is the number of clock updates to include in summary statistics. An example follows:

```bash
-]# phc2sys -s em3 -w -m -u 60
phc2sys[700,948]: rms 1837 max 10123 freq -36474 ± 4752 delay 2752 ± 16
phc2sys[760,954]: rms 194 max 457 freq -37084 ± 174 delay 2753 ± 12
phc2sys[820,963]: rms 211 max 487 freq -37085 ± 185 delay 2750 ± 19
phc2sys[880,968]: rms 183 max 440 freq -37102 ± 164 delay 2734 ± 91
phc2sys[940,973]: rms 244 max 584 freq -37095 ± 216 delay 2748 ± 16
phc2sys[1000,979]: rms 220 max 573 freq -36666 ± 182 delay 2747 ± 43
phc2sys[1060,984]: rms 266 max 675 freq -36759 ± 234 delay 2753 ± 17
```

### 16.7. Serving PTP Time with NTP

The `ntpd` daemon can be configured to distribute the time from the system clock synchronized by `ptp4l` or `phc2sys` by using the LOCAL reference clock driver. To prevent `ntpd` from adjusting the system clock, the `ntp.conf` file must not specify any NTP servers. The following is a minimal example of `ntp.conf`:

```bash
-]# cat /etc/ntp.conf
server   127.127.1.0
fudge    127.127.1.0 stratum 0
```

**Note**

When the DHCP client program, `dhclient`, receives a list of NTP servers from the DHCP server, it adds them to `ntp.conf` and restarts the service. To disable that feature, add `PEERNTP=no` to `/etc/sysconfig/network`.

### 16.8. Serving NTP Time with PTP

NTP to PTP synchronization in the opposite direction is also possible. When `ntpd` is used to synchronize the system clock, `ptp4l` can be configured with the `priority1` option (or other clock options included in the best master clock algorithm) to be the grandmaster clock and distribute the time from the system clock via PTP:

```bash
-]# cat /etc/ptp4l.conf
[global]
priority1 127
[em3]
# ptp4l -f /etc/ptp4l.conf
```

With hardware time stamping, `phc2sys` needs to be used to synchronize the PTP hardware clock to the system clock:

```bash
-]# phc2sys -c em3 -s CLOCK_REALTIME -w
```

To prevent quick changes in the PTP clock's frequency, the synchronization to the system clock can be loosened by using smaller `P` (proportional) and `I` (integral) constants of the PI servo:

```bash
-]# phc2sys -c em3 -s CLOCK_REALTIME -w -P 0.01 -I 0.0001
```
16.9. Improving Accuracy
Test results indicate that disabling the tickless kernel capability can significantly improve the stability of the system clock, and thus improve the PTP synchronization accuracy (at the cost of increased power consumption). The kernel tickless mode can be disabled by adding `nohz=off` to the kernel boot option parameters.

16.10. Additional Resources
The following sources of information provide additional resources regarding PTP and the `ptp4l` tools.

16.10.1. Installed Documentation
- `ptp4l(8)` man page — Describes `ptp4l` options including the format of the configuration file.
- `pmc(8)` man page — Describes the PTP management client and its command options.
- `phc2sys(8)` man page — Describes a tool for synchronizing the system clock to a PTP hardware clock (PHC).

16.10.2. Useful Websites
http://linuxptp.sourceforge.net/
The Linux PTP project.

http://www.nist.gov/el/isd/ieee/ieee1588.cfm
The IEEE 1588 Standard.
Part V. Monitoring and Automation

This part describes various tools that allow system administrators to monitor system performance, automate system tasks, and report bugs.
System Monitoring Tools

In order to configure the system, system administrators often need to determine the amount of free memory, how much free disk space is available, how the hard drive is partitioned, or what processes are running.

17.1. Viewing System Processes

17.1.1. Using the ps Command

The `ps` command allows you to display information about running processes. It produces a static list, that is, a snapshot of what is running when you execute the command. If you want a constantly updated list of running processes, use the `top` command or the System Monitor application instead.

To list all processes that are currently running on the system including processes owned by other users, type the following at a shell prompt:

```
ps ax
```

For each listed process, the `ps ax` command displays the process ID (`PID`), the terminal that is associated with it (`TTY`), the current status (`STAT`), the cumulated CPU time (`TIME`), and the name of the executable file (`COMMAND`). For example:

```
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.3  53128  2988 ?        Ss   13:28   0:02 /usr/lib/systemd/systemd --system --deserialize 20
root         2  0.0  0.0      0     0 ?        S    13:28   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    13:28   0:00 [ksoftirqd/0]
root         5  0.0  0.0      0     0 ?        S    13:28   0:00 [kworker/u:0]
root         6  0.0  0.0      0     0 ?        S    13:28   0:00 [migration/0]
```

To display the owner alongside each process, use the following command:

```
ps aux
```

Apart from the information provided by the `ps ax` command, `ps aux` displays the effective username of the process owner (`USER`), the percentage of the CPU (`%CPU`) and memory (`%MEM`) usage, the virtual memory size in kilobytes (`VSZ`), the non-swapped physical memory size in kilobytes (`RSS`), and the time or date the process was started. For instance:

```
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.3  53128  2988 ?        Ss   13:28   0:02 /usr/lib/systemd/systemd --system --deserialize 20
root         2  0.0  0.0      0     0 ?        S    13:28   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    13:28   0:00 [ksoftirqd/0]
root         5  0.0  0.0      0     0 ?        S    13:28   0:00 [kworker/u:0]
root         6  0.0  0.0      0     0 ?        S    13:28   0:00 [migration/0]
```

You can also use the `ps` command in a combination with `grep` to see if a particular process is running. For example, to determine if `Emacs` is running, type:

```
ps ax | grep Emacs
```
For a complete list of available command line options, refer to the `ps(1)` manual page.

### 17.1.2. Using the `top` Command

The `top` command displays a real-time list of processes that are running on the system. It also displays additional information about the system uptime, current CPU and memory usage, or total number of running processes, and allows you to perform actions such as sorting the list or killing a process.

To run the `top` command, type the following at a shell prompt:

```
$ top
```

For each listed process, the `top` command displays the process ID (`PID`), the effective username of the process owner (`USER`), the priority (`PR`), the nice value (`NI`), the amount of virtual memory the process uses (`VIRT`), the amount of non-swapped physical memory the process uses (`RES`), the amount of shared memory the process uses (`SHR`), the percentage of the CPU (%CPU) and memory (%MEM) usage, the cumulated CPU time (`TIME+`), and the name of the executable file (`COMMAND`). For example:

```
-] $ top
top - 19:22:08 up  5:53,  3 users,  load average: 1.08, 1.03, 0.82
Tasks: 117 total,   2 running, 115 sleeping,   0 stopped,   0 zombie
Cpu(s):  9.3%us,  1.3%sy,  0.0%ni, 85.1%id,  0.0%wa,  0.0%hi,  0.0%si,  2.6%st
Mem:    761956k total,   617256k used,   144700k free,    24356k buffers
Swap:  1540092k total,    55780k used,  1484312k free,   256408k cached
PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
510 john      20   0 1435m  99m  18m S  9.0 13.3   3:30.52 gnome-shell
32686 root      20   0  156m  27m 3628 R  2.0  3.7   0:48.69 Xorg
2625 john      20   0  488m  27m  14m S  0.3  3.7   0:00.70 emacs
 1 root      20   0 53128 2640 1152 S  0.0  0.3   0:02.83 systemd
 3 root      20   0  53128 2640 1152 S  0.0  0.3   0:02.83 systemd
 2 root      20   0  53128 2640 1152 S  0.0  0.3   0:02.83 systemd
 2 root      20   0  53128 2640 1152 S  0.0  0.3   0:02.83 systemd
 3 root      20   0  53128 2640 1152 S  0.0  0.3   0:02.83 systemd
 5 root      20   0  53128 2640 1152 S  0.0  0.3   0:02.83 systemd
 6 root      20   0  53128 2640 1152 S  0.0  0.3   0:02.83 systemd
 7 root      20   0  53128 2640 1152 S  0.0  0.3   0:02.83 systemd
 8 root      20   0  53128 2640 1152 S  0.0  0.3   0:02.83 systemd
 9 root      20   0  53128 2640 1152 S  0.0  0.3   0:02.83 systemd
10 root      20   0  53128 2640 1152 S  0.0  0.3   0:02.83 systemd
11 root      20   0  53128 2640 1152 S  0.0  0.3   0:02.83 systemd
12 root      20   0  53128 2640 1152 S  0.0  0.3   0:02.83 systemd
13 root      20   0  53128 2640 1152 S  0.0  0.3   0:02.83 systemd
14 root      20   0  53128 2640 1152 S  0.0  0.3   0:02.83 systemd
15 root      20   0  53128 2640 1152 S  0.0  0.3   0:02.83 systemd
```

**Table 17.1. Interactive top commands** contains useful interactive commands that you can use with `top`. For more information, refer to the `top(1)` manual page.

**Table 17.1. Interactive top commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter, Space</td>
<td>Immediately refreshes the display.</td>
</tr>
<tr>
<td>h, ?</td>
<td>Displays a help screen.</td>
</tr>
</tbody>
</table>
### 17.1.3. Using the System Monitor Tool

The **Processes** tab of the **System Monitor** tool allows you to view, search for, change the priority of, and kill processes from the graphical user interface.

To start the **System Monitor** tool, either select **Applications → System Tools → System Monitor** from the **Activities** menu, or type `gnome-system-monitor` at a shell prompt. Then click the **Processes** tab to view the list of running processes.

![Figure 17.1. System Monitor — Processes](image)

For each listed process, the **System Monitor** tool displays its name (**Process Name**), current status (**Status**), percentage of the memory usage (**% CPU**), nice value (**Nice**), process ID (**ID**), memory...
usage (Memory), the channel the process is waiting in (Waiting Channel), and additional details about the session (Session). To sort the information by a specific column in ascending order, click the name of that column. Click the name of the column again to toggle the sort between ascending and descending order.

By default, the System Monitor tool displays a list of processes that are owned by the current user. Selecting various options from the View menu allows you to:

- view only active processes,
- view all processes,
- view your processes,
- view process dependencies,
- view a memory map of a selected process,
- view the files opened by a selected process, and
- refresh the list of processes.

Additionally, various options in the Edit menu allows you to:

- stop a process,
- continue running a stopped process,
- end a process,
- kill a process,
- change the priority of a selected process, and
- edit the System Monitor preferences, such as the refresh interval for the list of processes, or what information to show.

You can also end a process by selecting it from the list and clicking the End Process button.

### 17.2. Viewing Memory Usage

#### 17.2.1. Using the free Command

The `free` command allows you to display the amount of free and used memory on the system. To do so, type the following at a shell prompt:

```bash
free
```

The `free` command provides information about both the physical memory (Mem) and swap space (Swap). It displays the total amount of memory (total), as well as the amount of memory that is in use (used), free (free), shared (shared), in kernel buffers (buffers), and cached (cached). For example:

```bash
~]$ free
```
By default, `free` displays the values in kilobytes. To display the values in megabytes, supply the `-m` command line option:

```
free -m
```

For instance:

```
$ free -m
```

<table>
<thead>
<tr>
<th>Mem:</th>
<th>total</th>
<th>used</th>
<th>free</th>
<th>shared</th>
<th>buffers</th>
<th>cached</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mem:</td>
<td>744</td>
<td>593</td>
<td>150</td>
<td>0</td>
<td>36</td>
<td>152</td>
</tr>
<tr>
<td>+/- buffers/cache:</td>
<td>404</td>
<td>339</td>
<td></td>
<td></td>
<td>36</td>
<td>152</td>
</tr>
</tbody>
</table>

For a complete list of available command line options, refer to the `free(1)` manual page.

### 17.2.2. Using the System Monitor Tool

The **Resources** tab of the **System Monitor** tool allows you to view the amount of free and used memory on the system.

To start the **System Monitor** tool, either select **Applications → System Tools → System Monitor** from the **Activities** menu, or type `gnome-system-monitor` at a shell prompt. Then click the **Resources** tab to view the system's memory usage.
In the Memory and Swap History section, the System Monitor tool displays a graphical representation of the memory and swap usage history, as well as the total amount of the physical memory (Memory) and swap space (Swap) and how much of it is in use.

17.3. Viewing CPU Usage

17.3.1. Using the System Monitor Tool

The Resources tab of the System Monitor tool allows you to view the current CPU usage on the system.

To start the System Monitor tool, either select Applications → System Tools → System Monitor from the Activities menu, or type `gnome-system-monitor` at a shell prompt. Then click the Resources tab to view the system's CPU usage.
In the CPU History section, the System Monitor tool displays a graphical representation of the CPU usage history and shows the percentage of how much CPU is currently in use.

17.4. Viewing Block Devices and File Systems

17.4.1. Using the lsblk Command

The `lsblk` command allows you to display a list of available block devices. To do so, type the following at a shell prompt:

```
lsblk
```

For each listed block device, the `lsblk` command displays the device name (NAME), major and minor device number (MAJ:MIN), if the device is removable (RM), what is its size (SIZE), if the device is read-only (RO), what type is it (TYPE), and where the device is mounted (MOUNTPOINT). For example:

```
$ lsblk

NAME   MAJ:MIN    RM  SIZE RO TYPE MOUNTPOINT
sr0     11:0       0  1024M  0  rom
vda     252:0      0  20G   0  disk
|---vda1 252:1      0  500M   0  part /boot
|---vda2 252:2      0  19.5G  0  part
```
### 17.4.2. Using the blkid Command

The `blkid` command allows you to display information about available block devices. To do so, type the following at a shell prompt as `root`:

```bash
blkid
```

For each listed block device, the `blkid` command displays available attributes such as its **universally unique identifier** (UUID), file system type (**TYPE**), or volume label (**LABEL**). For example:

```
~]$ blkid
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sr0 11:0 1 1024M 0 rom
vda 252:0 0 20G 0 disk
vda1 252:1 0 500M 0 part /boot
vda2 252:2 0 19.5G 0 part
vg_fedora-lv_swap (dm-0) 253:0 0 1.5G 0 lvm [SWAP]
vg_fedora-lv_root (dm-1) 253:1 0 18G 0 lvm /
```

By default, the `lsblk` command lists all available block devices. To display information about a particular device only, specify the device name on the command line:

```
blkid device_name
```

For instance, to display information about `/dev/vda1`, type:

```
~]$ blkid /dev/vda1
/dev/vda1: UUID="4ea24c68-ab10-47d4-8a6b-b8d3a002acba" TYPE="ext4"
```

You can also use the above command with the `-p` and `-o udev` command line options to obtain more detailed information. Note that `root` privileges are required to run this command:

```
blkid -po udev device_name
```
Draft

Using the partx Command

17.4.3. Using the partx Command

The `partx` command allows you to display a list of disk partitions. To list the partition table of a particular disk, as root, run this command with the `-s` option followed by the device name:

```
partx -s device_name
```

For example, to list partitions on `/dev/vda`, type:

```
~# partx -s /dev/vda
NR   START  END          SECTORS  SIZE   NAME   UUID
1     2048  1026047      1024000  500M
2 1026048  41943039   40916992  19.5G
```

For a complete list of available command line options, refer to the `partx(8)` manual page.

17.4.4. Using the findmnt Command

The `findmnt` command allows you to display a list of currently mounted file systems. To do so, type the following at a shell prompt:

```
findmnt
```

For each listed file system, the `findmnt` command displays the target mount point (`TARGET`), source device (`SOURCE`), file system type (`FSTYPE`), and relevant mount options (`OPTIONS`). For example:

```
<table>
<thead>
<tr>
<th>TARGET</th>
<th>SOURCE</th>
<th>FSTYPE</th>
<th>OPTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>/</td>
<td>/dev/mapper/vg_fedora-lv_root</td>
<td>ext4</td>
<td>rw,relatime,seclabel,data=o</td>
</tr>
<tr>
<td></td>
<td>proc</td>
<td>proc</td>
<td>proc</td>
</tr>
<tr>
<td></td>
<td>/proc/sys/fs/binfmt_misc</td>
<td>systemd-1 autosfs</td>
<td>relatime</td>
</tr>
<tr>
<td></td>
<td>/sys</td>
<td>sysfs</td>
<td>sysfs</td>
</tr>
<tr>
<td></td>
<td>/sys/kernel/security</td>
<td>securityfs securityfs</td>
<td>relatime</td>
</tr>
<tr>
<td></td>
<td>/sys/fs/selinux</td>
<td>selinuxfs</td>
<td>selinuxfs</td>
</tr>
<tr>
<td></td>
<td>/sys/fs/cgroup</td>
<td>tmpfs</td>
<td>tmpfs</td>
</tr>
</tbody>
</table>
```

For a complete list of available command line options, refer to the `findmnt(8)` manual page.
By default, `findmnt` lists file systems in a tree-like format. To display the information as an ordinary list, add the `-l` command line option:

```
findmnt -l
```

For instance:

```
TARGET                     SOURCE     FSTYPE   OPTIONS
/proc                      proc       proc     rw,nosuid,nodev,noexec,relatime
/sys                       sysfs      sysfs    rw,nosuid,nodev,noexec,relatime,s
/dev                       devtmpfs   devtmpfs rw,nosuid,seclabel,size=370080k,n
/dev/pts                   devpts     devpts   rw,nosuid,noexec,relatime,seclabe
/dev/shm                   tmpfs      tmpfs    rw,nosuid,nodev,seclabel
/run                       tmpfs      tmpfs    rw,nosuid,nodev,seclabel,mode=755
/                          /dev/mapper/vg_fedora-lv_root ext4   rw,relatime,seclabel,-data=ordered
/sys/kernel/security       securityfs security rw,nosuid,nodev,relatime
/sys/fs/selinux            selinuxfs  selinuxf rw,relatime
/sys/fs/cgroup             tmpfs      tmpfs    rw,nosuid,nodev,seclabel,m
/sys/fs/cgroup/systemd     cgroup     cgroup   rw,nosuid,nodev,exec,relatime,r
[output truncated]
```

You can also choose to list only file systems of a particular type. To do so, add the `-t` command line option followed by a file system type:

```
findmnt -t type
```

For example, to all list `ext4` file systems, type:

```
TARGET SOURCE                        FSTYPE OPTIONS
/      /dev/mapper/vg_fedora-lv_root ext4   rw,relatime,seclabel,-data=ordered
/boot  /dev/vda1                     ext4   rw,relatime,seclabel,-data=ordered
```

For a complete list of available command line options, refer to the `findmnt(8)` manual page.

### 17.4.5. Using the df Command

The `df` command allows you to display a detailed report on the system's disk space usage. To do so, type the following at a shell prompt:

```
df
```
For each listed file system, the `df` command displays its name (`Filesystem`), size (`1K-blocks` or `Size`), how much space is used (`Used`), how much space is still available (`Available`), the percentage of space usage (`Use%`), and where is the file system mounted (`Mounted on`). For example:

```
  ~ $ df
  Filesystem            1K-blocks   Used   Available  Use%  Mounted on
  rootfs                18877356 4605476  14082844  25%         /
  devtmpfs              370080       0    370080   0%         /dev
  tmpfs                  389796     256    380720   1%         /dev/shm
  tmpfs                  389796    3048    377928   1%         /run
  /dev/mapper/vg_fedora-lv_root 18877356 4605476  14082844  25%         /
  tmpfs                  389796       0    389796  0%         /sys/fs/cgroup
  tmpfs                  389796       0    389796  0%         /media
  /dev/vda1              508745   85018    398127  18%         /boot
```

By default, the `df` command 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, supply the `-h` command line option, which causes `df` to display the values in a human-readable format:

```
  ~ $ df -h
  Filesystem      Size  Used   Avail  Use%  Mounted on
  rootfs           19G  4.4G   14G  25%         /
  devtmpfs        362M     0  362M   0%         /dev
  tmpfs            373M  256K  372M   1%         /dev/shm
  tmpfs            373M  3.0M  370M   1%         /run
  /dev/mapper/vg_fedora-lv_root 19G  4.4G   14G  25%         /
  tmpfs            373M    0  373M  0%         /sys/fs/cgroup
  tmpfs            373M    0  373M  0%         /media
  /dev/vda1        497M   84M  389M  18%         /boot
```

Note that the `/dev/shm` entry represents the system's virtual memory file system, `/sys/fs/cgroup` is a cgroup file system, and `/run` contains information about the running system.

For a complete list of available command line options, refer to the `df(1)` manual page.

### 17.4.6. Using the du Command

The `du` command allows you to display the amount of space that is being used by files in a directory. To display the disk usage for each of the subdirectories in the current working directory, run the command with no additional command line options:

```
  ~ $ du
  8   ./gconf/apps/gnome-terminal/profiles/Default
  12  ./gconf/apps/gnome-terminal/profiles
  16  ./gconf/apps/gnome-terminal
  [output truncated]
```
By default, the `du` command displays the disk usage in kilobytes. To view the information in megabytes and gigabytes, supply the `-h` command line option, which causes the utility to display the values in a human-readable format:

```
du -h
```

For instance:

```
~]$ du -h
8.0K    ./.gconf/apps/gnome-terminal/profiles/Default
12K     ./.gconf/apps/gnome-terminal/profiles
16K     ./.gconf/apps/gnome-terminal
[output truncated]
460K    ./.gimp-2.6
68M     .
```

At the end of the list, the `du` command always shows the grand total for the current directory. To display only this information, supply the `-s` command line option:

```
du -sh
```

For example:

```
~]$ du -sh
68M     .
```

For a complete list of available command line options, refer to the `du(1)` manual page.

### 17.4.7. Using the System Monitor Tool

The **File Systems** tab of the **System Monitor** tool allows you to view file systems and disk space usage in the graphical user interface.

To start the **System Monitor** tool, either select **Applications → System Tools → System Monitor** from the **Activities** menu, or type `gnome-system-monitor` at a shell prompt. Then click the **File Systems** tab to view a list of file systems.
For each listed file system, the **System Monitor** tool displays the source device (**Device**), target mount point (**Directory**), and file system type (**Type**), as well as its size (**Total**) and how much space is free (**Free**), available (**Available**), and used (**Used**).

### 17.5. Viewing Hardware Information

#### 17.5.1. Using the `lspci` Command

The `lspci` command lists all PCI devices that are present in the system:

```
lspci
```

For example:

```
$ lspci
00:00.0 Host bridge: Intel Corporation 82X38/X48 Express DRAM Controller
00:01.0 PCI bridge: Intel Corporation 82X38/X48 Express Host-Primary PCI Express Bridge
00:1a.0 USB Controller: Intel Corporation 82801I (ICH9 Family) USB UHCI Controller #4 (rev 02)
00:1a.1 USB Controller: Intel Corporation 82801I (ICH9 Family) USB UHCI Controller #5 (rev 02)
00:1a.2 USB Controller: Intel Corporation 82801I (ICH9 Family) USB UHCI Controller #6 (rev 02)
[output truncated]
```

You can also use the `-v` command line option to display more verbose output, or `-vv` for very verbose output:

```
lspci -v|vv
```

For instance, to determine the manufacturer, model, and memory size of a system's video card, type:

```
$ lspci -v
```

```
01:00.0 VGA compatible controller: nVidia Corporation G84 [Quadro FX 370] (rev a1) (prog-if 0 [VGA controller])
  Subsystem: nVidia Corporation Device 0491
  Physical Slot: 2
  Flags: bus master, fast devsel, latency 0, IRQ 16
  Memory at f2000000 (32-bit, non-prefetchable) [size=16M]
  Memory at e0000000 (64-bit, prefetchable) [size=256M]
  Memory at f0000000 (64-bit, non-prefetchable) [size=32M]
  I/O ports at 1100 [size=128]
  Expansion ROM at <unassigned> [disabled]
  Capabilities: <access denied>
  Kernel driver in use: nouveau
  Kernel modules: nouveau, nvidiafb
[output truncated]
```

For a complete list of available command line options, refer to the `lspci(8)` manual page.

#### 17.5.2. Using the `lsusb` Command
The `lsusb` command allows you to display information about USB buses and devices that are attached to them. To list all USB devices that are in the system, type the following at a shell prompt:

```
lsusb
```

This displays a simple list of devices, for example:

```
~]$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
[output truncated]
Bus 001 Device 002: ID 0bda:0151 Realtek Semiconductor Corp. Mass Storage Device (Multicard Reader)
Bus 008 Device 002: ID 03f0:2c24 Hewlett-Packard Logitech M-UAL-96 Mouse
Bus 008 Device 003: ID 04b3:3025 IBM Corp.
```

You can also use the `-v` command line option to display more verbose output:

```
lsusb -v
```

For instance:

```
~]$ lsusb -v
[output truncated]
Bus 008 Device 002: ID 03f0:2c24 Hewlett-Packard Logitech M-UAL-96 Mouse
Device Descriptor:
 bLength          18
 bDescriptorType  1
 bcdUSB           2.00
 bDeviceClass     0 (Defined at Interface level)
 bDeviceSubClass  0
 bDeviceProtocol  0
 bMaxPacketSize0  8
 idVendor         0x03f0 Hewlett-Packard
 idProduct        0x2c24 Logitech M-UAL-96 Mouse
 bcdDevice        31.00
 iManufacturer    1
 iProduct         2
 iSerial          0
 bNumConfigurations 1
 Configuration Descriptor:
 bLength          9
 bDescriptorType  2
[output truncated]
```

For a complete list of available command line options, refer to the `lsusb(8)` manual page.

### 17.5.3. Using the lspcmcia Command

The `lspcmcia` command allows you to list all PCMCIA devices that are present in the system. To do so, type the following at a shell prompt:

```
lspcmcia
```

For example:

```
~]$ lspcmcia
```

354
You can also use the `-v` command line option to display more verbose information, or `-vv` to increase the verbosity level even further:

```
lsbmui -v
```

For instance:

```
$ lsbmui -v
Socket 0 Bridge: [yenta_cardbus] (bus ID: 0000:15:00.0)
  Configuration: state: on  ready: unknown
```

For a complete list of available command line options, refer to the `pccardctl(8)` manual page.

### 17.5.4. Using the lscpu Command

The `lscpu` command allows you to list information about CPUs that are present in the system, including the number of CPUs, their architecture, vendor, family, model, CPU caches, etc. To do so, type the following at a shell prompt:

```
lscpu
```

For example:

```
$ lscpu
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Byte Order: Little Endian
CPU(s): 4
On-line CPU(s) list: 0-3
Thread(s) per core: 1
Core(s) per socket: 4
Socket(s): 1
NUMA node(s): 1
Vendor ID: GenuineIntel
CPU family: 6
Model: 23
Stepping: 7
CPU MHz: 1998.000
BogoMIPS: 4999.98
Virtualization: VT-x
L1d cache: 32K
L1i cache: 32K
L2 cache: 3072K
NUMA node0 CPU(s): 0-3
```

For a complete list of available command line options, refer to the `lscpu(1)` manual page.

### 17.6. Monitoring Performance with Net-SNMP

Fedora 20 includes the Net-SNMP software suite, which includes a flexible and extensible Simple Network Management Protocol (SNMP) agent. This agent and its associated utilities can be used to provide performance data from a large number of systems to a variety of tools which support polling over the SNMP protocol.
This section provides information on configuring the Net-SNMP agent to securely provide performance data over the network, retrieving the data using the SNMP protocol, and extending the SNMP agent to provide custom performance metrics.

### 17.6.1. Installing Net-SNMP

The Net-SNMP software suite is available as a set of RPM packages in the Fedora software distribution. Table 17.2, “Available Net-SNMP packages” summarizes each of the packages and their contents.

#### Table 17.2. Available Net-SNMP packages

<table>
<thead>
<tr>
<th>Package</th>
<th>Provides</th>
</tr>
</thead>
<tbody>
<tr>
<td>net-snmp</td>
<td>The SNMP Agent Daemon and documentation. This package is required for exporting performance data.</td>
</tr>
<tr>
<td>net-snmp-libs</td>
<td>The netsnmp library and the bundled management information bases (MIBs). This package is required for exporting performance data.</td>
</tr>
<tr>
<td>net-snmp-utils</td>
<td>SNMP clients such as snmpget and snmpwalk. This package is required in order to query a system's performance data over SNMP.</td>
</tr>
<tr>
<td>net-snmp-perl</td>
<td>The mib2c utility and the NetSNMP Perl module.</td>
</tr>
<tr>
<td>net-snmp-python</td>
<td>An SNMP client library for Python.</td>
</tr>
</tbody>
</table>

To install any of these packages, use the `yum` command in the following form:

```
yum install package...
```

For example, to install the SNMP Agent Daemon and SNMP clients used in the rest of this section, type the following at a shell prompt:

```
~$ yum install net-snmp net-snmp-libs net-snmp-utils
```

Note that you must have superuser privileges (that is, you must be logged in as `root`) to run this command. For more information on how to install new packages in Fedora, refer to Section 5.2.4, “Installing Packages”.

### 17.6.2. Running the Net-SNMP Daemon

The `net-snmp` package contains snmpd, the SNMP Agent Daemon. This section provides information on how to start, stop, and restart the snmpd service, and shows how to enable or disable it in the multi-user target unit. For more information on the concept of target units and how to manage system services in Fedora in general, refer to Chapter 7, Services and Daemons.

#### 17.6.2.1. Starting the Service

To run the snmpd service in the current session, type the following at a shell prompt as `root`:

```
systemctl start snmpd.service
```

To configure the service to be automatically started at boot time, use the following command:

```
systemctl enable snmpd.service
```

This will enable the service in the multi-user target unit.
17.6.2.2. Stopping the Service
To stop the running `snmpd` service, type the following at a shell prompt as `root`:

```
systemctl stop snmpd.service
```

To disable starting the service at boot time, use the following command:

```
systemctl disable snmpd.service
```

This will disable the service in the multi-user target unit.

17.6.2.3. Restarting the Service
To restart the running `snmpd` service, type the following at a shell prompt:

```
systemctl restart snmpd.service
```

This will stop the service and start it again in quick succession. To only reload the configuration without stopping the service, run the following command instead:

```
systemctl reload snmpd.service
```

This will cause the running `snmpd` service to reload the configuration.

17.6.3. Configuring Net-SNMP
To change the Net-SNMP Agent Daemon configuration, edit the `/etc/snmp/snmpd.conf` configuration file. The default `snmpd.conf` file shipped with Fedora 20 is heavily commented and serves as a good starting point for agent configuration.

This section focuses on two common tasks: setting system information and configuring authentication. For more information about available configuration directives, refer to the `snmpd.conf(5)` manual page. Additionally, there is a utility in the `net-snmp` package named `snmpconf` which can be used to interactively generate a valid agent configuration.

Note that the `net-snmp-utils` package must be installed in order to use the `snmpwalk` utility described in this section.

### Applying the changes
For any changes to the configuration file to take effect, force the `snmpd` service to re-read the configuration by running the following command as `root`:

```
systemctl reload snmpd.service
```

17.6.3.1. Setting System Information
Net-SNMP provides some rudimentary system information via the system tree. For example, the following `snmpwalk` command shows the system tree with a default agent configuration.
By default, the `sysName` object is set to the hostname. The `sysLocation` and `sysContact` objects can be configured in the `/etc/snmp/snmpd.conf` file by changing the value of the `syslocation` and `syscontact` directives, for example:

```
syslocation Datacenter, Row 3, Rack 2
syscontact UNIX Admin <admin@example.com>
```

After making changes to the configuration file, reload the configuration and test it by running the `snmpwalk` command again:

```
~]# systemctl reload snmpd.service
~]# snmpwalk -v2c -c public localhost system
```

### 17.6.3.2. Configuring Authentication

The Net-SNMP Agent Daemon supports all three versions of the SNMP protocol. The first two versions (1 and 2c) provide for simple authentication using a *community string*. This string is a shared secret between the agent and any client utilities. The string is passed in clear text over the network however and is not considered secure. Version 3 of the SNMP protocol supports user authentication and message encryption using a variety of protocols. The Net-SNMP agent also supports tunneling over SSH, TLS authentication with X.509 certificates, and Kerberos authentication.

#### Configuring SNMP Version 2c Community

To configure an *SNMP version 2c community*, use either the `rocommunity` or `rwcommunity` directive in the `/etc/snmp/snmpd.conf` configuration file. The format of the directives is the following:

```
directive community [source [OID]]
```

... where `community` is the community string to use, `source` is an IP address or subnet, and `OID` is the SNMP tree to provide access to. For example, the following directive provides read-only access to the system tree to a client using the community string "redhat" on the local machine:

```
rocommunity redhat 127.0.0.1 .1.3.6.1.2.1.1
```

To test the configuration, use the `snmpwalk` command with the `-v` and `-c` options.

```
~]# snmpwalk -v2c -c redhat localhost system
```
Configuring SNMP Version 3 User

To configure an **SNMP version 3 user**, use the `net-snmp-create-v3-user` command. This command adds entries to the `/var/lib/net-snmp/snmpd.conf` and `/etc/snmp/snmpd.conf` files which create the user and grant access to the user. Note that the `net-snmp-create-v3-user` command may only be run when the agent is not running. The following example creates the “sysadmin” user with the password “redhatsnmp”:

```
[~]# systemctl stop snmpd.service
[~]# net-snmp-create-v3-user
Enter a SNMPv3 user name to create: admin
Enter authentication pass-phrase: redhatsnmp
Enter encryption pass-phrase: [press return to reuse the authentication pass-phrase]
adding the following line to /var/lib/net-snmp/snmpd.conf:
   createUser admin MD5 "redhatsnmp" DES
adding the following line to /etc/snmp/snmpd.conf:
   rwuser admin
[~]# systemctl start snmpd.service
```

The **rwuser** directive (or **rouser** when the `-ro` command line option is supplied) that `net-snmp-create-v3-user` adds to `/etc/snmp/snmpd.conf` has a similar format to the **rwcommunity** and **rocommunity** directives:

```
directive user [noauth|auth|priv] [OID]
```

... where **user** is a username and **OID** is the SNMP tree to provide access to. By default, the Net-SNMP Agent Daemon allows only authenticated requests (the **auth** option). The **noauth** option allows you to permit unauthenticated requests, and the **priv** option enforces the use of encryption. The **authpriv** option specifies that requests must be authenticated and replies should be encrypted.

For example, the following line grants the user “admin” read-write access to the entire tree:

```
rwuser admin authpriv .1
```

To test the configuration, create a `.snmp` directory in your user's home directory and a configuration file named `snmp.conf` in that directory (`~/.snmp/snmp.conf`) with the following lines:

```
defVersion 3
defSecurityLevel authPriv
defSecurityName admin
defPassphrase redhatsnmp
```

The **snmpwalk** command will now use these authentication settings when querying the agent:

```
[~]$ snmpwalk -v3 localhost system
```
17.6.4. Retrieving Performance Data over SNMP

The Net-SNMP Agent in Fedora provides a wide variety of performance information over the SNMP protocol. In addition, the agent can be queried for a listing of the installed RPM packages on the system, a listing of currently running processes on the system, or the network configuration of the system.

This section provides an overview of OIDs related to performance tuning available over SNMP. It assumes that the `net-snmp-utils` package is installed and that the user is granted access to the SNMP tree as described in Section 17.6.3.2, "Configuring Authentication".

17.6.4.1. Hardware Configuration

The Host Resources MIB included with Net-SNMP presents information about the current hardware and software configuration of a host to a client utility. Table 17.3, "Available OIDs" summarizes the different OIDs available under that MIB.

### Table 17.3. Available OIDs

<table>
<thead>
<tr>
<th>OID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOST-RESOURCES-MIB::hrSystem</td>
<td>Contains general system information such as uptime, number of users, and number of running processes.</td>
</tr>
<tr>
<td>HOST-RESOURCES-MIB::hrStorage</td>
<td>Contains data on memory and file system usage.</td>
</tr>
<tr>
<td>HOST-RESOURCES-MIB::hrDevices</td>
<td>Contains a listing of all processors, network devices, and file systems.</td>
</tr>
<tr>
<td>HOST-RESOURCES-MIB::hrSWRun</td>
<td>Contains a listing of all running processes.</td>
</tr>
<tr>
<td>HOST-RESOURCES-MIB::hrSWRunPerf</td>
<td>Contains memory and CPU statistics on the process table from HOST-RESOURCES-MIB::hrSWRun.</td>
</tr>
<tr>
<td>HOST-RESOURCES-MIB::hrSWInstalled</td>
<td>Contains a listing of the RPM database.</td>
</tr>
</tbody>
</table>

There are also a number of SNMP tables available in the Host Resources MIB which can be used to retrieve a summary of the available information. The following example displays HOST-RESOURCES-MIB::hrFSTable:

```
~$ snmptable -Cb localhost HOST-RESOURCES-MIB::hrFSTable
SNMP table: HOST-RESOURCES-MIB::hrFSTable

Index MountPoint RemoteMountPoint Access Bootable StorageIndex LastFullBackupDate LastPartialBackupDate Type
 1  "/"               "/"               ""                      ""                      ""              HOST-RESOURCES-TYPES::hrFSLinuxExt2
readWrite true 31 0-1-1,0:0:0:0.0 0-1-1,0:0:0:0.0
 5 "/dev/shm"               "/"               ""                      ""                      ""              HOST-RESOURCES-TYPES::hrFSOther
readWrite false 35 0-1-1,0:0:0:0.0 0-1-1,0:0:0:0.0
 6 "/boot"               "/"               ""                      ""                      ""              HOST-RESOURCES-TYPES::hrFSLinuxExt2
readWrite false 36 0-1-1,0:0:0:0.0 0-1-1,0:0:0:0.0
```

For more information about HOST-RESOURCES-MIB, see the `/usr/share/snmp/mibs/HOST-RESOURCES-MIB.txt` file.
17.6.4.2. CPU and Memory Information

Most system performance data is available in the UCD SNMP MIB. The systemStats OID provides a number of counters around processor usage:

```
$ snmpwalk localhost UCD-SNMP-MIB::systemStats
UCD-SNMP-MIB::ssIndex.0 = INTEGER: 1
UCD-SNMP-MIB::ssErrorName.0 = STRING: systemStats
UCD-SNMP-MIB::ssSwapIn.0 = INTEGER: 0 kB
UCD-SNMP-MIB::ssSwapOut.0 = INTEGER: 0 kB
UCD-SNMP-MIB::ssIOSent.0 = INTEGER: 0 blocks/s
UCD-SNMP-MIB::ssIOReceive.0 = INTEGER: 0 blocks/s
UCD-SNMP-MIB::ssSysInterrupts.0 = INTEGER: 29 interrupts/s
UCD-SNMP-MIB::ssSysContext.0 = INTEGER: 18 switches/s
UCD-SNMP-MIB::ssCpuUser.0 = INTEGER: 0
UCD-SNMP-MIB::ssCpuSystem.0 = INTEGER: 0
UCD-SNMP-MIB::ssCpuIdle.0 = INTEGER: 99
UCD-SNMP-MIB::ssCpuRawUser.0 = Counter32: 2278
UCD-SNMP-MIB::ssCpuRawSystem.0 = Counter32: 1395
UCD-SNMP-MIB::ssCpuRawSystem.0 = Counter32: 6826
UCD-SNMP-MIB::ssCpuRawIdle.0 = Counter32: 3383736
UCD-SNMP-MIB::ssCpuRawWait.0 = Counter32: 7629
UCD-SNMP-MIB::ssCpuRawKernel.0 = Counter32: 0
UCD-SNMP-MIB::ssCpuRawInterrupt.0 = Counter32: 434
UCD-SNMP-MIB::ssCpuRawSent.0 = Counter32: 266770
UCD-SNMP-MIB::ssCpuRawReceived.0 = Counter32: 427302
UCD-SNMP-MIB::ssCpuRawInterrupts.0 = Counter32: 743442
UCD-SNMP-MIB::ssCpuRawContexts.0 = Counter32: 718557
UCD-SNMP-MIB::ssCpuRawSoftIRQ.0 = Counter32: 128
UCD-SNMP-MIB::ssCpuRawSwapIn.0 = Counter32: 0
UCD-SNMP-MIB::ssCpuRawSwapOut.0 = Counter32: 0
```

In particular, the ssCpuRawUser, ssCpuRawSystem, ssCpuRawWait, and ssCpuRawIdle OIDs provide counters which are helpful when determining whether a system is spending most of its processor time in kernel space, user space, or I/O. ssRawSwapIn and ssRawSwapOut can be helpful when determining whether a system is suffering from memory exhaustion.

More memory information is available under the UCD-SNMP-MIB::memory OID, which provides similar data to the `free` command:

```
$ snmpwalk localhost UCD-SNMP-MIB::memory
UCD-SNMP-MIB::memIndex.0 = INTEGER: 0
UCD-SNMP-MIB::memErrorName.0 = STRING: swap
UCD-SNMP-MIB::memTotalSwap.0 = INTEGER: 1023992 kB
UCD-SNMP-MIB::memAvailSwap.0 = INTEGER: 1023992 kB
UCD-SNMP-MIB::memTotalReal.0 = INTEGER: 1021588 kB
UCD-SNMP-MIB::memAvailReal.0 = INTEGER: 634260 kB
UCD-SNMP-MIB::memTotalFree.0 = INTEGER: 1658252 kB
UCD-SNMP-MIB::memMinimumSwap.0 = INTEGER: 16000 kB
UCD-SNMP-MIB::memBuffer.0 = INTEGER: 30760 kB
UCD-SNMP-MIB::memCached.0 = INTEGER: 216290 kB
UCD-SNMP-MIB::memSwapError.0 = INTEGER: noError(0)
UCD-SNMP-MIB::memSwapErrorMsg.0 = STRING: 
```

Load averages are also available in the UCD SNMP MIB. The SNMP table UCD-SNMP-MIB::laTable has a listing of the 1, 5, and 15 minute load averages:

```
$ snmptable localhost UCD-SNMP-MIB::laTable
SNMP table: UCD-SNMP-MIB::laTable
   laIndex laNames laLoad laConfig laLoadInt laLoadFloat laErrorFlag laErrMessage
 1  Load-1  0.00  12.00  0  0.000000  noError
 2  Load-5  0.00  12.00  0  0.000000  noError
```
17.6.4.3. File System and Disk Information

The Host Resources MIB provides information on file system size and usage. Each file system (and also each memory pool) has an entry in the HOST-RESOURCES-MIB::hrStorageTable table:

```bash
$ snmptable -Cb localhost HOST-RESOURCES-MIB::hrStorageTable
SNMP table: HOST-RESOURCES-MIB::hrStorageTable

Index AllocationUnits Size Used AllocationFailures
1 HOST-RESOURCES-TYPES::hrStorageRam Physical memory
   1024 Bytes 1021588 388064
2 HOST-RESOURCES-TYPES::hrStorageVirtualMemory Virtual memory
   1024 Bytes 2045580 388064
6 HOST-RESOURCES-TYPES::hrStorageOther Memory buffers
   1024 Bytes 1021588 31048
7 HOST-RESOURCES-TYPES::hrStorageOther Cached memory
   1024 Bytes 216604 216604
10 HOST-RESOURCES-TYPES::hrStorageVirtualMemory Swap space
   1024 Bytes 1023992 0
31 HOST-RESOURCES-TYPES::hrStorageFixedDisk /
   4096 Bytes 2277614 258391
35 HOST-RESOURCES-TYPES::hrStorageFixedDisk /dev/shm
   4096 Bytes 127698 0
36 HOST-RESOURCES-TYPES::hrStorageFixedDisk /boot
   1024 Bytes 198337 26694
```

The OIDs under HOST-RESOURCES-MIB::hrStorageSize and HOST-RESOURCES-MIB::hrStorageUsed can be used to calculate the remaining capacity of each mounted file system.

I/O data is available both in UCD-SNMP-MIB::systemStats (ssIORawSent.0 and ssIORawRecieved.0) and in UCD-DISKIO-MIB::diskIOTable. The latter provides much more granular data. Under this table are OIDs for diskIONReadX and diskIONWrittenX, which provide counters for the number of bytes read from and written to the block device in question since the system boot:

```bash
$ snmptable -Cb localhost UCD-DISKIO-MIB::diskIOTable
SNMP table: UCD-DISKIO-MIB::diskIOTable

Index Device NRead NWritten Reads Writes LA1 LA5 LA15 NReadX NWrittenX
... 25 sda 216886272 139109376 16409 4894 ? ? ? 216886272 139109376
26 sda1 2455552 5120 613 2 ? ? ? 2455552 5120
27 sda2 1486848 0 332 0 ? ? ? 1486848 0
28 sda3 212321280 139104256 15312 4871 ? ? ? 212321280 139104256
```

17.6.4.4. Network Information

Information on network devices is provided by the Interfaces MIB. IF-MIB::ifTable provides an SNMP table with an entry for each interface on the system, the configuration of the interface, and various packet counters for the interface. The following example shows the first few columns of ifTable on a system with two physical network interfaces:

```bash
$ snmptable -Cb localhost IF-MIB::ifTable
SNMP table: IF-MIB::ifTable

Index Descr Type Mtu Speed PhysAddress AdminStatus
1 lo softwareLoopback 16436 10000000 up
```
Network traffic is available under the OIDs IF-MIB::ifOutOctets and IF-MIB::ifInOctets. The following SNMP queries will retrieve network traffic for each of the interfaces on this system:

```
~]$ snmpwalk localhost IF-MIB::ifDescr
IF-MIB::ifDescr.1 = STRING: lo
IF-MIB::ifDescr.2 = STRING: eth0
IF-MIB::ifDescr.3 = STRING: eth1
~]$ snmpwalk localhost IF-MIB::ifOutOctets
IF-MIB::ifOutOctets.1 = Counter32: 10060699
IF-MIB::ifOutOctets.2 = Counter32: 650
IF-MIB::ifOutOctets.3 = Counter32: 0
~]$ snmpwalk localhost IF-MIB::ifInOctets
IF-MIB::ifInOctets.1 = Counter32: 10060699
IF-MIB::ifInOctets.2 = Counter32: 78650
IF-MIB::ifInOctets.3 = Counter32: 0
```

### 17.6.5. Extending Net-SNMP

The Net-SNMP Agent can be extended to provide application metrics in addition to raw system metrics. This allows for capacity planning as well as performance issue troubleshooting. For example, it may be helpful to know that an email system had a 5-minute load average of 15 while being tested, but it is more helpful to know that the email system has a load average of 15 while processing 80,000 messages a second. When application metrics are available via the same interface as the system metrics, this also allows for the visualization of the impact of different load scenarios on system performance (for example, each additional 10,000 messages increases the load average linearly until 100,000).

A number of the applications that ship with Fedora extend the Net-SNMP Agent to provide application metrics over SNMP. There are several ways to extend the agent for custom applications as well. This section describes extending the agent with shell scripts and Perl plug-ins. It assumes that the `net-snmp-utils` and `net-snmp-perl` packages are installed, and that the user is granted access to the SNMP tree as described in Section 17.6.3.2, “Configuring Authentication”.

#### 17.6.5.1. Extending Net-SNMP with Shell Scripts

The Net-SNMP Agent provides an extension MIB (NET-SNMP-EXTEND-MIB) that can be used to query arbitrary shell scripts. To specify the shell script to run, use the `extend` directive in the `/etc/snmp/snmpd.conf` file. Once defined, the Agent will provide the exit code and any output of the command over SNMP. The example below demonstrates this mechanism with a script which determines the number of `httpd` processes in the process table.

```bash
#!/bin/sh
```

Using the proc directive

The Net-SNMP Agent also provides a built-in mechanism for checking the process table via the `proc` directive. Refer to the `snmpd.conf(5)` manual page for more information.

The exit code of the following shell script is the number of `httpd` processes running on the system at a given point in time:

```bash
#!/bin/sh
```
NUMPIDS=`pgrep httpd | wc -l`
exit $NUMPIDS

To make this script available over SNMP, copy the script to a location on the system path, set the executable bit, and add an `extend` directive to the `/etc/snmp/snmpd.conf` file. The format of the `extend` directive is the following:

```
extend name prog args
```

... where `name` is an identifying string for the extension, `prog` is the program to run, and `args` are the arguments to give the program. For instance, if the above shell script is copied to `/usr/local/bin/check_apache.sh`, the following directive will add the script to the SNMP tree:

```
extend httpd_pids /bin/sh /usr/local/bin/check_apache.sh
```

The script can then be queried at NET-SNMP-EXTEND-MIB::nsExtendObjects:

```
~$ snmpwalk localhost NET-SNMP-EXTEND-MIB::nsExtendObjects
NET-SNMP-EXTEND-MIB::nsExtendNumEntries.0 = INTEGER: 1
NET-SNMP-EXTEND-MIB::nsExtendCommand."httpd_pids" = STRING: /bin/sh
NET-SNMP-EXTEND-MIB::nsExtendArgs."httpd_pids" = STRING: /usr/local/bin/check_apache.sh
NET-SNMP-EXTEND-MIB::nsExtendInput."httpd_pids" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendCacheTime."httpd_pids" = INTEGER: 5
NET-SNMP-EXTEND-MIB::nsExtendExecType."httpd_pids" = INTEGER: exec(1)
NET-SNMP-EXTEND-MIB::nsExtendRunType."httpd_pids" = INTEGER: run-on-read(1)
NET-SNMP-EXTEND-MIB::nsExtendStorage."httpd_pids" = INTEGER: permanent(4)
NET-SNMP-EXTEND-MIB::nsExtendStatus."httpd_pids" = INTEGER: active(1)
NET-SNMP-EXTEND-MIB::nsExtendOutput1Line."httpd_pids" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."httpd_pids" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendOutNumLines."httpd_pids" = INTEGER: 1
NET-SNMP-EXTEND-MIB::nsExtendResult."httpd_pids" = INTEGER: 8
```

Note that the exit code ("8" in this example) is provided as an INTEGER type and any output is provided as a STRING type. To expose multiple metrics as integers, supply different arguments to the script using the `extend` directive. For example, the following shell script can be used to determine the number of processes matching an arbitrary string, and will also output a text string giving the number of processes:

```
#!/bin/sh
PATTERN=$1
NUMPIDS=`pgrep $PATTERN | wc -l`
echo "There are $NUMPIDS $PATTERN processes."
exit $NUMPIDS
```

The following `/etc/snmp/snmpd.conf` directives will give both the number of httpd PIDs as well as the number of snmpd PIDs when the above script is copied to `/usr/local/bin/check_proc.sh`:

```
extend httpd_pids /bin/sh /usr/local/bin/check_proc.sh httpd
extend snmpd_pids /bin/sh /usr/local/bin/check_proc.sh snmpd
```

The following example shows the output of an `snmpwalk` of the nsExtendObjects OID:
### Integer exit codes are limited

Integer exit codes are limited to a range of 0–255. For values that are likely to exceed 256, either use the standard output of the script (which will be typed as a string) or a different method of extending the agent.

This last example shows a query for the free memory of the system and the number of `httpd` processes. This query could be used during a performance test to determine the impact of the number of processes on memory pressure:

```bash
~]$ snmpget localhost \
   'NET-SNMP-EXTEND-MIB::nsExtendResult."httpd_pids"' \
   UCD-SNMP-MIB::memAvailReal.0
NET-SNMP-EXTEND-MIB::nsExtendResult."httpd_pids" = INTEGER: 8
UCD-SNMP-MIB::memAvailReal.0 = INTEGER: 799664 kB
```

### 17.6.5.2. Extending Net-SNMP with Perl

Executing shell scripts using the `extend` directive is a fairly limited method for exposing custom application metrics over SNMP. The Net-SNMP Agent also provides an embedded Perl interface for exposing custom objects. The `net-snmp-perl` package provides the `NetSNMP::agent` Perl module that is used to write embedded Perl plug-ins on Fedora.

The `NetSNMP::agent` Perl module provides an `agent` object which is used to handle requests for a part of the agent's OID tree. The `agent` object's constructor has options for running the agent as a sub-agent of `snmpd` or a standalone agent. No arguments are necessary to create an embedded agent:

```perl
use NetSNMP::agent (':all');
my $agent = new NetSNMP::agent();
```

The `agent` object has a `register` method which is used to register a callback function with a particular OID. The `register` function takes a name, OID, and pointer to the callback function. The following example will register a callback function named `hello_handler` with the SNMP Agent which will handle requests under the OID `1.3.6.1.4.1.8072.9999.9999`:

```perl
$agent->register("hello_world", ".1.3.6.1.4.1.8072.9999.9999", \&hello_handler);
```
Obtaining a root OID

The OID .1.3.6.1.4.1.8072.9999.9999 (NET-SNMP-MIB::netSnmpPlaypen) is typically used for demonstration purposes only. If your organization does not already have a root OID, you can obtain one by contacting your Name Registration Authority (ANSI in the United States).

The handler function will be called with four parameters, \texttt{HANDLER}, \texttt{REGISTRATION\_INFO}, \texttt{REQUEST\_INFO}, and \texttt{REQUESTS}. The \texttt{REQUESTS} parameter contains a list of requests in the current call and should be iterated over and populated with data. The \texttt{request} objects in the list have get and set methods which allow for manipulating the OID and value of the request. For example, the following call will set the value of a request object to the string “hello world”:

\begin{verbatim}
$request->setValue(ASN\_OCTET\_STR, "hello world");
\end{verbatim}

The handler function should respond to two types of SNMP requests: the GET request and the GETNEXT request. The type of request is determined by calling the \texttt{getMode} method on the \texttt{REQUEST\_INFO} object passed as the third parameter to the handler function. If the request is a GET request, the caller will expect the handler to set the value of the \texttt{request} object, depending on the OID of the request. If the request is a GETNEXT request, the caller will also expect the handler to set the OID of the request to the next available OID in the tree. This is illustrated in the following code example:

\begin{verbatim}
my $request;
my $string_value = "hello world";
my $integer_value = "8675309";

for($request = $requests; $request; $request = $request->next()) {
    my $oid = $request->getOID();
    if ($request_info->getMode() == MODE\_GET) {
        if ($oid == new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.0")) {
            $request->setValue(ASN\_OCTET\_STR, $string_value);
        } elsif ($oid == new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.1")) {
            $request->setValue(ASN\_INTEGER, $integer_value);
        }
    } elsif ($request_info->getMode() == MODE\_GETNEXT) {
        if ($oid == new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.0")) {
            $request->setOID(.1.3.6.1.4.1.8072.9999.9999.1.1);
            $request->setValue(ASN\_INTEGER, $integer_value);
        } elsif ($oid < new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.0")) {
            $request->setOID(".1.3.6.1.4.1.8072.9999.9999.1.0");
            $request->setValue(ASN\_OCTET\_STR, $string_value);
        }
    }
}
\end{verbatim}

When \texttt{getMode} returns \texttt{MODE\_GET}, the handler analyzes the value of the \texttt{getOID} call on the \texttt{request} object. The value of the \texttt{request} is set to either \texttt{string\_value} if the OID ends in “.1.0”, or set to \texttt{integer\_value} if the OID ends in “.1.1”. If the \texttt{getMode} returns \texttt{MODE\_GETNEXT}, the handler determines whether the OID of the request is “.1.0”, and then sets the OID and value for “.1.1”. If the request is higher on the tree than “.1.0”, the OID and value for “.1.0” is set. This in effect returns the “next” value in the tree so that a program like \texttt{snmpwalk} can traverse the tree without prior knowledge of the structure.
The type of the variable is set using constants from NetSNMP::ASN. See the perldoc for NetSNMP::ASN for a full list of available constants.

The entire code listing for this example Perl plug-in is as follows:

```perl
#!/usr/bin/perl
use NetSNMP::agent (':all');
use NetSNMP::ASN qw(ASN_OCTET_STR ASN_INTEGER);

sub hello_handler {
    my ($handler, $registration_info, $request_info, $requests) = @_;
    my $request;
    my $string_value = "hello world";
    my $integer_value = 8675309;

    for($request = $requests; $request; $request = $request->next()) {
        my $oid = $request->getOID();
        if ($request_info->getMode() == MODE_GET) {
            if ($oid == new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.0")) {
                $request->setValue(ASN_OCTET_STR, $string_value);
            } elsif ($oid == new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.1")) {
                $request->setValue(ASN_INTEGER, $integer_value);
            }
        } elsif ($request_info->getMode() == MODE_GETNEXT) {
            if ($oid == new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.0")) {
                $request->setOID(".1.3.6.1.4.1.8072.9999.9999.1.1");
                $request->setValue(ASN_INTEGER, $integer_value);
            } elsif ($oid < new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.0")) {
                $request->setOID(".1.3.6.1.4.1.8072.9999.9999.1.0");
                $request->setValue(ASN_OCTET_STR, $string_value);
            }
        }
    }

    my $agent = new NetSNMP::agent();
    $agent->register("hello_world", ".1.3.6.1.4.1.8072.9999.9999", 
        \&hello_handler);
}

my $agent = new NetSNMP::agent();
$agent->register("hello_world", ".1.3.6.1.4.1.8072.9999.9999",
    \&hello_handler);
```

To test the plug-in, copy the above program to `/usr/share/snmp/hello_world.pl` and add the following line to the `/etc/snmp/snmpd.conf` configuration file:

```
perl do "/usr/share/snmp/hello_world.pl"
```

The SNMP Agent Daemon will need to be restarted to load the new Perl plug-in. Once it has been restarted, an `snmpwalk` should return the new data:

```
~$ snmpwalk localhost NET-SNMP-MIB::netSnmpPlaypen
NET-SNMP-MIB::netSnmpPlaypen.1.0 = STRING: "hello world"
NET-SNMP-MIB::netSnmpPlaypen.1.1 = INTEGER: 8675309
```

The `snmpget` should also be used to exercise the other mode of the handler:

```
~$ snmpget localhost \
    NET-SNMP-MIB::netSnmpPlaypen.1.0 \
    NET-SNMP-MIB::netSnmpPlaypen.1.1
NET-SNMP-MIB::netSnmpPlaypen.1.0 = STRING: "hello world"
NET-SNMP-MIB::netSnmpPlaypen.1.1 = INTEGER: 8675309
```
17.7. Additional Resources
To learn more about gathering system information, refer to the following resources.

17.7.1. Installed Documentation
• \texttt{ps(1)} — The manual page for the \texttt{ps} command.
• \texttt{top(1)} — The manual page for the \texttt{top} command.
• \texttt{free(1)} — The manual page for the \texttt{free} command.
• \texttt{df(1)} — The manual page for the \texttt{df} command.
• \texttt{du(1)} — The manual page for the \texttt{du} command.
• \texttt{lspci(8)} — The manual page for the \texttt{lspci} command.
• \texttt{snmpd(8)} — The manual page for the \texttt{snmpd} service.
• \texttt{snmpd.conf(5)} — The manual page for the \texttt{/etc/snmp/snmpd.conf} file containing full documentation of available configuration directives.
Viewing and Managing 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 login 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 rsyslogd. A list of log files maintained by rsyslogd can be found in the /etc/rsyslog.conf configuration file.

rsyslog is an enhanced, multi-threaded syslog daemon which replaced the syslogd daemon. rsyslog supports the same functionality as syslogd and extends it with enhanced filtering, encryption protected relaying of messages, various configuration options, or support for transportation via the TCP or UDP protocols. Note that rsyslog is compatible with syslogd.

18.1. Configuring rsyslog

The main configuration file for rsyslog is /etc/rsyslog.conf. It consists of global directives, rules or comments (any empty lines or any text following a hash sign (#)). Both, global directives and rules are extensively described in the sections below.

18.1.1. Global Directives

Global directives specify configuration options that apply to the rsyslogd daemon. They usually specify a value for a specific pre-defined variable that affects the behavior of the rsyslogd daemon or a rule that follows. All of the global directives must start with a dollar sign ($). Only one directive can be specified per line. The following is an example of a global directive that specifies the maximum size of the syslog message queue:

```
$MainMsgQueueSize 50000
```

The default size defined for this directive (10,000 messages) can be overridden by specifying a different value (as shown in the example above).

You may define multiple directives in your /etc/rsyslog.conf configuration file. A directive affects the behavior of all configuration options until another occurrence of that same directive is detected.

A comprehensive list of all available configuration directives and their detailed description can be found in /usr/share/doc/rsyslog-<version-number>/rsyslog_conf_global.html.

18.1.2. Modules

Due to its modular design, rsyslog offers a variety of modules which provide dynamic functionality. Note that modules can be written by third parties. Most modules provide additional inputs (see Input Modules below) or outputs (see Output Modules below). Other modules provide special functionality specific to each module. The modules may provide additional configuration directives that become available after a module is loaded. To load a module, use the following syntax:

```
$ModLoad <MODULE>
```
where \texttt{$\text{ModLoad}$} is the global directive that loads the specified module and \texttt{<MODULE>} represents your desired module. For example, if you want to load the \textbf{Text File Input Module (imfile)}—enables \texttt{rsyslog} to convert any standard text files into syslog messages), specify the following line in your /\texttt{/etc/rsyslog.conf} configuration file:

\begin{verbatim}
$ModLoad imfile
\end{verbatim}

\texttt{rsyslog} offers a number of modules which are split into these main categories:

- Input Modules — Input modules gather messages from various sources. The name of an input module always starts with the \texttt{im} prefix, such as \texttt{imfile, imrelp}, etc.

- Output Modules — Output modules provide a facility to store messages into various targets such as sending them across network, storing them in a database or encrypting them. The name of an output module always starts with the \texttt{om} prefix, such as \texttt{omsnmp, omrelp}, etc.

- Filter Modules — Filter modules provide the ability to filter messages according to specified rules. The name of a filter module always starts with the \texttt{fm} prefix.

- Parser Modules — Parser modules use the message parsers to parse message content of any received messages. The name of a parser module always starts with the \texttt{pm} prefix, such as \texttt{pmrfc5424, pmrfc3164}, etc.

- Message Modification Modules — Message modification modules change the content of a syslog message. The message modification modules only differ in their implementation from the output and filter modules but share the same interface.

- String Generator Modules — String generator modules generate strings based on the message content and strongly cooperate with the template feature provided by \texttt{rsyslog}. For more information on templates, refer to Section 18.1.3.3, "Templates". The name of a string generator module always starts with the \texttt{sm} prefix, such as \texttt{smfile, smtradfile}, etc.

- Library Modules — Library modules generally provide functionality for other loadable modules. These modules are loaded automatically by \texttt{rsyslog} when needed and cannot be configured by the user.

A comprehensive list of all available modules and their detailed description can be found at \url{http://www.rsyslog.com/doc/rsyslog_conf_modules.html}\footnote{\url{http://www.rsyslog.com/doc/rsyslog_conf_modules.html/}}

\begin{important}
\textbf{Make sure you use trustworthy modules only}

Note that when \texttt{rsyslog} loads any modules, it provides them with access to some of its functions and data. This poses a possible security threat. To minimize security risks, use trustworthy modules only.
\end{important}

\subsection*{18.1.3. Rules}

A rule is specified by a \texttt{filter} part, which selects a subset of syslog messages, and an \texttt{action} part, which specifies what to do with the selected messages. To define a rule in your /\texttt{/etc/rsyslog.conf}
configuration file, define both, a filter and an action, on one line and separate them with one or more spaces or tabs. For more information on filters, refer to Section 18.1.3.1, “Filter Conditions” and for information on actions, refer to Section 18.1.3.2, “Actions”.

18.1.3.1. Filter Conditions

rsyslog offers various ways how to filter syslog messages according to various properties. This sections sums up the most used filter conditions.

Facility/Priority-based filters

The most used and well-known way to filter syslog messages is to use the facility/priority-based filters which filter syslog messages based on two conditions: facility and priority. To create a selector, use the following syntax:

\(<\text{FACILITY}>\).\(<\text{PRIORITY}>\)

where:

- \(<\text{FACILITY}>\) specifies the subsystem that produces a specific syslog message. For example, the \text{mail} subsystem handles all mail related syslog messages. \(<\text{FACILITY}>\) can be represented by one of these keywords: \text{auth}, \text{authpriv}, \text{cron}, \text{daemon}, \text{kern}, \text{lpr}, \text{mail}, \text{news}, \text{syslog}, \text{user}, \text{uucp}, and \text{local0} through \text{local7}.

- \(<\text{PRIORITY}>\) specifies a priority of a syslog message. \(<\text{PRIORITY}>\) can be represented by one of these keywords (listed in an ascending order): \text{debug}, \text{info}, \text{notice}, \text{warning}, \text{err}, \text{crit}, \text{alert}, and \text{emerg}.

By preceding any priority with an equal sign (=), you specify that only syslog messages with that priority will be selected. All other priorities will be ignored. Conversely, preceding a priority with an exclamation mark (!) selects all syslog messages but those with the defined priority. By not using either of these two extensions, you specify a selection of syslog messages with the defined or higher priority.

In addition to the keywords specified above, you may also use an asterisk (*) to define all facilities or priorities (depending on where you place the asterisk, before or after the dot). Specifying the keyword \text{none} serves for facilities with no given priorities.

To define multiple facilities and priorities, simply separate them with a comma (,). To define multiple filters on one line, separate them with a semi-colon (;).

The following are a few examples of simple facility/priority-based filters:

```
kern.*       # Selects all kernel syslog messages with any priority

mail.crit    # Selects all mail syslog messages with priority \text{crit} and higher.

cron.!info,!debug     # Selects all cron syslog messages except those with the \text{info} or \text{debug} priority.
```

Property-based filters

Property-based filters let you filter syslog messages by any property, such as \text{timegenerated} or \text{syslogtag}. For more information on properties, refer to Section 18.1.3.3.2, “Properties”. Each
of the properties specified in the filters lets you compare it to a specific value using one of the compare-operations listed in Table 18.1, “Property-based compare-operations”.

Table 18.1. Property-based compare-operations

<table>
<thead>
<tr>
<th>Compare-operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>contains</td>
<td>Checks whether the provided string matches any part of the text provided by the property.</td>
</tr>
<tr>
<td>isequal</td>
<td>Compares the provided string against all of the text provided by the property.</td>
</tr>
<tr>
<td>startswith</td>
<td>Checks whether the provided string matches a prefix of the text provided by the property.</td>
</tr>
<tr>
<td>regex</td>
<td>Compares the provided POSIX BRE (Basic Regular Expression) regular expression against the text provided by the property.</td>
</tr>
<tr>
<td>ereregex</td>
<td>Compares the provided POSIXERE (Extended Regular Expression) regular expression against the text provided by the property.</td>
</tr>
</tbody>
</table>

To define a property-based filter, use the following syntax:

```
:PROPERTY>, [!]<COMPARE_OPERATION>, "<STRING>"
```

where:

- The `<PROPERTY>` attribute specifies the desired property (for example, `timegenerated`, `hostname`, etc.).
- The optional exclamation point (!) negates the output of the compare-operation (if prefixing the compare-operation).
- The `<COMPARE_OPERATION>` attribute specifies one of the compare-operations listed in Table 18.1, “Property-based compare-operations”.
- The `<STRING>` attribute specifies the value that the text provided by the property is compared to. To escape certain character (for example a quotation mark (")), use the backslash character (\).

The following are few examples of property-based filters:

- The following filter selects syslog messages which contain the string `error` in their message text:

  ```
  :msg, contains, "error"
  ```

- The following filter selects syslog messages received from the hostname `host1`:

  ```
  :hostname, isequal, "host1"
  ```

- The following filter selects syslog messages which do not contain any mention of the words `fatal` and `error` with any or no text between them (for example, `fatal lib error`):

  ```
  :msg, !regex, "fatal .* error"
  ```
Expression-based filters

Expression-based filters select syslog messages according to defined arithmetic, boolean or string operations. Expression-based filters use rsyslog's own scripting language. The syntax of this language is defined in /usr/share/doc/rsyslog-<version-number>/rscript_abnf.html along with examples of various expression-based filters.

To define an expression-based filter, use the following syntax:

\[
\text{if } \text{<EXPRESSION>} \text{ then } \text{<ACTION>}
\]

where:

- The <EXPRESSION> attribute represents an expression to be evaluated, for example: $msg \text{ startswith 'DEVNAME'}$ or $\text{syslogfacility-text == 'local0'}$.
- The <ACTION> attribute represents an action to be performed if the expression returns the value true.

### Define an expression-based filter on a single line

When defining an expression-based filter, it must be defined on a single line.

### Do not use regular expressions

Regular expressions are currently not supported in expression-based filters.

BSD-style blocks

rsyslog supports BSD-style blocks inside the /etc/rsyslog.conf configuration file. Each block consists of rules which are preceded with a program or hostname label. Use the '!<PROGRAM>' or '-<PROGRAM>' labels to include or exclude programs, respectively. Use the '+<HOSTNAME>' or '-<HOSTNAME>' labels include or exclude hostnames, respectively.

Example 18.1, “BSD-style block” shows a BSD-style block that saves all messages generated by yum to a file.

#### Example 18.1. BSD-style block

```plaintext
!yum
.*:* /var/log/named.log
```

### 18.1.3.2. Actions

Actions specify what is to be done with the messages filtered out by an already-defined selector. The following are some of the actions you can define in your rule:
Saving syslog messages to log files

The majority of actions specify to which log file a syslog message is saved. This is done by specifying a file path after your already-defined selector. The following is a rule comprised of a selector that selects all cron syslog messages and an action that saves them into the /var/log/cron.log log file:

```
cron.* /var/log/cron.log
```

Use a dash mark (-) as a prefix of the file path you specified if you want to omit syncing the desired log file after every syslog message is generated.

Your specified file path can be either static or dynamic. Static files are represented by a simple file path as was shown in the example above. Dynamic files are represented by a template and a question mark (?) prefix. For more information on templates, refer to Section 18.1.3.3.1, “Generating dynamic file names”.

If the file you specified is an existing tty or /dev/console device, syslog messages are sent to standard output (using special tty-handling) or your console (using special /dev/console-handling) when using the X Window System, respectively.

Sending syslog messages over the network

rsyslog allows you to send and receive syslog messages over the network. This feature allows to administer syslog messages of multiple hosts on one machine. To forward syslog messages to a remote machine, use the following syntax:

```
@([<OPTION>])<HOST>:[<PORT>]
```

where:

- The at sign (@) indicates that the syslog messages are forwarded to a host using the UDP protocol. To use the TCP protocol, use two at signs with no space between them (@@).
- The `<OPTION>` attribute can be replaced with an option such as `z<NUMBER>`. This option enables zlib compression for syslog messages; the `<NUMBER>` attribute specifies the level of compression. To define multiple options, simply separate each one of them with a comma (,).
- The `<HOST>` attribute specifies the host which receives the selected syslog messages.
- The `<PORT>` attribute specifies the host machine's port.

When specifying an IPv6 address as the host, enclose the address in square brackets ([, ]).

The following are some examples of actions that forward syslog messages over the network (note that all actions are preceded with a selector that selects all messages with any priority):

```
.* @192.168.0.1    # Forwards messages to 192.168.0.1 via the UDP protocol

.* @example.com:18    # Forwards messages to "example.com" using port 18 and the TCP protocol

.* @(z9)[2001::1]    # Compresses messages with zlib (level 9 compression) and forwards them to 2001::1 using the UDP protocol
```
Output channels

Output channels are primarily used for log file rotation (for more info on log file rotation, refer to Section 18.2.1, "Configuring logrotate"), that is, to specify the maximum size a log file can grow to. To define an output channel, use the following syntax:

```
$outchannel <NAME>, <FILE_NAME>, <MAX_SIZE>, <ACTION>
```

where:

- The `<NAME>` attribute specifies the name of the output channel.
- The `<FILE_NAME>` attribute specifies the name of the output file.
- The `<MAX_SIZE>` attribute represents the maximum size the specified file (in `<FILE_NAME>`) can grow to. This value is specified in bytes.
- The `<ACTION>` attribute specifies the action that is taken when the maximum size, defined in `<MAX_SIZE>`, is hit.

Example 18.2, "Output channel log rotation" shows a simple log rotation through the use of an output channel. First, the output channel is defined via the `$outchannel` directive and then used in a rule which selects every syslog message with any priority and executes the previously-defined output channel on the acquired syslog messages. Once the limit (in the example 100 MB) is hit, the `/home/joe/log_rotation_script` is executed. This script can contain anything from moving the file into a different folder, editing specific content out of it, or simply removing it.

```
Example 18.2. Output channel log rotation

$outchannel log_rotation,/var/log/test_log.log, 104857600, /home/joe/log_rotation_script

`.*` $log_rotation
```

Support for output channels is to be removed in the future

Output channels are currently supported by rsyslog, however, they are planned to be removed in the nearby future.

Sending syslog messages to specific users

rsyslog can send syslog messages to specific users by simply specifying a username of the user you wish to send the messages to. To specify more than one user, separate each username with a comma (,). To send messages to every user that is currently logged on, use an asterisk (*).

Executing a program

rsyslog lets you execute a program for selected syslog messages and uses the system() call to execute the program in shell. To specify a program to be executed, prefix it with a caret character (^). Consequently, specify a template that formats the received message and passes it to the specified executable as a one line parameter (for more information on templates, refer to Section 18.1.3.3, "Templates"). In the following example, any syslog message with any priority is
selected, formatted with the template template and passed as a parameter to the test-program program, which is then executed with the provided parameter:

```
.* ^test-program;template
```

Be careful when using the shell execute action

When accepting messages from any host, and using the shell execute action, you may be vulnerable to command injection. An attacker may try to inject and execute commands specified by the attacker in the program you specified (in your action) to be executed. To avoid any possible security threats, thoroughly consider the use of the shell execute action.

Inputting syslog messages in a database

Selected syslog messages can be directly written into a database table using the database writer action. The database writer uses the following syntax:

```
:<PLUGIN>:<DB_HOST>,<DB_NAME>,<DB_USER>,<DB_PASSWORD>;<<TEMPLATE>]
```

where:

- The `<PLUGIN>` calls the specified plug-in that handles the database writing (for example, the ommysql plug-in).
- The `<DB_HOST>` attribute specifies the database hostname.
- The `<DB_NAME>` attribute specifies the name of the database.
- The `<DB_USER>` attribute specifies the database user.
- The `<DB_PASSWORD>` attribute specifies the password used with the aforementioned database user.
- The `<TEMPLATE>` attribute specifies an optional use of a template that modifies the syslog message. For more information on templates, refer to Section 18.1.3.3, “Templates”.
Using MySQL and PostgreSQL

Currently, rsyslog provides support for MySQL (for more information, refer to /usr/share/doc/rsyslog-<version-number>/rsyslog_mysql.html) and PostgreSQL databases only. In order to use the MySQL and PostgreSQL database writer functionality, install the rsyslog-mysql and rsyslog-pgsql packages installed, respectively. Also, make sure you load the appropriate modules in your /etc/rsyslog.conf configuration file:

```
$ModLoad ommysql    # Output module for MySQL support
$ModLoad ompgsql    # Output module for PostgreSQL support
```

For more information on rsyslog modules, refer to Section 18.1.2, "Modules".

Alternatively, you may use a generic database interface provided by the omlibdb module. However, this module is currently not compiled.

Discarding syslog messages

To discard your selected messages, use the tilde character (~). The following rule discards any cron syslog messages:

```
cron.* ~
```

For each selector, you are allowed to specify multiple actions. To specify multiple actions for one selector, write each action on a separate line and precede it with an ampersand character (&). Only the first action is allowed to have a selector specified on its line. The following is an example of a rule with multiple actions:

```
kern.=crit joe
& ^test-program;temp
& @192.168.0.1
```

In the example above, all kernel syslog messages with the critical priority (crit) are sent to user joe, processed by the template temp and passed on to the test-program executable, and forwarded to 192.168.0.1 via the UDP protocol.

Specifying multiple actions improves the overall performance of the desired outcome since the specified selector has to be evaluated only once.

Note that any action can be followed by a template that formats the message. To specify a template, suffix an action with a semicolon (;) and specify the name of the template.

Using templates

A template must be defined before it is used in an action, otherwise, it is ignored.

For more information on templates, refer to Section 18.1.3.3, "Templates".
18.1.3.3. Templates
Any output that is generated by rsyslog can be modified and formatted according to your needs through the use of templates. To create a template use the following syntax:

$template <TEMPLATE_NAME>, "text %<PROPERTY>% more text", [<OPTION>]

where:
• $template is the template directive that indicates that the text following it, defines a template.
• <TEMPLATE_NAME> is the name of the template. Use this name to refer to the template.
• Anything between the two quotation marks ("...") is the actual template text. Within this text, you are allowed to escape characters in order to use their functionality, such as \n for new line or \r for carriage return. Other characters, such as % or ", have to be escaped in case you want to those characters literally.

The text specified within two percent signs (%) specifies a property that is consequently replaced with the property's actual value. For more information on properties, refer to Section 18.1.3.3.2, "Properties".

• The <OPTION> attribute specifies any options that modify the template functionality. Do not mistake these for property options, which are defined inside the template text (between "..."). The currently supported template options are sql and stdsql used for formatting the text as an SQL query.

The sql and stdsql options
Note that the database writer (for more information, refer to section Inputting syslog messages in a database in Section 18.1.3.2, "Actions") checks whether the sql and stdsql options are specified in the template. If they are not, the database writer does not perform any action. This is to prevent any possible security threats, such as SQL injection.

18.1.3.3.1. Generating dynamic file names
Templates can be used to generate dynamic file names. By specifying a property as a part of the file path, a new file will be created for each unique property. For example, use the timegenerated property to generate a unique file name for each syslog message:

$template DynamicFile, "/var/log/test_logs/%timegenerated%-test.log"

Keep in mind that the $template directive only specifies the template. You must use it inside a rule for it to take effect:

".*" ?DynamicFile

18.1.3.3.2. Properties
Properties defined inside a template (within two percent signs (%)) allow you to access various contents of a syslog message through the use of a property replacer. To define a property inside a template (between the two quotation marks ("...")), use the following syntax:
where:

- The `<PROPERTY_NAME>` attribute specifies the name of a property. A comprehensible list of all available properties and their detailed description can be found in `/usr/share/doc/rsyslog-<version-number>/property_replacer.html` under the section Available Properties.

- `<FROM_CHAR>` and `<TO_CHAR>` attributes denote a range of characters that the specified property will act upon. Alternatively, regular expressions can be used to specify a range of characters. To do so, specify the letter `R` as the `<FROM_CHAR>` attribute and specify your desired regular expression as the `<TO_CHAR>` attribute.

- The `<OPTION>` attribute specifies any property options. A comprehensible list of all available properties and their detailed description can be found in `/usr/share/doc/rsyslog-<version-number>/property_replacer.html` under the section Property Options.

The following are some examples of simple properties:

- The following property simply obtains the whole message text of a syslog message:

  ```
  %msg%
  ```

- The following property obtains the first two characters of the message text of a syslog message:

  ```
  %msg:1:2%
  ```

- The following property obtains the whole message text of a syslog message and drops its last line feed character:

  ```
  %msg:::drop-last-lf%
  ```

- The following property obtains the first 10 characters of the timestamp that is generated when the syslog message is received and formats it according to the RFC 3999 date standard.

  ```
  %timegenerated:1:10:date-rfc3339%
  ```

### 18.1.3.3.3. Template Examples

This section presents few examples of `rsyslog` templates.

**Example 18.3. “A verbose syslog message template”** shows a template that formats a syslog message so that it outputs the message’s severity, facility, the timestamp of when the message was received, the hostname, the message tag, the message text, and ends with a new line.

**Example 18.3. A verbose syslog message template**

```$template verbose,"%syslogseverity%,%syslogfacility%,%timegenerated%,%HOSTNAME%,%syslogtag %,%msg\n"
```

**Example 18.4. “A wall message template”** shows a template that resembles a traditional wall message (a message that is send to every user that is logged in and has their `mesg(1)` permission set to yes).
This template outputs the message text, along with a hostname, message tag and a timestamp, on a new line (using `\r` and `\n`) and rings the bell (using `\7`).

**Example 18.4. A wall message template**

```bash
$template wallmsg, "\r\n\7Message from syslogd@%HOSTNAME% at %timegenerated% ...\n%syslogtag% %msg%\n"
```

**Example 18.5. "A database formatted message template"** shows a template that formats a syslog message so that it can be used as a database query. Notice the use of the `sql` option at the end of the template specified as the template option. It tells the database writer to format the message as an MySQL SQL query.

**Example 18.5. A database formatted message template**

```bash
$template dbFormat, "insert into SystemEvents (Message, Facility, FromHost, Priority, DeviceReportedTime, ReceivedAt, InfoUnitID, SysLogTag) values ('%msg%', %syslogfacility%, '%HOSTNAME%', %syslogpriority%, '%timereported:::date-mysql%', '%timegenerated:::date-mysql%', %iut%, '%syslogtag%')", sql
```

rsyslog also contains a set of predefined templates identified by the **RSYSLOG_** prefix. It is advisable to not create a template using this prefix to avoid any conflicts. The following list shows these predefined templates along with their definitions.

**RSYSLOG_DebugFormat**

```
"Debug line with all properties:
FROMHOST: '%FROMHOST%', fromhost-ip: '%fromhost-ip%', HOSTNAME: '%HOSTNAME%', PRI: %PRI%, syslogtag '%syslogtag%', programname: '%programname%
, APP-NAME: '%APP-NAME%', PROCID: '%PROCID%', MSGID: '%MSGID%', TIMESTAMP: '%TIMESTAMP%
, STRUCTURED-DATA: '%STRUCTURED-DATA%', msg: '%msg%'
escaped msg: '%msg:::drop-cc%
rawmsg: '%rawmsg%'
"
```

**RSYSLOG_SyslogProtocol23Format**

```
"<%PRI%>1 %TIMESTAMP:::date-rfc3339% %HOSTNAME% %APP-NAME% %PROCID% %MSGID% %STRUCTURED-
DATA% %msg%
"
```

**RSYSLOG_FileFormat**

```
"%TIMESTAMP:::date-rfc3339% %HOSTNAME% %syslogtag%%msg:::sp-if-no-1st-sp%%msg:::drop-
last-lf%
"
```

**RSYSLOG_TraditionalFileFormat**

```
"%TIMESTAMP% %HOSTNAME% %syslogtag%%msg:::sp-if-no-1st-sp%%msg:::drop-last-lf%
"
```

**RSYSLOG_ForwardFormat**

```
"<%PRI%>%TIMESTAMP:::date-rfc3339% %HOSTNAME% %syslogtag:1:32%%msg:::sp-if-no-1st-sp%msg%
"
RSYSLOG_TraditionalForwardFormat

"<%PRI%>%TIMESTAMP% %HOSTNAME% %syslogtag:1:32%%msg:::sp-if-no-1st-sp%%msg%"

18.1.4. rsyslog Command Line Configuration

Some of rsyslog's functionality can be configured through the command line options, as syslogd's can. Note that as of version 3 of rsyslog, this method was deprecated. To enable some of these option, you must specify the compatibility mode rsyslog should run in. However, configuring rsyslog through the command line options should be avoided.

To specify the compatibility mode rsyslog should run in, use the -c option. When no parameter is specified, rsyslog tries to be compatible with syslogd. This is partially achieved by activating configuration directives that modify your configuration accordingly. Therefore, it is advisable to supply this option with a number that matches the major version of rsyslog that is in use and update your /etc/rsyslog.conf configuration file accordingly. If you want to, for example, use syslogd options (which were deprecated in version 3 of rsyslog), you can specify so by executing the following command:

```
~# rsyslogd -c 2
```

Options that are passed to the syslogd daemon, including the backward compatibility mode, can be specified in the /etc/sysconfig/rsyslog configuration file.

For more information on various syslogd options, refer to man syslogd.

18.2. 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 /var/log/ directory with numbers after them (for example, cron-20100906). These numbers represent a timestamp that has been added to a rotated log file. 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.

18.2.1. Configuring logrotate

The following is a sample /etc/logrotate.conf configuration file:

```
# rotate log files weekly
weekly
# keep 4 weeks worth of backlogs
rotate 4
# uncomment this if you want your log files compressed
compress
```

All of the lines in the sample configuration file define global options that apply to every log file. In our example, log files are rotated weekly, rotated log files are kept for the duration of 4 weeks, and all
rotated log files are compressed by gzip into the .gz format. Any lines that begin with a hash sign (#) are comments and are not processed.

You may define configuration options for a specific log file and place it under the global options. However, it is advisable to create a separate configuration file for any specific log file in the /etc/logrotate.d/ directory and define any configuration options there.

The following is an example of a configuration file placed in the /etc/logrotate.d/ directory:

```
/var/log/messages {
    rotate 5
    weekly
    postrotate
    /usr/bin/killall -HUP syslogd
    endscript
}
```

The configuration options in this file are specific for the /var/log/messages log file only. The settings specified here override the global settings where possible. Thus the rotated /var/log/messages log file will be kept for five weeks instead of four weeks as was defined in the global options.

The following is a list of some of the directives you can specify in your logrotate configuration file:

- **weekly** — Specifies the rotation of log files on a weekly basis. Similar directives include:
  - **daily**
  - **monthly**
  - **yearly**

- **compress** — Enables compression of rotated log files. Similar directives include:
  - **nocompress**
  - **compresscmd** — Specifies the command to be used for compressing.
  - **uncompresscmd**
  - **compressext** — Specifies what extension is to be used for compressing.
  - **compressoptions** — Lets you specify any options that may be passed to the used compression program.

- **delaycompress** — Postpones the compression of log files to the next rotation of log files.

- **rotate <INTEGER>** — Specifies the number of rotations a log file undergoes before it is removed or mailed to a specific address. If the value 0 is specified, old log files are removed instead of rotated.

- **mail <ADDRESS>** — This option enables mailing of log files that have been rotated as many times as is defined by the rotate directive to the specified address. Similar directives include:
  - **nomail**
  - **mailfirst** — Specifies that the just-rotated log files are to be mailed, instead of the about-to-expire log files.
• *maillast* — Specifies that the just-rotated log files are to be mailed, instead of the about-to-expire log files. This is the default option when mail is enabled.

For the full list of directives and various configuration options, refer to the `logrotate` man page (`man logrotate`).

### 18.3. 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 Log File Viewer.

#### Installing the gnome-system-log package

In order to use the Log File Viewer, first ensure the `gnome-system-log` package is installed on your system by running, as root:

```sh
yum install gnome-system-log
```

For more information on installing packages with Yum, refer to Section 5.2.4, "Installing Packages".

After you have installed the `gnome-system-log` package, you can open the Log File Viewer by selecting Applications → System Tools → Log File Viewer from the Activities menu, or type the following command at a shell prompt:

```sh
 gnome-system-log
```

The application only displays log files that exist; thus, the list might differ from the one shown in Figure 18.1, "Log File Viewer".
Chapter 18. Viewing and Managing Log Files

Figure 18.1. Log File Viewer

The **Log File Viewer** application lets you filter any existing log file. Click on **Filters** from the menu and select **Manage Filters** to define or edit your desired filter.

Adding or editing a filter lets you define its parameters as is shown in Figure 18.3, “Log File Viewer — defining a filter”.

![Log File Viewer](image)

![Log File Viewer — filters](image)

![Log File Viewer — defining a filter](image)
When defining a filter, you can edit the following parameters:

- **Name** — Specifies the name of the filter.

- **Regular Expression** — Specifies the regular expression that will be applied to the log file and will attempt to match any possible strings of text in it.

- **Effect**
  - **Highlight** — If checked, the found results will be highlighted with the selected color. You may select whether to highlight the background or the foreground of the text.
  - **Hide** — If checked, the found results will be hidden from the log file you are viewing.

When you have at least one filter defined, you may select it from the **Filters** menu and it will automatically search for the strings you have defined in the filter and highlight/hide every successful match in the log file you are currently viewing.
When you check the **Show matches only** option, only the matched strings will be shown in the log file you are currently viewing.

### 18.4. 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 file name of the log file you wish to view. **Figure 18.5, “Log File Viewer — adding a log file”** illustrates the **Open Log** window.

![Figure 18.5. Log File Viewer — adding a log file](image)

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 its contents.

**Reading zipped log files**

The **Log File Viewer** also allows you to open log files zipped in the `.gz` format.

### 18.5. Monitoring Log Files

**Log File 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. **Figure 18.6, “Log File Viewer — new log alert”** illustrates a new alert in the `yum.log` log file and in the `messages` log file. Clicking on the `messages` log file displays the logs in the file with the new lines in bold.
18.6. Additional Resources

To learn more about rsyslog, logrotate, and log files in general, refer to the following resources.

18.6.1. Installed Documentation

- **rsyslogd** manual page — Type `man rsyslogd` to learn more about **rsyslogd** and its many options.

- **rsyslog.conf** manual page — Type `man rsyslog.conf` to learn more about the `/etc/rsyslog.conf` configuration file and its many options.

- `/usr/share/doc/rsyslog-<version-number>/` — After installing the rsyslog package, this directory contains extensive documentation in the html format.

- **logrotate** manual page — Type `man logrotate` to learn more about **logrotate** and its many options.

18.6.2. Useful Websites

- **http://www.rsyslog.com/** — Offers a thorough technical breakdown of rsyslog features, documentation, configuration examples, and video tutorials.

Automating System Tasks

In Linux, tasks, which are also known as jobs, 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**.

19.1. Cron and Anacron

Both, Cron and Anacron, are daemons 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 job is scheduled, it is not executed. Cron allows jobs to be run as often as every minute. Anacron does not assume the system is always on, remembers every scheduled job, and executes it the next time the system is up. However, Anacron can only run a job once a day. To schedule recurring jobs, refer to Section 19.1.2, “Configuring Anacron Jobs” or Section 19.1.3, “Configuring Cron Jobs”. To schedule one-time jobs, refer to Section 19.2, “At and Batch”.

To use the cron service, the **cronie** RPM package must be installed and the **crond** service must be running. **anacron** is a sub-package of **cronie**. To determine if these packages are installed, use the **rpm -q cronie cronie-anacron** command.

19.1.1. Starting and Stopping the Service

To determine if the service is running, use the following command:

```
systemctl is-active crond.service
```

To start the cron service, type the following at a shell prompt as **root**:

```
systemctl start crond.service
```

To stop the service, run the following command as **root**:

```
systemctl stop crond.service
```

It is recommended that you start the service at boot time. To do so, use the following command as **root**:

```
systemctl enable crond.service
```

Refer to **Chapter 7, Services and Daemons** for more information on how to configure services in Fedora.

19.1.2. Configuring Anacron Jobs

The main configuration file to schedule jobs is **/etc/anacrontab** (only **root** is allowed to modify this file), which contains the following lines:
Chapter 19. Automating System Tasks

The first three lines are variables used to configure the environment in which the anacron tasks are run. The SHELL variable tells the system which shell environment to use (in this example the bash shell). The PATH variable defines the path used to execute commands. The output of the anacron jobs are emailed to the username defined with the MAILTO variable. If the MAILTO variable is not defined, (i.e. is empty, MAILTO=), email is not sent.

The next two lines are variables that modify the time for each scheduled job. The RANDOM_DELAY variable denotes the maximum number of minutes that will be added to the delay in minutes variable which is specified for each job. The minimum delay value is set, by default, to 6 minutes. A RANDOM_DELAY set to 12 would therefore add, randomly, between 6 and 12 minutes to the delay in minutes for each job in that particular anacrontab. RANDOM_DELAY can also be set to a value below 6, or even 0. When set to 0, no random delay is added. This proves to be useful when, for example, more computers that share one network connection need to download the same data every day. The START_HOURS_RANGE variable defines an interval (in hours) when scheduled jobs can be run. In case this time interval is missed, for example, due to a power down, then scheduled jobs are not executed that day.

The rest of the lines in the /etc/anacrontab file represent scheduled jobs and have the following format:

<table>
<thead>
<tr>
<th>period in days</th>
<th>delay in minutes</th>
<th>job-identifier</th>
<th>command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>cron.daily</td>
<td>nice run-parts /etc/cron.daily</td>
</tr>
<tr>
<td>7</td>
<td>25</td>
<td>cron.weekly</td>
<td>nice run-parts /etc/cron.weekly</td>
</tr>
<tr>
<td>@monthly</td>
<td>45</td>
<td>cron.monthly</td>
<td>nice run-parts /etc/cron.monthly</td>
</tr>
</tbody>
</table>

19.1.2.1. Examples of Anacron Jobs

The following example shows a simple /etc/anacrontab file:

SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

Any lines that begin with a hash sign (#) are comments and are not processed.
# the maximal random delay added to the base delay of the jobs
RANDOM_DELAY=30
# the jobs will be started during the following hours only
START_HOURS_RANGE=16-20

<table>
<thead>
<tr>
<th>period in days</th>
<th>delay in minutes</th>
<th>job-identifier</th>
<th>command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>dailyjob</td>
<td>nice run-parts /etc/cron.daily</td>
</tr>
<tr>
<td>7</td>
<td>25</td>
<td>weeklyjob</td>
<td>/etc/weeklyjob.bash</td>
</tr>
<tr>
<td>@monthly</td>
<td>45</td>
<td>monthlyjob</td>
<td>ls /proc &gt;&gt; /tmp/proc</td>
</tr>
</tbody>
</table>

All jobs defined in this anacrontab file are randomly delayed by 6-30 minutes and can be executed between 16:00 and 20:00. Thus, the first defined job will run anywhere between 16:26 and 16:50 every day. The command specified for this job will execute all present programs in the /etc/cron.daily directory (using the run-parts script which takes a directory as a command-line argument and sequentially executes every program within that directory). The second specified job will be executed once a week and will execute the weeklyjob.bash script in the /etc directory. The third job is executed once a month and runs a command to write the contents of the /proc to the /tmp/proc file (e.g. ls /proc >> /tmp/proc).

19.1.2.1. Disabling Anacron

In case your system is continuously on and you do not require anacron to run your scheduled jobs, you may uninstall the cronie-anacron package. Thus, you will be able to define jobs using crontabs only.

19.1.3. Configuring Cron Jobs

The configuration file to configure cron jobs, /etc/crontab (only root is allowed to modify this file), contains the following lines:

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# For details see man 4 crontabs
# Example of job definition:
#  .---------------- minute (0 - 59)
#  | .------------- hour (0 - 23)
#  | | .---------- day of month (1 - 31)
#  | | | .------- month (1 - 12) OR jan,feb,mar,apr ...
#  | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
#  | | | | | |
#  | | | | * * * * * user command to be executed
```

The first three lines contain the same variables as an anacrontab file, SHELL, PATH and MAILTO. For more information about these variables, refer to Section 19.1.2, “Configuring Anacron Jobs”. The fourth line contains the HOME variable. The HOME variable can be used to set the home directory to use when executing commands or scripts.

The rest of the lines in the /etc/crontab file represent scheduled jobs and have the following format:

```
minute hour day month day of week user command
```

- **minute** — any integer from 0 to 59
- **hour** — any integer from 0 to 23
Chapter 19. Automating System Tasks

- **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)
- **day of week** — any integer from 0 to 7, where 0 or 7 represents Sunday (or the short name of the week such as sun or mon)
- **user** — specifies the user under which the jobs are run
- **command** — the command to execute (the command can either be a command such as `ls /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 sign (#) are comments and are not processed.

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`. To list the contents of your own personal crontab file, use the `crontab -l` command.

**Do not specify a user**

When using the `crontab` utility, there is no need to specify a user when defining a job.

The `/etc/cron.d/` directory contains files that have the same syntax as the `/etc/crontab` file. Only root is allowed to create and modify files in this directory.
19.1.4. 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 checked each time a user tries to add or delete a cron job.

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.

Access can also be controlled through Pluggable Authentication Modules (PAM). These settings are stored in /etc/security/access.conf. For example, adding the following line in this file forbids creating crontabs for all users except the root user:

```
:*:ALL EXCEPT root :cron
```

The forbidden jobs are logged in an appropriate log file or, when using "crontab -e", returned to the standard output. For more information, refer to access.conf.5 (i.e. man 5 access.conf).

19.1.5. Black/White Listing of Cron Jobs

Black/White listing of jobs is used to omit parts of the defined jobs that do not need to be executed. When calling the run-parts script on a cron folder, such as /etc/cron.daily, we can define which of the programs in this folder will not be executed by run-parts.

To define a black list, create a jobs.deny file in the folder that run-parts will be executing from. For example, if we need to omit a particular program from /etc/cron.daily, then, a file /etc/cron.daily/jobs.deny has to be created. In this file, specify the names of the omitted programs from the same directory. These will not be executed when a command, such as run-parts /etc/cron.daily, is executed by a specific job.

To define a white list, create a jobs.allow file.

The principles of jobs.deny and jobs.allow are the same as those of cron.deny and cron.allow described in section Section 19.1.4, “Controlling Access to Cron”.

19.2. 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 following command:

```
systemctl is-active atd.service
```

### 19.2.1. 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 press **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 press **Ctrl+D**. Alternatively, a shell script can be entered at the prompt, pressing **Enter** after each line in the script, and pressing **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 output, the output is emailed to the user.

Use the command **atq** to view pending jobs. Refer to Section 19.2.3, “Viewing Pending Jobs” for more information.

Usage of the **at** command can be restricted. For more information, refer to Section 19.2.5, “Controlling Access to At and Batch” for details.

### 19.2.2. 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 press **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 press Ctrl+D. Alternatively, a shell script can be entered at the prompt, pressing Enter after each line in the script, and pressing 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 19.2.3, “Viewing Pending Jobs” for more information.

Usage of the batch command can be restricted. For more information, refer to Section 19.2.5, “Controlling Access to At and Batch” for details.

### 19.2.3. 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.

### 19.2.4. Additional Command Line Options

Additional command line options for at and batch include:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f</td>
<td>Read the commands or shell script from a file instead of specifying them at the prompt.</td>
</tr>
<tr>
<td>-m</td>
<td>Send email to the user when the job has been completed.</td>
</tr>
<tr>
<td>-v</td>
<td>Display the time that the job is executed.</td>
</tr>
</tbody>
</table>

### 19.2.5. 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.

### 19.2.6. Starting and Stopping the Service

To start the at service, use the following command as root:

```
systemctl start atd.service
```
To stop the service, as root, type the following at a shell prompt:

```
systemctl stop atd.service
```

It is recommended that you start the service at boot time. To do so, run the following command as root:

```
systemctl enable atd.service
```

Refer to Chapter 7, Services and Daemons for more information on how to configure services in Fedora.

### 19.3. Additional Resources

To learn more about configuring automated tasks, refer to the following resources.

#### 19.3.1. Installed Documentation

- **cron** man page — contains an 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.

- **anacron** man page — contains an overview of anacron.

- **anacrontab** man page — contains an overview of the anacrontab file.

- **/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.
Automatic Bug Reporting Tool (ABRT)

20.1. Overview

For Red Hat Enterprise Linux 6.2, the Automatic Bug Reporting Tool has been upgraded to version 2.0. The ABRT 2-series brings major improvements to automatic bug detection and reporting.

ABRT is the Automatic Bug Reporting Tool. ABRT consists of a daemon, abrtd, which runs silently in the background most of the time. It springs into action when an application crashes, or a kernel oops is detected. The daemon then collects the relevant problem data such as a core file if there is one, the crashing application's command line parameters, and other data of forensic utility.

ABRT currently supports detection of crashes in applications written in the C/C++ and Python languages, as well as kernel oopses.

ABRT is capable of reporting problems to a remote issue tracker. Reporting can be configured to happen automatically whenever an issue is detected, or problem data can be stored locally, reviewed, reported, and deleted manually by a user. The reporting tools can send problem data to a Bugzilla database, a Red Hat Technical Support (RHTSupport) site, upload it using FTP/SCP, email it, or write it to a file.

The part of ABRT which handles already-existing problem data (as opposed to, for example, creation of new problem data) has been factored out into a separate project, libreport. The libreport library provides a generic mechanism for analyzing and reporting problems, and it is used by applications other than ABRT. However, ABRT and libreport operation and configuration is closely integrated. They are therefore discussed as one in this document.

The ABRT packages provide the following crucial components, among others:

- abrtd — The ABRT daemon which runs under root as a background service.
- abrt-applet — The program that receives messages from abrtd and informs you whenever a new problem occurs.
- abrt-gui — The GUI application that shows collected problem data and allows you to further process it.
- abrt-cli — The command line interface, which provides similar functionality to the GUI.
- abr t-ccpp — The ABRT service that provides the C/C++ problems analyzer
- abr t-oops — The ABRT service that provides the kernel oopses analyzer.

20.2. Installing ABRT and Starting its Services

As the first step in order to use ABRT, you should ensure that the abrtdesktop package is installed on your system by running the following command as the root user:
With `abrt-desktop` installed, you will be able to use **ABRT** only in its graphical interface. If you intend to use **ABRT** on the command line, install the `abrt-cli` package:

```bash
~]$ yum install abrt-cli
```

For more information on how to install packages with the **Yum** package manager, refer to Section 5.2.4, "Installing Packages".

Your next step should be to verify that `abrtd` is running. The daemon is typically configured to start up at boot time. You can use the following command as root to verify its current status:

```bash
~]$ service abrtd status
abrtd (pid 1535) is running...
```

If the `service` command returns the `abrtd is stopped` message, the daemon is not running. It can be started for the current session by entering this command:

```bash
~]$ service abrtd start
Starting abrt daemon: [ OK ]
```

You can run the following `chkconfig` command to ensure that the `abrtd` service initializes every time the system starts up:

```bash
~]$ chkconfig abrtd on
```

Similarly, you can follow the same steps to check and configure the `abrt-ccpp` service if you want **ABRT** to catch C/C++ crashes. To set **ABRT** to detect kernel oopses, use the same steps for the `abrt-oops` service. Note that this service cannot catch kernel oopses which cause the system to fail to become unresponsive or to reboot immediately.

Finally, you can verify that the **ABRT** notification applet is running:

```bash
~]$ ps -el | grep abrt-applet
  S   500  2036  1824  0 80  0 -  61604 poll_s ?  00:00:00 abrt-applet
```

If the **ABRT** notification applet is not running, you can start it manually in your current desktop session by running the `abrt-applet` program:

```bash
~]$ abrt-applet &
[1] 2261
```

The applet can be configured to start automatically when your graphical desktop session starts. For example, on the GNOME desktop this can be achieved by accessing the **System → Preferences → Startup Applications** menu and ensuring that the **ABRT** notification applet is added to the list of programs and selected to run on at system startup.
20.3. Running ABRT

Whenever a problem is detected, ABRT compares it with all existing problem data and determines whether that same problem has been recorded. If it has been, the existing problem data is updated and the most recent (duplicate) problem is not recorded again. If this problem is not recognized by
ABRT, a problem data directory is created. A problem data directory typically consists of files such as:

- analyzer
- architecture
- coredump
- cmdline
- executable
- kernel
- os_release
- reason
- time
- uid

Other files, such as backtrace, can be created during analysis depending on which analyzer method is used and its configuration settings. Each of these files holds specific information about the system and the problem itself. For example, the kernel file records the version of the crashed kernel.

### 20.3.1. Using the Graphical User Interface

The ABRT daemon sends a broadcast D-Bus message whenever a problem report is created. If the ABRT notification applet is running, it catches this message and displays an orange alarm icon in the Notification Area. You can open the ABRT GUI application using this icon. As an alternative, you can display the ABRT GUI by selecting the Application → System Tools → Automatic Bug Reporting Tool menu item.
Alternatively, you can run the ABRT GUI from the command line as follows:

```
~]$ abrt-gui &
```

The ABRT GUI provides an easy and intuitive way of viewing, reporting and deleting of reported problems. The ABRT window displays a list of detected problems. Each problem entry consists of the name of the failing application, the reason why the application crashed, and the date of the last occurrence of the problem.
Figure 20.4. An example of running **ABRT GUI**.

If you double-click on a problem report line, you can access the detailed problem description and proceed with the process of determining how the problem should be analyzed, and where it should be reported.

![Problem description](image)

Figure 20.5. A detailed problem data example.

You are first asked to provide additional information about the problem which occurred. You should provide detailed information on how the problem happened and what steps should be done in order
to reproduce it. In the next steps, choose how the problem will be analyzed and generate a backtrace depending on your configuration. You can skip the analysis and backtrace-generation steps but remember that developers need as much information about the problem as possible. You can always modify the backtrace and remove any sensitive information you do not want to provide before you send the problem data out.

Figure 20.6. Selecting how to analyze the problem.

Figure 20.7. ABRT analyzing the problem

Next, choose how you want to report the issue. If you are using Red Hat Enterprise Linux, Red Hat Customer Support is the preferred choice.
Chapter 20. Automatic Bug Reporting Tool (ABRT)

Figure 20.8. Selecting a problem reporter.

If you choose to report to Red Hat Customer Support, and you have not configured this event yet, you will be warned that this event is not configured properly and you will be offered an option to do so.

Figure 20.9. Warning - missing Red Hat Customer Support configuration.

Here, you need to provide your Red Hat login information (Refer to Section 20.4.3, “Event Configuration in ABRT GUI” for more information on how to acquire it and how to set this event.), otherwise you will fail to report the problem.

Figure 20.10. Red Hat Customer Support configuration window.
After you have chosen a reporting method and have it set up correctly, review the backtrace and confirm the data to be reported.

![Review the backtrace](image)

**Figure 20.11. Reviewing the problem backtrace.**

Finally, the problem data is sent to the chosen destination, and you can now decide whether to continue with reporting the problem using another available method or finish your work on this problem. If you have reported your problem to the Red Hat Customer Support database, a problem case is filed in the database. From now on, you will be informed about the problem resolution progress via email you provided during the process of reporting. You can also oversee the problem case.
using the URL that is provided to you by ABRT GUI when the problem case is created, or via emails received from Red Hat Support.

![Image of Reporting window]

Figure 20.13. Problem is being reported to the Red Hat Customer Support database.

20.3.2. Using the Command Line Interface

Problem data saved by abrtd can be viewed, reported, and deleted using the command line interface.

General usage of the abrt-cli tool can be described using the following syntax:

```
abrt-cli [-version] <command> [<args>]
```

...where `<args>` stands for a problem data directory and/or options modifying the commands, and `<command>` is one of the following sub-commands:

- **list** — lists problems and views the problem data.
- **report** — analyzes and reports problems.
- **rm** — removes unneeded problems.
- **info** — provides information about a particular problem.

To display help on particular abrt-cli command use:

```
abrt-cli <command> --help
```

The rest of the commands used with abrt-cli are described in the following sections.

20.3.2.1. Viewing Problems

To view detected problems, enter the abrt-cli list command:


- ]# abrt-cli list  
  count: 2  
  executable: /usr/bin/gdb  
  package: gdb-7.2-48.el6  
  time: Tue 13 Sep 2011 10:18:14 AM CEST  
  uid: 500  

  count: 1  
  executable: /bin/bash  
  package: bash-4.1.2-8.el6  
  time: Wed 21 Sep 2011 06:18:07 PM CEST  
  uid: 500  

- **Directory** — Shows the problem data directory that contains all information about the problem.  
- **count** — Shows how many times this particular problem occurred.  
- **executable** — Indicates which binary or executable script crashed.  
- **package** — Shows the name of the package that contains the program that caused the problem.  
- **time** — Shows the date and time of the last occurrence of the problem.  
- **uid** — Shows the ID of the user which ran the program that crashed.  

The following table shows options available with the `abrt-cli list` command. All options are mutually inclusive so you can combine them according to your need. The command output will be the most comprehensive if you combine all options, and you will receive the least details if you use no additional options.  

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>With no additional option, the <code>abrt-cli list</code> command displays only basic information for problems that have not been reported yet.</td>
</tr>
<tr>
<td><strong>-d, --detailed</strong></td>
<td>Displays all stored information about problems listed, including a backtrace if it has already been generated.</td>
</tr>
<tr>
<td><strong>-f, --full</strong></td>
<td>Displays basic information for all problems including the already-reported ones.</td>
</tr>
<tr>
<td><strong>-v, --verbose</strong></td>
<td>Provides additional information on its actions.</td>
</tr>
</tbody>
</table>

If you want to view information just about one particular problem, you can use the command:  

- `abrt-cli info <DIR>`  

...where `<DIR>` stands for the **problem data directory** of the problem that is being viewed. The following table shows options available with the `abrt-cli info` command. All options are mutually inclusive so you can combine them according to your need. The command output will be the most comprehensive if you combine all options, and you will receive the least details if you use no additional options.
Table 20.2. The \texttt{abrt-cli info} command options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>With no additional option, the \texttt{abrt-cli info} command displays only basic information for the problem specified by the problem data directory argument.</td>
</tr>
<tr>
<td>-d, \texttt{--detailed}</td>
<td>Displays all stored information for the problem specified by the problem data directory argument, including a backtrace if it has already been generated.</td>
</tr>
<tr>
<td>-v, \texttt{--verbose}</td>
<td>\texttt{abrt-cli info} provides additional information on its actions.</td>
</tr>
</tbody>
</table>

20.3.2.2. Reporting Problems

To report a certain problem, use the command:

\begin{verbatim}
abrt-cli report <DIR>
\end{verbatim}

...where \texttt{<DIR>} stands for the problem data directory of the problem that is being reported. For example:

\begin{verbatim}
~\$ abrt-cli report /var/spool/abrt/ccpp-2011-09-13-10:18:14-2895
How you would like to analyze the problem?
1) Collect .xsession-errors
2) Local GNU Debugger
Select analyzer: _
\end{verbatim}

\texttt{ABRT} prompts you to select an analyzer event for the problem that is being reported. After selecting an event, the problem is analyzed. This can take a considerable amount of time. When the problem report is ready, \texttt{abrt-cli} opens a text editor with the content of the report. You can see what is being reported, and you can fill in instructions on how to reproduce the crash and other comments. You should also check the backtrace, because the backtrace might be sent to a public server and viewed by anyone, depending on the problem reporter event settings.

\section*{Selecting a preferred text editor}

You can choose which text editor is used to check the reports. \texttt{abrt-cli} uses the editor defined in the ABRT\_EDITOR environment variable. If the variable is not defined, it checks the VISUAL and EDITOR variables. If none of these variables is set, \texttt{vi} is used. You can set the preferred editor in your \texttt{.bashrc} configuration file. For example, if you prefer GNU Emacs, add the following line to the file:

\begin{verbatim}
export VISUAL=emacs
\end{verbatim}

When you are done with the report, save your changes and close the editor. You will be asked which of the configured \texttt{ABRT} reporter events you want to use to send the report.

\begin{verbatim}
How would you like to report the problem?
1) Logger
2) Red Hat Customer Support
Select reporter(s): _
\end{verbatim}
After selecting a reporting method, you can proceed with reviewing data to be sent with the report. The following table shows options available with the `abrt-cli report` command.

Table 20.3. The `abrt-cli report` command options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>With no additional option, the <code>abrt-cli report</code> provides the usual output.</td>
</tr>
<tr>
<td><code>-v</code>, <code>--verbose</code></td>
<td><code>abrt-cli report</code> provides additional information on its actions.</td>
</tr>
</tbody>
</table>

### 20.3.2.3. Deleting Problems

If you are certain that you do not want to report a particular problem, you can delete it. To delete a problem so ABRT does not keep information about it, use the command:

```
abrt-cli rm <DIR>
```

...where `<DIR>` stands for the problem data directory of the problem being deleted. For example:

```
$ abrt-cli rm /var/spool/abrt/ccpp-2011-09-12-18:37:24-4413
rm '/var/spool/abrt/ccpp-2011-09-12-18:37:24-4413'
```

#### Deletion of a problem can lead to frequent ABRT notification

Note that ABRT performs a detection of duplicate problems by comparing new problems with all locally saved problems. For a repeating crash, ABRT requires you to act upon it only once. However, if you delete the crash dump of that problem, the next time this specific problem occurs, ABRT will treat it as a new crash: ABRT will alert you about it, prompt you to fill in a description, and report it. To avoid having ABRT notifying you about a recurring problem, do not delete its problem data.

The following table shows options available with the `abrt-cli rm` command.

Table 20.4. The `abrt-cli rm` command options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>With no additional option, the <code>abrt-cli rm</code></td>
</tr>
<tr>
<td><code>-v</code>, <code>--verbose</code></td>
<td><code>abrt-cli rm</code> provides additional information on its actions.</td>
</tr>
</tbody>
</table>

### 20.4. Configuring ABRT

A problem life cycle is driven by events in ABRT. For example:

- Event 1 — a problem data directory is created.
- Event 2 — problem data is analyzed.
- Event 3 — a problem is reported to Bugzilla.

When a problem is detected and its defining data is stored, the problem is processed by running events on the problem's data directory. For more information on events and how to define one, refer to Section 20.4.1, “ABRT Events”. Standard ABRT installation currently supports several default events.
that can be selected and used during problem reporting process. Refer to Section 20.4.2, “Standard ABRT Installation Supported Events” to see the list of these events.

Upon installation, ABRT and libreport place their respective configuration files into the several directories on a system:

- `/etc/libreport/` — contains the `report_event.conf` main configuration file. More information about this configuration file can be found in Section 20.4.1, “ABRT Events”.
- `/etc/libreport/events/` — holds files specifying the default setting of predefined events.
- `/etc/libreport/events.d/` — keeps configuration files defining events.
- `/etc/libreport/plugins/` — contains configuration files of programs that take part in events.
- `/etc/abrt/` — holds ABRT specific configuration files used to modify the behavior of ABRT’s services and programs. More information about certain specific configuration files can be found in Section 20.4.4, “ABRT Specific Configuration”.
- `/etc/abrt/plugins/` — keeps configuration files used to override the default setting of ABRT’s services and programs. For more information on some specific configuration files refer to Section 20.4.4, “ABRT Specific Configuration”.

20.4.1. ABRT Events

Each event is defined by one rule structure in a respective configuration file. The configuration files are typically stored in the `/etc/libreport/events.d/` directory. These configuration files are used by the main configuration file, `/etc/libreport/report_event.conf`.

The `/etc/libreport/report_event.conf` file consists of include directives and rules. Rules are typically stored in other configuration files in the `/etc/libreport/events.d/` directory. In the standard installation, the `/etc/libreport/report_event.conf` file contains only one include directive:

```
include events.d/*.conf
```

If you would like to modify this file, please note that it respects shell metacharacters (*,$,?, etc.) and interprets relative paths relatively to its location.

Each rule starts with a line with a non-space leading character, all subsequent lines starting with the space character or the tab character are considered a part of this rule. Each rule consists of two parts, a condition part and a program part. The condition part contains conditions in one of the following forms:

- `VAR=VAL`,
- `VAR!=VAL`, or
- `VAL~REGEX`

...where:

- `VAR` is either the EVENT key word or a name of a problem data directory element (such as executable, package, hostname, etc.),
- `VAL` is either a name of an event or a problem data element, and
- `REGEX` is a regular expression.
The program part consists of program names and shell interpretable code. If all conditions in the condition part are valid, the program part is run in the shell. The following is an event example:

```
EVENT=post-create date > /tmp/dt
    echo $HOSTNAME `uname -r`
```

This event would overwrite the contents of the /tmp/dt file with the current date and time, and print the hostname of the machine and its kernel version on the standard output.

Here is an example of a yet more complex event which is actually one of the predefined events. It saves relevant lines from the ~/.xsession-errors file to the problem report for any problem for which the abrt-ccpp services has been used to process that problem, and the crashed application has loaded any X11 libraries at the time of crash:

```
EVENT=analyze_xsession_errors analyzer=CCpp dso_list=.*/libX11.*
    test -f ~/.xsession-errors || { echo "No ~/.xsession-errors"; exit 1; }
    test -r ~/.xsession-errors || { echo "Can't read ~/.xsession-errors"; exit 1; }
    executable=`cat executable` &&
    base_executable=${executable##*/} &&
    grep -F -e "$base_executable" ~/.xsession-errors | tail -999 >xsession_errors &&
    echo "Element 'xsession_errors' saved"
```

The set of possible events is not hard-set. System administrators can add events according to their need. Currently, the following event names are provided with standard ABRT and libreport installation:

- **post-create**
  This event is run by abrtd on newly created problem data directories. When the post-create event is run, abrtd checks whether the UUID identifier of the new problem data matches the UUID of any already existing problem directories. If such a problem directory exists, the new problem data is deleted.

- **analyze_<NAME_SUFFIX>**
  ...where <NAME_SUFFIX> is the adjustable part of the event name. This event is used to process collected data. For example, the analyze_LocalGDB runs the GNU Debugger (GDB) utility on a core dump of an application and produces a backtrace of a program. You can view the list of analyze events and choose from it using abrt-gui.

- **collect_<NAME_SUFFIX>**
  ...where <NAME_SUFFIX> is the adjustable part of the event name. This event is used to collect additional information on a problem. You can view the list of collect events and choose from it using abrt-gui.

- **report_<NAME_SUFFIX>**
  ...where <NAME_SUFFIX> is the adjustable part of the event name. This event is used to report a problem. You can view the list of report events and choose from it using abrt-gui.

Additional information about events (such as their description, names and types of parameters which can be passed to them as environment variables, and other properties) is stored in the /etc/libreport/events/<event_name>.xml files. These files are used by abrt-gui and abrt-cli to make the user interface more friendly. Do not edit these files unless you want to modify the standard installation.

### 20.4.2. Standard ABRT Installation Supported Events

Standard ABRT installation currently provides a number of default analyzing, collecting and reporting events. Some of these events are also configurable using the ABRT GUI application (for more
information on event configuration using **ABRT** GUI, refer to *Section 20.4.3, “Event Configuration in ABRT GUI”*. **ABRT** GUI only shows the event’s unique part of the name which is more readable the user, instead of the complete event name. For example, the `analyze_xsession_errors` event is shown as `Collect .xsession-errors` in **ABRT** GUI. The following is a list of default analyzing, collecting and reporting events provided by the standard installation of **ABRT**:

**analyze_LocalGDB** — Local GNU Debugger  
Runs **GDB** (the GNU debugger) on problem data of an application and generates a **backtrace** of a program. It is defined in the `/etc/libreport/events.d/ccpp_event.conf` configuration file.

**analyze_xsession_errors** — Collect .xsession-errors  
Saves relevant lines from the `~/.xsession-errors` file to the problem report. It is defined in the `/etc/libreport/events.d/ccpp_event.conf` configuration file.

**report_Logger** — Logger  
Creates a problem report and saves it to a specified local file. It is defined in the `/etc/libreport/events.d/print_event.conf` configuration file.

**report_RHTSupport** — Red Hat Customer Support  
Reports problems to the Red Hat Technical Support system. This possibility is intended for users of Red Hat Enterprise Linux. It is defined in the `/etc/libreport/events.d/rhtsupport_event.conf` configuration file.

**report_Mailx** — Mailx  
Sends a problem report via the **Mailx** utility to a specified email address. It is defined in the `/etc/libreport/events.d/mailx_event.conf` configuration file.

**report_Kerneloops** — Kerneloops.org  
Sends a kernel problem to the oops tracker. It is defined in the `/etc/libreport/events.d/koops_event.conf` configuration file.

**report_Uploader** — Report uploader  
Uploads a tarball (.tar.gz) archive with problem data to the chosen destination using the FTP or the SCP protocol. It is defined in the `/etc/libreport/events.d/uploader_event.conf` configuration file.

### 20.4.3. Event Configuration in ABRT GUI

Events can use parameters passed to them as environment variables (for example, the `report_Logger` event accepts an output file name as a parameter). Using the respective `/etc/libreport/events/<event_name>.xml` file, **ABRT** GUI determines which parameters can be specified for a selected event and allows a user to set the values for these parameters. These values are saved by **ABRT** GUI and reused on subsequent invocations of these events.

Open the **Event Configuration** window by clicking **Edit → Preferences**. This window shows a list of all available events that can be selected during the reporting process. When you select one of the configurable events, you can click the **Configure Event** button and you will be able to configure settings for that event. If you change any of the events' parameters, they are saved in the **Gnome** keyring and will be used in the future GUI sessions.
Do not store sensitive data in global configuration files

All files in the /etc/libreport/ directory hierarchy are world readable and are meant to be used as global settings. Thus, it is not advisable to store usernames, passwords or any other sensitive data in them. The per-user settings (set in the GUI application and readable by the owner of $HOME only) are stored in the Gnome keyring or can be stored in a text file in $HOME/.abrt/*.conf for use in abrt-cli.

The following is a list of all configuration options available for each predefined event that is configurable in the ABRT GUI application.

Logger
In the Logger event configuration window, you can configure the following parameter:

- **Log file** — Specifies a file into which the crash reports are saved (by default, set to /var/log/abrt.log).

  When the **Append** option is checked, the Logger event will append new crash reports to the log file specified in the **Logger file** option. When unchecked, the new crash report always replaces the previous one.

Red Hat Customer Support

In the Red Hat Customer Support event configuration window, you can configure the following parameters:

- **RH Portal URL** — Specifies the Red Hat Customer Support URL where crash dumps are sent (by default, set to https://api.access.redhat.com/rs).
Chapter 20. Automatic Bug Reporting Tool (ABRT)

• **Username** — User login which is used to log into Red Hat Customer Support and create a Red Hat Customer Support database entry for a reported crash. Use your Red Hat Login acquired by creating an account on [http://www.redhat.com/](http://www.redhat.com/), the Red Hat Customer Portal ([https://access.redhat.com/home](https://access.redhat.com/home)) or the Red Hat Network ([https://rhn.redhat.com/](https://rhn.redhat.com/)).

• **Password** — Password used to log into Red Hat Customer Support (that is, password associated with your Red Hat Login)

When the SSL verify option is checked, the SSL protocol is used when sending the data over the network.

**MailX**

In the **MailX** event configuration window, you can configure the following parameters:

• **Subject** — A string that appears in the **Subject** field of a problem report email sent by Mailx (by default, set to "\[abrt\] detected a crash").

• **Sender** — A string that appears in the **From** field of a problem report email.

• **Recipient** — Email address of the recipient of a problem report email.

When the **Send Binary Data** option is checked, the problem report email will also contain all binary files associated with the problem in an attachment. The core dump file is also sent as an attachment.

**Kerneloops.org**

In the **Kerneloops.org** event configuration window, you can configure the following parameter:

• **Kerneloops URL** — Specifies the URL where Kernel problems are reported to (by default, set to [http://submit.kerneloops.org/submitoops.php](http://submit.kerneloops.org/submitoops.php))

**Report Uploader**

In the **Report Uploader** event configuration window, you can configure the following parameter:

• **URL** — Specifies the URL where a tarball containing compressed problem data is uploaded using the FTP or SCP protocol (by default, set to ftp://localhost:/tmp/upload).

### 20.4.4. ABRT Specific Configuration

Standard **ABRT** installation currently provides the following **ABRT** specific configuration files:

• **/etc/abrt/abrt.conf** — allows you to modify the behavior of the **abrd** service.

• **/etc/abrt/abrt-action-save-package-data.conf** — allows you to modify the behavior of the **abrt-action-save-package-data** program.

• **/etc/abrt/plugins/CCpp.conf**, — allows you to modify the behavior of **ABRT**'s core catching hook.

The following configuration directives are supported in the **/etc/abrt/abrt.conf** file:

\[WatchCrashdumpArchiveDir = /var/spool/abrt-upload\]

This directive is commented out by default. Enable it if you want **abrd** to auto-unpack crashdump tarball archives (.tar.gz) which are located in the specified directory. In the example above, it is the **/var/spool/abrt-upload/** directory. Whichever directory you specify in this directive, you must ensure that it exists and it is writable for **abrd**. The **ABRT** daemon will not create it automatically.
ABRT Specific Configuration

Do not modify this option in SELinux

If you are using SELinux, do not modify the default setting of this option unless you reflect the change in SELinux rules. Changing the location for crashdump archives without previous modification of respective rules will cause SELinux denials. See the `abrt_selinux(8)` manual page for more information on running ABRT in SELinux.

Remember that if you enable this option when using SELinux, you need to execute the following command in order to set the appropriate boolean allowing ABRT to write into the public_content_rw_t domain:

```
setsebool -P abrt_anon_write 1
```

MaxCrashReportsSize = `<size_in_megabytes>`

This option sets the amount of storage space, in megabytes, used by ABRT to store all problem information from all users. The default setting is 1000 MB. Once the quota specified here has been met, ABRT will continue catching problems, and in order to make room for the new crash dumps, it will delete the oldest and largest ones.

DumpLocation = `/var/spool/abrt`

This directive is commented out by default. It specifies the location where problem data directories are created and in which problem core dumps and all other problem data are stored. The default location is set to the `/var/spool/abrt` directory. Whichever directory you specify in this directive, you must ensure that it exists and it is writable for abrtd.

Do not modify this option in SELinux

Do not modify the default setting of this option if you are using SELinux. Changing the dump location will cause SELinux denials unless you reflect the change in respective SELinux rules first. See the `abrt_selinux(8)` manual page for more information on running ABRT in SELinux.

Remember that if you enable this option when using SELinux, you need to execute the following command in order to set the appropriate boolean allowing ABRT to write into the public_content_rw_t domain:

```
setsebool -P abrt_anon_write 1
```

The following configuration directives are supported in the `/etc/abrt/abrt-action-save-package-data.conf` file:

OpenGPGCheck = `<yes/no>`

Setting the `OpenGPGCheck` directive to `yes` (the default setting) tells ABRT to only analyze and handle crashes in applications provided by packages which are signed by the GPG keys whose locations are listed in the `/etc/abrt/gpg_keys` file. Setting `OpenGPGCheck` to `no` tells ABRT to catch crashes in all programs.
BlackList = nspluginwrapper, valgrind, strace, [<MORE_PACKAGES>]

Crashes in packages and binaries listed after the BlackList directive will not be handled by ABRT. If you want ABRT to ignore other packages and binaries, list them here separated by commas.

ProcessUnpackaged = <yes/no>

This directive tells ABRT whether to process crashes in executables that do not belong to any package. The default setting is no.

BlackListedPaths = */usr/share/doc/*, */example*

Crashes in executables in these paths will be ignored by ABRT.

The following configuration directives are supported in the /etc/abrt/plugins/CCpp.conf file:

MakeCompatCore = <yes/no>

This directive specifies whether ABRT’s core catching hook should create a core file, as it could be done if ABRT would not be installed. The core file is typically created in the current directory of the crashed program but only if the ulimit -c setting allows it. The directive is set to yes by default.

SaveBinaryImage = <yes/no>

This directive specifies whether ABRT’s core catching hook should save a binary image to a core dump. It is useful when debugging crashes which occurred in binaries that were deleted. The default setting is no.

### 20.4.5. Configuring Automatic Reporting

ABRT can be configured to report any detected issues or crashes automatically without any user interaction. This can be achieved by specifying an analyze-and-report rule as a post-create rule. For example, you can instruct ABRT to report Python crashes to Bugzilla immediately without any user interaction by enabling the rule and replacing the EVENT=report_Bugzilla condition with the EVENT=port-create condition in the /etc/libreport/events.d/python_event.conf file:

```
EVENT=post-create analyzer=Python
test -f component || abrt-action-save-package-data
reporter-bugzilla -c /etc/abrt/plugins/Bugzilla.conf
```

Please note that the post-create event is run by abrtd, which usually runs with root privileges.

### 20.4.6. Uploading and reporting using a proxy server

The reporter-bugzilla and the reporter-upload tools respect the http_proxy and the ftp_proxy environment variables. When you use environment variables as a part of a reporting event, they inherit their values from the process which performs reporting, usually abrt-gui or abrt-cli. Therefore, you can specify HTTP or FTP proxy servers by using these variables in your working environment.

If you arrange these tools to be a part of the post-create event, they will run as children of the abrtd process. You should either adjust the environment of abrtd or modify the rules to set these variables. For example:
20.5. Configuring Centralized Crash Collection

You can set up ABRT so that crash reports are collected from multiple systems and sent to a dedicated system for further processing. This is useful when an administrator does not want to log into hundreds of systems and manually check for crashes found by ABRT. In order to use this method, you need to install the libreport-plugin-reportuploader plug-in (yum install libreport-plugin-reportuploader). See the following sections on how to configure systems to use ABRT's centralized crash collection.

20.5.1. Configuration Steps Required on a Dedicated System

Complete the following steps on a dedicated (server) system:

1. Create a directory to which you want the crash reports to be uploaded to. Usually, /var/spool/abrt-upload/ is used (the rest of the document assumes you are using this directory). Make sure this directory is writable by the abrt user.

   The abrt user and group

   When the abrt-desktop package is installed, it creates a new system user and a group, both named abrt. This user is used by the abrtd daemon, for example, as the owner:group of /var/spool/abrt/* directories.

2. In the /etc/abrt/abrt.conf configuration file, set the WatchCrashdumpArchiveDir directive to the following:

   WatchCrashdumpArchiveDir = /var/spool/abrt-upload/

3. Choose your preferred upload mechanism; for example, FTP or SCP. For more information on how to configure FTP, refer to Section 13.2, “FTP”. For more information on how to configure SCP, refer to Section 9.3.2, “Using the scp Utility”.

   It is advisable to check whether your upload method works. For example, if you use FTP, upload a file using an interactive FTP client:

   ```
   ~]$ ftp
   ftp> open SERVERNAME
   Name: USERNAME
   Password: PASSWORD
   ftp> cd /var/spool/abrt-upload
   250 Operation successful
   ftp> put TESTFILE
   ftp> quit
   ```

   Check whether TESTFILE appeared in the correct directory on the server system.
4. The `MaxCrashReportsSize` directive (in the `/etc/abrt/abrt.conf` configuration file) needs to be set to a larger value if the expected volume of crash data is larger than the default 1000 MB.

5. Consider whether you would like to generate a backtrace of C/C++ crashes.

   You can disable backtrace generation on the server if you do not wish to generate backtraces all at once, or if you decide to create them locally on the machine where a problem occurred. In the standard ABRT installation, a backtrace of a C/C++ crash is generated using the following rule in the `/etc/libreport/events.d/ccpp_events.conf` configuration file:

   ```
   EVENT=analyze_LocalGDB analyzer=CCpp
   abrt-action-analyze-core.py --core=coredump -o build_ids &&
   abrt-action-install-debuginfo-to-abrt-cache --size_mb=4096 &&
   abrt-action-generate-backtrace &&
   abrt-action-analyze-backtrace
   ```

   You can ensure that this rule is not applied for uploaded problem data by adding the `remote!=1` condition to the rule.

6. Decide whether you want to collect package information (the `package` and the `component` elements) in the problem data. Refer to Section 20.5.3, “Saving Package Information” to find out whether you need to collect package information in your centralized crash collection configuration and how to configure it properly.

### 20.5.2. Configuration Steps Required on a Client System

Complete the following steps on every client system which will use the central management method:

1. If you do not wish to generate a backtrace, or if you decided to generate it on a server system, you need to delete or comment out the corresponding rules in the `/etc/libreport/events.d/ccpp_events.conf` file. Refer to Section 20.5.1, “Configuration Steps Required on a Dedicated System” for an example of such an example.

2. If you decided to not collect package information on client machines, delete, comment out or modify the rule which runs `abrt-action-save-package-data` in the `/etc/libreport/events.d/abrt_event.conf` file. Refer to Section 20.5.3, “Saving Package Information” to find out whether you need to collect package information in your centralized crash collection configuration and how to configure it properly.

3. Add a rule for uploading problem reports to the server system in the corresponding configuration file. For example, if you want to upload all problems automatically as soon as they are detected, you can use the following rule in the `/etc/libreport/events.d/abrt_event.conf` configuration file:

   ```
   Alternatively, you can use a similar rule that runs the reporter-upload program as the `report_SFX` event if you want to store problem data locally on clients and upload it later using ABRT GUI/CLI. The following is an example of such an event:

### 20.5.3. Saving Package Information

In a single-machine ABRT installation, problems are usually reported to external bug databases such as RHTSupport or Bugzilla. Reporting to these bug databases usually requires knowledge about the component and package in which the problem occurred. The `post-create` event runs the `abrt-action-save-package-data` tool (among other steps) in order to provide this information in the standard ABRT installation.
If you are setting up a centralized crash collection system, your requirements may be significantly different. Depending on your needs, you have two options:

**Internal analysis of problems**

After collecting problem data, you do not need to collect package information if you plan to analyze problems in-house, without reporting them to any external bug databases. You might be also interested in collecting crashes that occur in programs written by your organization. Such programs do not belong to any package in the first place. In this case take the following steps on both, client systems and a dedicated crash collecting system:

- Remove the following rule from the `/etc/libreport/events.d/abrt_event.conf` file:

- Prevent deletion of problem data directories which do not correspond to any installed package by setting the following directive in the `/etc/abrt/abrt-action-save-package-data.conf` file:

  ```
  ProcessUnpackaged = yes
  ```

**Reporting to external bug database**

Alternatively, you may want to report crashes to RHTSupport or Bugzilla. In this case, you need to collect package information. Generally, client machines and dedicated crash collecting systems have non-identical sets of installed packages. Therefore, it may happen that problem data uploaded from a client does not correspond to any package installed on the dedicated crash collecting system. In the standard ABRT configuration, this will lead to deletion of problem data (ABRT will consider it to be a crash in an unpackaged executable). To prevent this from happening, it is necessary to modify ABRT's configuration on the dedicated system in the following way:

- Prevent inadvertent collection of package information for problem data uploaded from client machines, by adding the `remote!=1` condition in the `/etc/libreport/events.d/abrt_event.conf` file:

- Prevent deletion of problem data directories which do not correspond to any installed package by setting the following directive in `/etc/abrt/abrt-action-save-package-data.conf`:

  ```
  ProcessUnpackaged = yes
  ```

**Note**

Note that in this case, no such modifications are necessary on client systems: they continue to collect package information, and continue to ignore crashes in unpackaged executables.

**20.5.4. Testing ABRT's Crash Detection**

After completing all the steps of the configuration process, the basic setup is finished. To test that this setup works properly use the `kill -s SEGV PID` command to terminate a process on a client system. For example, start a `sleep` process and terminate it with the `kill` command in the following way:

```
$ sleep 100 &
[1] 2823
```
~]$ kill -s SEGV 2823

**ABRT** should detect a crash shortly after executing the **kill** command. Check that the crash was detected by **ABRT** on the client system (this can be checked by examining the appropriate syslog file, by running the **abrt-cli list --full** command, or by examining the crash dump created in the `/var/spool/abrt` directory), copied to the server system, unpacked on the server system and can be seen and acted upon using **abrt-cli** or **abrt-gui** on the server system.
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 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.

- **oprofile does not associate samples for inline functions properly** — oprofile 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.

- **Hardware performance counters do not work on guest virtual machines** — Because the hardware performance counters are not available on virtual systems, you need to use the timer mode. Run the command opcontrol --deinit, and then execute modprobe oprofile timer=1 to enable the timer mode.

- **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).

### 21.1. Overview of Tools

Table 21.1, “OProfile Commands” provides a brief overview of the tools provided with the oprofile package.
Table 21.1. OProfile Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ophelp</td>
<td>Displays available events for the system's processor along with a brief description of each.</td>
</tr>
<tr>
<td>opimport</td>
<td>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.</td>
</tr>
<tr>
<td>opannotate</td>
<td>Creates annotated source for an executable if the application was compiled with debugging symbols. Refer to Section 21.5.4, “Using opannotate” for details.</td>
</tr>
<tr>
<td>opcontrol</td>
<td>Configures what data is collected. Refer to Section 21.2, “Configuring OProfile” for details.</td>
</tr>
<tr>
<td>oreport</td>
<td>Retrieves profile data. Refer to Section 21.5.1, “Using oreport” for details.</td>
</tr>
<tr>
<td>oprofiled</td>
<td>Runs as a daemon to periodically write sample data to disk.</td>
</tr>
</tbody>
</table>

21.2. 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.

21.2.1. 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
```

Install the debuginfo package

The debuginfo package for the kernel 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 21.6, “Understanding /dev/oprofile/” for details about this directory.

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 21.2.3, “Separating Kernel and User-space Profiles”.

### 21.2.2. Setting Events to Monitor

Most processors contain counters, which are used by OProfile to monitor specific events. As shown in Table 21.2, “OProfile Processors and Counters”, the number of counters available depends on the processor.

<table>
<thead>
<tr>
<th>Processor</th>
<th>cpu_type</th>
<th>Number of Counters</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMD64</td>
<td>x86-64/hammer</td>
<td>4</td>
</tr>
<tr>
<td>AMD Athlon</td>
<td>i386/athlon</td>
<td>4</td>
</tr>
<tr>
<td>AMD Family 10h</td>
<td>x86-64/family10</td>
<td>4</td>
</tr>
<tr>
<td>AMD Family 11h</td>
<td>x86-64/family11</td>
<td>4</td>
</tr>
<tr>
<td>AMD Family 12h</td>
<td>x86-64/family12</td>
<td>4</td>
</tr>
<tr>
<td>AMD Family 14h</td>
<td>x86-64/family14</td>
<td>4</td>
</tr>
<tr>
<td>AMD Family 15h</td>
<td>x86-64/family15</td>
<td>6</td>
</tr>
<tr>
<td>IBM eServer System i and IBM eServer System p</td>
<td>timer</td>
<td>1</td>
</tr>
<tr>
<td>IBM POWER4</td>
<td>ppc64/power4</td>
<td>8</td>
</tr>
<tr>
<td>IBM POWER5</td>
<td>ppc64/power5</td>
<td>6</td>
</tr>
<tr>
<td>IBM PowerPC 970</td>
<td>ppc64/970</td>
<td>8</td>
</tr>
<tr>
<td>IBM S/390 and IBM System z</td>
<td>timer</td>
<td>1</td>
</tr>
<tr>
<td>Intel Core i7</td>
<td>i386/core_i7</td>
<td>4</td>
</tr>
<tr>
<td>Intel Nehalem microarchitecture</td>
<td>i386/nehalem</td>
<td>4</td>
</tr>
<tr>
<td>Intel Pentium 4 (non-hyper-threaded)</td>
<td>i386/p4</td>
<td>8</td>
</tr>
<tr>
<td>Intel Pentium 4 (hyper-threaded)</td>
<td>i386/p4-ht</td>
<td>4</td>
</tr>
<tr>
<td>Intel Westmere microarchitecture</td>
<td>i386/westmere</td>
<td>4</td>
</tr>
<tr>
<td>TIMER_INT</td>
<td>timer</td>
<td>1</td>
</tr>
</tbody>
</table>

Use Table 21.2, “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 21.3, “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 21.9, “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.

## 21.2.2.1. 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.

![Warning]

**Sampling too frequently can overload the system**

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.

### 21.2.2.2. 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 operation.

```
$ opcontrol --event=event-name:sample-rate:unit-mask
```

### 21.2.3. 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
```
Chapter 21. OProfile

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.

### Restart the OProfile profiler

These configuration changes will take effect when the OProfile profiler is restarted.

## 21.3. 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:

```
```

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`.
**Disable the nmi_watchdog registers**

On a Fedora 20 system, the `nmi_watchdog` registers with the `perf` subsystem. Due to this, the `perf` subsystem grabs control of the performance counter registers at boot time, blocking OProfile from working.

To resolve this, either boot with the `nmi_watchdog=0` kernel parameter set, or run the following command to disable `nmi_watchdog` at run time:

```bash
~# echo 0 > /proc/sys/kernel/nmi_watchdog
```

To re-enable `nmi_watchdog`, use the following command:

```bash
~# echo 1 > /proc/sys/kernel/nmi_watchdog
```

To stop the profiler, execute the following command as root:

```bash
~# opcontrol --shutdown
```

### 21.4. 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.

```bash
~# opcontrol --save=name
```

The directory `/var/lib/oprofile/samples/name/` is created and the current sample files are copied to it.

### 21.5. 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:

```bash
~# 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.

---

**Back up the executable and the sample files**

The executable being profiled must be used with these tools to analyze the data. If it must change after the data is collected, back up 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 is not 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 21.4, “Saving Data” for details on how to back up the sample file.

**21.5.1. 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
6  0.0301 libgd-k-x11-2.0.so.0.400.13
6  0.0226 libgobject-2.0.so.0.400.7
5  0.0188 oprofile
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.

### 21.5.2. 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.
Chapter 21. OProfile

- **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`:

<table>
<thead>
<tr>
<th>samples % symbol name</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 100.000 __gconv_transform_utf8_internal</td>
</tr>
</tbody>
</table>

  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`:

<table>
<thead>
<tr>
<th>vma samples % symbol name</th>
</tr>
</thead>
<tbody>
<tr>
<td>00a98640 12 100.000 __gconv_transform_utf8_internal</td>
</tr>
<tr>
<td>00a98640 1 8.3333</td>
</tr>
<tr>
<td>00a9868c 2 16.6667</td>
</tr>
<tr>
<td>00a9869a 1 8.3333</td>
</tr>
<tr>
<td>00a986c1 1 8.3333</td>
</tr>
<tr>
<td>00a98720 1 8.3333</td>
</tr>
<tr>
<td>00a98749 1 8.3333</td>
</tr>
<tr>
<td>00a98753 1 8.3333</td>
</tr>
<tr>
<td>00a98789 1 8.3333</td>
</tr>
<tr>
<td>00a98864 1 8.3333</td>
</tr>
<tr>
<td>00a98869 1 8.3333</td>
</tr>
<tr>
<td>00a98b08 1 8.3333</td>
</tr>
</tbody>
</table>

  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/oprobe(samples/)` directory.

**21.5.3. 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.
To get a more detailed view of the actions of the module, you will need to either have the module "unstripped" (that is installed from a custom build) or have the debuginfo package installed for the kernel.

Find out which kernel is running with the `uname -a` command, obtain the appropriate debuginfo package and install it on the machine.

Then proceed with clearing out the samples from previous runs with the following command:

```
~# opcontrol --reset
```

To start the monitoring process, for example, on a machine with Westmere processor, run the following command:

```
~# opcontrol --setup --vmlinux=/usr/lib/debug/lib/modules/'uname -r'/vmlinux \
 --event=CPU_CLK_UNHALTED:500000
```

Then the detailed information, for instance, for the ext4 module can be obtained with:

```
~# opreport /ext4 -l --image-path /lib/modules/`uname -r`/kernel
```

21.5.4. 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 appropriate debuginfo package for the executable must be installed on the system. By default, Fedora debuginfo packages are not installed together with their corresponding packages, which contain the executable, so that you have to obtain and install the debuginfo packages separately.

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.
21.6. 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:

\[
\texttt{\textasciitilde\# cat /dev/oprofile/cpu_type}
\]

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.
- **extra** — Used on machines with Nehalem processors to further specify 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:

\[
\texttt{\textasciitilde\# cat /dev/oprofile/0/count}
\]

21.7. 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.

21.8. OProfile Support for Java

OProfile allows you to profile dynamically compiled code (also known as “just-in-time” or JIT code) of the Java Virtual Machine (JVM). OProfile in Fedora 20 includes build-in support for the JVM Tools Interface (JVMTI) agent library, which supports Java 1.5 and higher.
To profile JIT code from the Java Virtual Machine with the JVMTI agent, add the following to the JVM startup parameters:

```
-agentlib:jvmti_oprofile
```

**Install the oprofile-jit package**
The `oprofile-jit` package must be installed on the system in order to profile JIT code with OProfile.

To learn more about Java support in OProfile, refer to the OProfile Manual, which is linked from Section 21.11, "Additional Resources".

21.9. 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 21.2.2, "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.
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 21.2.3, “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 21.2.3, “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 21.2.2.1, “Sampling Rate”.

If any unit masks are available for the currently selected event, as discussed in Section 21.2.2.2, “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**.
If the **Verbose** option is selected, the **oprofiled** daemon log includes more information.

If **Per-application profiles** is selected, OProfile generates per-application profiles for libraries. This is equivalent to the `opcontrol --separate=library` command. If **Per-application profiles, including kernel** is selected, OProfile generates per-application profiles for the kernel and kernel modules as discussed in Section 21.2.3, “Separating Kernel and User-space Profiles”. This is equivalent to the `opcontrol --separate=kernel` command.

To force data to be written to samples files as discussed in Section 21.5, “Analyzing the Data”, click the **Flush** button. This is equivalent to the `opcontrol --dump` command.

To start OProfile from the graphical interface, click **Start**. To stop the profiler, click **Stop**. Exiting the application does not stop OProfile from sampling.

### 21.10. OProfile and SystemTap

SystemTap is a tracing and probing tool that allows users to study and monitor the activities of the operating system in fine detail. It provides information similar to the output of tools like `netstat`, `ps`, and others.
top, and iostat; however, SystemTap is designed to provide more filtering and analysis options for collected information.

While using OProfile is suggested in cases of collecting data on where and why the processor spends time in a particular area of code, it is less usable when finding out why the processor stays idle.

You might want to use SystemTap when instrumenting specific places in code. Because SystemTap allows you to run the code instrumentation without having to stop and restart the instrumentation, it is particularly useful for instrumenting the kernel and daemons.

For more information on SystemTap, refer to Section 21.11.2, “Useful Websites” for the relevant SystemTap documentation.

21.11. Additional Resources

This chapter only highlights OProfile and how to configure and use it. To learn more, refer to the following resources.

21.11.1. Installed Docs

• /usr/share/doc/oprofile-version/oprofile.html — OProfile Manual

• oprofile man page — Discusses opcontrol, opreport, opannotate, and ophelp

21.11.2. Useful Websites

• http://oprofile.sourceforge.net/ — Contains the latest documentation, mailing lists, IRC channels, and more.

• SystemTap Beginners Guide1 — Provides basic instructions on how to use SystemTap to monitor different subsystems of Fedora in finer detail.

Part VI. Kernel, Module and Driver Configuration

This part covers various tools that assist administrators with kernel customization.
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 a kernel is released, it must first pass a rigorous set of quality assurance tests.

Fedora kernels are packaged in the RPM format so that they are easy to upgrade and verify using the Yum or PackageKit package managers. PackageKit automatically queries the Yum repositories and informs you of packages with available updates, including kernel packages.

This chapter is therefore only useful for users who need to manually update a kernel package using the `rpm` command instead of `yum`.

⚠️ Use Yum to install kernels whenever possible

Whenever possible, use either the Yum or PackageKit package manager to install a new kernel because they always install a new kernel instead of replacing the current one, which could potentially leave your system unable to boot.

For more information on installing kernel packages with Yum, refer to Section 5.1.2, “Updating Packages”.

22.1. Overview of Kernel Packages

Fedora contains the following kernel packages:

- `kernel` — Contains the kernel for single, multicore and multiprocessor systems.
- `kernel-debug` — Contains a kernel with numerous debugging options enabled for kernel diagnosis, at the expense of reduced performance.
- `kernel-devel` — Contains the kernel headers and makefiles sufficient to build modules against the `kernel` package.
- `kernel-debug-devel` — Contains the development version of the kernel with numerous debugging options enabled for kernel diagnosis, at the expense of reduced performance.
- `kernel-doc` — 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-kernel_version/` directory.

- `kernel-headers` — Includes the C header files that specify the interface between the Linux kernel and user-space libraries and programs. The header files define structures and constants that are needed for building most standard programs.
- `linux-firmware` — Contains all of the firmware files that are required by various devices to operate.
• **perf** — This package contains supporting scripts and documentation for the **perf** tool shipped in each kernel image subpackage.

### 22.2. Preparing to Upgrade

Before upgrading the kernel, it is recommended that you take some precautionary steps.

First, ensure that 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.

USB media often comes in the form of flash devices sometimes called **pen drives**, **thumb disks**, or **keys**, or as an externally-connected hard disk device. Almost all media of this type is formatted as a VFAT file system. You can create bootable USB media on media formatted as ext2, ext3, ext4, or VFAT.

You can transfer a distribution image file or a minimal boot media image file to USB media. Make sure that sufficient free space is available on the device. Around 4 GB is required for a distribution DVD image, around 700 MB for a distribution CD image, or around 10 MB for a minimal boot media image.

You must have a copy of the **boot.iso** file from a Fedora installation DVD, or installation CD-ROM#1, and you need a USB storage device formatted with the VFAT file system and around 16 MB of free space. The following procedure will not affect existing files on the USB storage device unless they have the same path names as the files that you copy onto it. To create USB boot media, perform the following commands as the root user:

1. Install the **SYSLINUX** bootloader on the USB storage device:

   ```
   syslinux /dev/sdX1
   ...
   ```

   where **sdX** is the device name.

2. Create mount points for **boot.iso** and the USB storage device:

   ```
   mkdir /mnt/isoboot /mnt/diskboot
   ```

3. Mount **boot.iso**:

   ```
   mount -o loop boot.iso /mnt/isoboot
   ```

4. Mount the USB storage device:

   ```
   mount /dev/sdX1 /mnt/diskboot
   ```

5. Copy the **ISOLINUX** files from the **boot.iso** to the USB storage device:

   ```
   cp /mnt/isoboot/isolinux/* /mnt/diskboot
   ```

6. Use the **isolinux.cfg** file from **boot.iso** as the **syslinux.cfg** file for the USB device:

   ```
   grep -v local /mnt/isoboot/isolinux/isolinux.cfg > /mnt/diskboot/syslinux.cfg
   ```
7. Unmount `boot.iso` and the USB storage device:

```
    umount /mnt/isoboot /mnt/diskboot
```

8. You should reboot the machine with the boot media and verify that you are able to boot with it before continuing.

Alternatively, on systems with a floppy drive, you can create a boot diskette by installing the `mkbootdisk` package and running the `mkbootdisk` command as `root`. Refer to `man mkbootdisk` man page after installing the package for usage information.

To determine which kernel packages are installed, execute the command `yum list installed "kernel-*"` at a shell prompt. The output will comprise some or all of the following packages, depending on the system's architecture, and the version numbers may differ:

```
-][# yum list installed "kernel-*"
Loaded plugins: langpacks, presto, refresh-packagekit
Installed Packages
  kernel.x86_64  3.1.0-0.rc6.git0.3.fc16 @updates-testing
  kernel.x86_64  3.1.0-0.rc9.git0.0.fc16 @updates-testing
  kernel-doc.x86_64  3.1.0-0.rc6.git0.3.fc16 @updates-testing
  kernel-doc.x86_64  3.1.0-0.rc9.git0.0.fc16 @updates-testing
  kernel-headers.x86_64  3.1.0-0.rc6.git0.3.fc16 @updates-testing
  kernel-headers.x86_64  3.1.0-0.rc9.git0.0.fc16 @updates-testing
```

From the output, determine which packages need to be downloaded for the kernel upgrade. For a single processor system, the only required package is the `kernel` package. Refer to Section 22.1, “Overview of Kernel Packages” for descriptions of the different packages.

### 22.3. Downloading the Upgraded Kernel

There are several ways to determine if an updated kernel is available for the system.

- Security Advisories — Refer to [http://fedoraproject.org/wiki/FSA](http://fedoraproject.org/wiki/FSA) for information on Security Advisories, including kernel upgrades that fix security issues.

- Via Fedora Update System — Download and install the kernel RPM packages. For more information, refer to [http://admin.fedoraproject.org/updates/](http://admin.fedoraproject.org/updates/).

To install the kernel manually, continue to Section 22.4, “Performing the Upgrade”.

### 22.4. Performing the Upgrade

After retrieving all of the necessary packages, it is time to upgrade the existing kernel.

Keep the old kernel when performing the upgrade

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 22.5, “Verifying the Initial RAM Disk Image” for details.

### 22.5. Verifying the Initial RAM Disk Image

The job of the initial RAM disk image is to preload the block device modules, such as for IDE, SCSI or RAID, so that the root file system, on which those modules normally reside, can then be accessed and mounted. On Fedora 20 systems, whenever a new kernel is installed using either the Yum, PackageKit, or RPM package manager, the Dracut utility is always called by the installation scripts to create an initramfs (initial RAM disk image).

On all architectures other than IBM eServer System i (see the section called “Verifying the Initial RAM Disk Image and Kernel on IBM eServer System i”), you can create an initramfs by running the `dracut` command. However, you usually don't need to create an initramfs manually: this step is automatically performed if the kernel and its associated packages are installed or upgraded from RPM packages distributed by The Fedora Project.

On architectures that use the GRUB 2 boot loader, you can verify that an initramfs corresponding to your current kernel version exists and is specified correctly in the `/boot/grub2/grub.cfg` configuration file by following this procedure:

**Procedure 22.1. Verifying the Initial RAM Disk Image**

1. As root, list the contents in the `/boot` directory and find the kernel (vmlinuz-kernel_version) and initramfs-kernel_version with the latest (most recent) version number:

   ```
   $ ls /boot
   config-3.1.0-0.rc6.git0.3.fc16.x86_64
   config-3.1.0-0.rc9.git0.0.fc16.x86_64
   elf-memtest86+-4.20
   grub
   grub2
   initramfs-3.1.0-0.rc6.git0.3.fc16.x86_64.img
   initramfs-3.1.0-0.rc9.git0.0.fc16.x86_64.img
   initrd-plymouth.img
   memtest86+-4.20
   System.map-3.1.0-0.rc6.git0.3.fc16.x86_64
   System.map-3.1.0-0.rc9.git0.0.fc16.x86_64
   vmlinuz-3.1.0-0.rc6.git0.3.fc16.x86_64
   vmlinuz-3.1.0-0.rc9.git0.0.fc16.x86_64
   ```

   The example above shows that:

   - we have two kernels installed (or, more correctly, two kernel files are present in the `/boot` directory),
   - the latest kernel is `vmlinuz-vmlinuz-3.1.0-0.rc9.git0.0.fc16.x86_64`, and
   - an initramfs file matching our kernel version, `initramfs-3.1.0-0.rc9.git0.0.fc16.x86_64.img`, also exists.
Verifying the Initial RAM Disk Image

initrd files in the /boot directory are not the same as initramfs files

In the /boot directory you may find several initrd-<kernel_version>kdump.img files. These are special files created by the kdump mechanism for kernel debugging purposes, are not used to boot the system, and can safely be ignored. For more information on kdump, refer to Chapter 24, The kdump Crash Recovery Service.

2. (Optional) If your initramfs-kernel_version file does not match the version of the latest kernel in /boot, or, in certain other situations, you may need to generate an initramfs file with the Dracut utility. Simply invoking dracut as root without options causes it to generate an initramfs file in the /boot directory for the latest kernel present in that directory:

```
~# dracut
```

You must use the --force option if you want dracut to overwrite an existing initramfs (for example, if your initramfs has become corrupt). Otherwise dracut will refuse to overwrite the existing initramfs file:

```
~# dracut
F: Will not override existing initramfs (/boot/initramfs-3.1.0-0.fc16.x86_64.img) without --force
```

You can create an initramfs in the current directory by calling dracut initramfs_name kernel_version, for example:

```
~# dracut "initramfs-$(uname -r).img" $(uname -r)
```

If you need to specify specific kernel modules to be preloaded, add the names of those modules (minus any file name suffixes such as .ko) inside the parentheses of the add_dracutmodules="module [more_modules]" directive of the /etc/dracut.conf configuration file. You can list the file contents of an initramfs image file created by dracut by using the lsinitrd initramfs_file command:

```
~# lsinitrd /boot/initramfs-3.1.0-0.fc9.git0.0.fc16.x86_64.img
```

Refer to man dracut and man dracut.conf for more information on options and usage.

3. Examine the /boot/grub2/grub.cfg configuration file to ensure that an initrd /path/initramfs-kernel_version.img exists for the kernel version you are booting. For example:
Chapter 22. Manually Upgrading the Kernel

```
-]# grep initrd /boot/grub2/grub.cfg
  initrd /initramfs-3.1.0-0.rc6.git0.3.fc16.x86_64.img
  initrd /initramfs-3.1.0-0.rc9.git0.0.fc16.x86_64.img
```

Refer to Section 22.6, “Verifying the Boot Loader” for more information on how to read and update the `/boot/grub2/grub.cfg` file.

## Verifying the Initial RAM Disk Image and Kernel on IBM eServer System i

On IBM eServer System i machines, the initial RAM disk and kernel files are combined into a single 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 The Fedora Project; thus, it does not need to be executed manually. To verify that it was created, run the following command as root to make sure the `/boot/vmlinitrd-kernel_version` file already exists:

```
ls -l /boot
```

The `kernel_version` should match the version of the kernel just installed.

### 22.6. Verifying the Boot Loader

When you install a kernel using `rpm`, the kernel package creates an entry in the boot loader configuration file for that new kernel. However, `rpm` does not configure the new kernel to boot as the default kernel. You must do this manually when installing a new kernel with `rpm`.

It is always recommended to double-check the boot loader configuration file after installing a new kernel with `rpm` to ensure that the configuration is correct. Otherwise, the system may not be able to boot into Fedora properly. If this happens, boot the system with the boot media created earlier and re-configure the boot loader.

In the following table, find your system's architecture to determine the boot loader it uses, and then click on the “Refer to” link to jump to the correct instructions for your system.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Boot Loader</th>
<th>Refer to</th>
</tr>
</thead>
<tbody>
<tr>
<td>x86</td>
<td>GRUB 2</td>
<td>Section 22.6.1, “Configuring the GRUB 2 Boot Loader”</td>
</tr>
<tr>
<td>AMD AMD64 or Intel 64</td>
<td>GRUB 2</td>
<td>Section 22.6.1, “Configuring the GRUB 2 Boot Loader”</td>
</tr>
<tr>
<td>IBM eServer System i</td>
<td>OS/400</td>
<td>Section 22.6.2, “Configuring the OS/400 Boot Loader”</td>
</tr>
<tr>
<td>IBM eServer System p</td>
<td>YABOOT</td>
<td>Section 22.6.3, “Configuring the YABOOT Boot Loader”</td>
</tr>
<tr>
<td>IBM System z</td>
<td>z/IPL</td>
<td>—</td>
</tr>
</tbody>
</table>
22.6.1. Configuring the GRUB 2 Boot Loader

Fedora 20 is distributed with GRUB 2, which reads its configuration from the /boot/grub2/grub.cfg file. This file is generated by the grub2-mkconfig utility based on Linux kernels located in the /boot directory, template files located in /etc/grub.d/, and custom settings in the /etc/default/grub file and is automatically updated each time you install a new kernel from an RPM package. To update this configuration file manually, type the following at a shell prompt as root:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

Among various code snippets and directives, the /boot/grub2/grub.cfg configuration file contains one or more `menuentry` blocks, each representing a single GRUB 2 boot menu entry. These blocks always start with the `menuentry` keyword followed by a title, list of options, and opening curly bracket, and end with a closing curly bracket. Anything between the opening and closing bracket should be indented. For example, the following is a sample `menuentry` block for Fedora 17 with Linux kernel 3.4.0-1.fc17.x86_64:

```
menuentry 'Fedora (3.4.0-1.fc17.x86_64)' --class fedora --class gnu-linux --class gnu --class os $menuentry_id_option 'gnulinux-simple-77ba9149-751a-48e0-974f-ad94911734b9' {
    load_video
    set gfxpayload=keep
    insmod gizo
    insmod part_msdos
    insmod ext2
    set root='hd0,msdos1'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint='hd0,msdos1' 4ea24c68-ab10-47d4-8a6b-b8d3a02acba
    else
        search --no-floppy --fs-uuid --set=root 4ea24c68-ab10-47d4-8a6b-b8d3a02acba
    fi
    echo 'Loading Fedora (3.4.0-1.fc17.x86_64)'
    linux /vmlinuz-3.4.0-1.fc17.x86_64 root=/dev/vg_fedora-lv_root ro rd.md=0 rd.dm=0 SYSFONT=True rd.lvm.lv=vg_fedora/lv_swap KEYTABLE=us rd.lvm.lv=vg_fedora/lv_root rd.luks=0 LANG=en_US.UTF-8 rhgb quiet
    echo 'Loading initial ramdisk ...'
    initrd /initramfs-3.4.0-1.fc17.x86_64.img
}
```

Each `menuentry` block that represents an installed Linux kernel contains `linux` and `initrd` directives followed by the path to the kernel and the initramfs image respectively. If a separate /boot partition was created, the paths to the kernel and the initramfs image are relative to /boot. In the example above, the `initrd /initramfs-3.4.0-1.fc17.x86_64.img` line means that the initramfs image is actually located at /boot/initramfs-3.4.0-1.fc17.x86_64.img when the root file system is mounted, and likewise for the kernel path.

The kernel version number as given on the `linux /vmlinuz-kernel_version` line must match the version number of the initramfs image given on the `initrd /initramfs-kernel_version.img` line of each `menuentry` block. For more information on how to verify the initial RAM disk image, refer to Procedure 22.1, “Verifying the Initial RAM Disk Image”.
Chapter 22. Manually Upgrading the Kernel

The initrd directive in grub.cfg refers to an initramfs image

In menuentry blocks, the initrd directive must point to the location (relative to the /boot directory if it is on a separate partition) of the initramfs file corresponding to the same kernel version. This directive is called initrd because the previous tool which created initial RAM disk images, mkinitrd, created what were known as initrd files. The grub.cfg directive remains initrd to maintain compatibility with other tools. The file-naming convention of systems using the dracut utility to create the initial RAM disk image is initramfs-kernel_version.img.

For information on using Dracut, refer to Section 22.5, “Verifying the Initial RAM Disk Image”.

After installing a new kernel with rpm, verify that /boot/grub2/grub.cfg is correct and reboot the computer into the new kernel. Ensure your hardware is detected by watching the boot process output.

If GRUB 2 presents an error and is unable to boot into the new kernel, it is often easiest to try to boot into an alternative or older kernel so that you can fix the problem. Alternatively, use the boot media you created earlier to boot the system.

Causing the GRUB 2 boot menu to display

If you set the GRUB_TIMEOUT option in the /etc/default/grub file to 0, GRUB 2 will not display its list of bootable kernels when the system starts up. In order to display this list when booting, press and hold any alphanumeric key while and immediately after BIOS information is displayed, and GRUB 2 will present you with the GRUB menu.

22.6.2. Configuring the OS/400 Boot Loader

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.

22.6.3. Configuring the YABOOT Boot Loader

IBM eServer System p uses YABOOT as its boot loader. YABOOT uses /etc/aboot.conf as its configuration file. Confirm that the file contains an image section with the same version as the kernel package just installed, and likewise for the initramfs image:

```
boot=/dev/sdal init-message=Welcome to Fedora! Hit <TAB> for boot options
```
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.
Working with Kernel Modules

The Linux kernel is modular, which means it can extend its capabilities through the use of dynamically-loaded kernel modules. A kernel module can provide:

- a device driver which adds support for new hardware; or,
- support for a file system such as btrfs or NFS.

Like the kernel itself, modules can take parameters that customize their behavior, though the default parameters work well in most cases. User-space tools can list the modules currently loaded into a running kernel; query all available modules for available parameters and module-specific information; and load or unload (remove) modules dynamically into or from a running kernel. Many of these utilities, which are provided by the kmod package, take module dependencies into account when performing operations so that manual dependency-tracking is rarely necessary.

On modern systems, kernel modules are automatically loaded by various mechanisms when the conditions call for it. However, there are occasions when it is necessary to load or unload modules manually, such as when one module is preferred over another although either could provide basic functionality, or when a module is misbehaving.

This chapter explains how to:

- use the user-space kmod utilities to display, query, load and unload kernel modules and their dependencies;
- set module parameters both dynamically on the command line and permanently so that you can customize the behavior of your kernel modules; and,
- load modules at boot time.

Installing the kmod package

In order to use the kernel module utilities described in this chapter, first ensure the kmod package is installed on your system by running, as root:

```
]$ yum install kmod
```

For more information on installing packages with Yum, refer to Section 5.2.4, “Installing Packages”.

23.1. Listing Currently-Loaded Modules

You can list all kernel modules that are currently loaded into the kernel by running the `lsmod` command, for example:

```
]$ lsmod
Module                  Size  Used by
xfs                   803635  1
exportfs                3424  1 xfs
vfat                    8216  1 vfat
fat                    43410  1 vfat
```
Each row of `lsmod` output specifies:

- the name of a kernel module currently loaded in memory;
- the amount of memory it uses; and,
- the sum total of processes that are using the module and other modules which depend on it, followed by a list of the names of those modules, if there are any. Using this list, you can first unload all the modules depending the module you want to unload. For more information, refer to Section 23.4, “Unloading a Module”.

Finally, note that `lsmod` output is less verbose and considerably easier to read than the content of the `/proc/modules` pseudo-file.

# 23.2. Displaying Information About a Module

You can display detailed information about a kernel module by running the `modinfo module_name` command.

**Module names do not end in .ko**

When entering the name of a kernel module as an argument to one of the `kmod` utilities, do not append a `.ko` extension to the end of the name. Kernel module names do not have extensions; their corresponding files do.
Example 23.1. Listing information about a kernel module with lsmod

To display information about the e1000e module, which is the Intel PRO/1000 network driver, run:

```
[-]# modinfo e1000e
filename: /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/net/e1000e/e1000e.ko
version: 1.2.7-k2
license: GPL
description: Intel(R) PRO/1000 Network Driver
author: Intel Corporation, <linux.nics@intel.com>
srcversion: 93CB73D39958501872B2982
alias: pci:v00008086d00001503sv*sd*bc*sc*i*
alias: pci:v00008086d00001502sv*sd*bc*sc*i*
alias: [some alias lines omitted]
alias: pci:v00008086d0000105Esv*sd*bc*sc*i*
depends:
vermagic: 2.6.32-71.el6.x86_64 SMP mod_unload modversions
parm: copybreak:Maximum size of packet that is copied to a new buffer on receive
      (uint)
parm: TxIntDelay:Transmit Interrupt Delay (array of int)
parm: TxAbsIntDelay:Transmit Absolute Interrupt Delay (array of int)
parm: RxIntDelay:Receive Interrupt Delay (array of int)
parm: RxAbsIntDelay:Receive Absolute Interrupt Delay (array of int)
parm: InterruptThrottleRate:Interrupt Throttling Rate (array of int)
parm: IntMode:Interrupt Mode (array of int)
parm: SmartPowerDownEnable:Enable PHY smart power down (array of int)
parm: KumeranLockLoss:Enable Kumeran lock loss workaround (array of int)
parm: WriteProtectNVM:Write-protect NVM [WARNING: disabling this can lead to
      corrupted NVM] (array of int)
parm: CrcStripping:Enable CRC Stripping, disable if your BMC needs the CRC
      (array of int)
parm: EEE:Enable/disable on parts that support the feature (array of int)
```

Here are descriptions of a few of the fields in `modinfo` output:

filename

The absolute path to the .ko kernel object file. You can use `modinfo -n` as a shortcut command for printing only the filename field.

description

A short description of the module. You can use `modinfo -d` as a shortcut command for printing only the description field.

alias

The alias field appears as many times as there are aliases for a module, or is omitted entirely if there are none.

depends

This field contains a comma-separated list of all the modules this module depends on.

Omitting the depends field

If a module has no dependencies, the depends field may be omitted from the output.
parm

Each parm field presents one module parameter in the form parameter_name: description, where:

- parameter_name is the exact syntax you should use when using it as a module parameter on the command line, or in an option line in a .conf file in the /etc/modprobe.d/ directory; and,
- description is a brief explanation of what the parameter does, along with an expectation for the type of value the parameter accepts (such as int, unit or array of int) in parentheses.

**Example 23.2. Listing module parameters**

You can list all parameters that the module supports by using the -p option. However, because useful value type information is omitted from modinfo -p output, it is more useful to run:

```
~]$ modinfo e1000e | grep "^parm" | sort
parm:              copybreak:Maximum size of packet that is copied to a new buffer on receive (uint)
parm:              CrcStripping:Enable CRC Stripping, disable if your BMC needs the CRC (array of int)
parm:              EEE:Enable/disable on parts that support the feature (array of int)
parm:              InterruptThrottleRate:Interrupt Throttling Rate (array of int)
parm:              IntMode:Interrupt Mode (array of int)
parm:              KumeranLockLoss:Enable Kumeran lock loss workaround (array of int)
parm:              RxAbsIntDelay:Receive Absolute Interrupt Delay (array of int)
parm:              RxIntDelay:Receive Interrupt Delay (array of int)
parm:              SmartPowerDownEnable:Enable PHY smart power down (array of int)
parm:              TxAbsIntDelay:Transmit Absolute Interrupt Delay (array of int)
parm:              TxIntDelay:Transmit Interrupt Delay (array of int)
parm:              WriteProtectNVM:Write-protect NVM [WARNING: disabling this can lead to corrupted NVM] (array of int)
```

### 23.3. Loading a Module

To load a kernel module, run modprobe module_name as root. For example, to load the wacom module, run:

```
~]$ modprobe wacom
```

By default, modprobe attempts to load the module from /lib/modules/kernel_version/kernel/drivers/. In this directory, each type of module has its own subdirectory, such as net/ and scsi/, for network and SCSI interface drivers respectively.

Some modules have dependencies, which are other kernel modules that must be loaded before the module in question can be loaded. The modprobe command always takes dependencies into account when performing operations. When you ask modprobe to load a specific kernel module, it first examines the dependencies of that module, if there are any, and loads them if they are not already loaded into the kernel. modprobe resolves dependencies recursively: it will load all dependencies of dependencies, and so on, if necessary, thus ensuring that all dependencies are always met.

You can use the -v (or --verbose) option to cause modprobe to display detailed information about what it is doing, which may include loading module dependencies.
Example 23.3. \texttt{modprobe -v} shows module dependencies as they are loaded

You can load the Fibre Channel over Ethernet module verbosely by typing the following at a shell prompt:

\begin{verbatim}
[-]# modprobe -v fcoe
  insmod /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/scsi/scsi_tgt.ko
  insmod /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/scsi/scsi_transport_fc.ko
  insmod /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/scsi/libfc/libfc.ko
  insmod /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/scsi/fcoe/libfcoe.ko
  insmod /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/scsi/fcoe/fcoe.ko
\end{verbatim}

In this example, you can see that \texttt{modprobe} loaded the \texttt{scsi_tgt}, \texttt{scsi_transport_fc}, \texttt{libfc} and \texttt{libfcoe} modules as dependencies before finally loading \texttt{fcoe}. Also note that \texttt{modprobe} used the more “primitive” \texttt{insmod} command to insert the modules into the running kernel.

\textbf{Always use modprobe instead of insmod!}

Although the \texttt{insmod} command can also be used to load kernel modules, it does not resolve dependencies. Because of this, you should \textit{always} load modules using \texttt{modprobe} instead.

23.4. Unloading a Module

You can unload a kernel module by running \texttt{modprobe -r module_name} as root. For example, assuming that the \texttt{wacom} module is already loaded into the kernel, you can unload it by running:

\begin{verbatim}
[-]# modprobe -r wacom
\end{verbatim}

However, this command will fail if a process is using:

\begin{itemize}
  \item the \texttt{wacom} module,
  \item a module that \texttt{wacom} directly depends on, or,
  \item any module that \texttt{wacom}—through the dependency tree—depends on indirectly.
\end{itemize}

Refer to Section 23.1, “Listing Currently-Loaded Modules” for more information about using \texttt{lsmod} to obtain the names of the modules which are preventing you from unloading a certain module.

Example 23.4. Unloading a kernel module

For example, if you want to unload the \texttt{firewire_ohci} module (because you believe there is a bug in it that is affecting system stability, for example), your terminal session might look similar to this:

\begin{verbatim}
[-]# modinfo -F depends firewire_ohci
  depends: firewire-core
[-]# modinfo -F depends firewire_core
  depends: crc-itu-t
\end{verbatim}
You have figured out the dependency tree (which does not branch in this example) for the loaded Firewire modules: firewire_ohci depends on firewire_core, which itself depends on crc-itu-t.

You can unload firewire_ohci using the `modprobe -v -r module_name` command, where `-r` is short for `--remove` and `-v` for `--verbose`:

```bash
modprobe -r -v firewire_ohci
```

The output shows that modules are unloaded in the reverse order that they are loaded, given that no processes depend on any of the modules being unloaded.

---

**Do not use `rmmod` directly!**

Although the `rmmod` command can be used to unload kernel modules, it is recommended to use `modprobe -r` instead.

---

### 23.5. Setting Module Parameters

Like the kernel itself, modules can also take parameters that change their behavior. Most of the time, the default ones work well, but occasionally it is necessary or desirable to set custom parameters for a module. Because parameters cannot be dynamically set for a module that is already loaded into a running kernel, there are two different methods for setting them.

1. You can unload all dependencies of the module you want to set parameters for, unload the module using `modprobe -r`, and then load it with `modprobe` along with a list of customized parameters. This method is often used when the module does not have many dependencies, or to test different combinations of parameters without making them persistent, and is the method covered in this section.

2. Alternatively, you can list the new parameters in an existing or newly-created file in the `/etc/modprobe.d/` directory. This method makes the module parameters persistent by ensuring that they are set each time the module is loaded, such as after every reboot or `modprobe` command. This method is covered in Section 23.6, “Persistent Module Loading”, though the following information is a prerequisite.

You can use `modprobe` to load a kernel module with custom parameters using the following command line format:

```
modprobe module_name [parameter=value]
```

When loading a module with custom parameters on the command line, be aware of the following:
• You can enter multiple parameters and values by separating them with spaces.

• Some module parameters expect a list of comma-separated values as their argument. When entering the list of values, do not insert a space after each comma, or modprobe will incorrectly interpret the values following spaces as additional parameters.

• The modprobe command silently succeeds with an exit status of 0 if:
  • it successfully loads the module, or
  • the module is already loaded into the kernel.

Thus, you must ensure that the module is not already loaded before attempting to load it with custom parameters. The modprobe command does not automatically reload the module, or alert you that it is already loaded.

Here are the recommended steps for setting custom parameters and then loading a kernel module. This procedure illustrates the steps using the e1000e module, which is the network driver for Intel PRO/1000 network adapters, as an example:

Procedure 23.1. Loading a Kernel Module with Custom Parameters

1. First, ensure the module is not already loaded into the kernel. For example:

```
~]# lsmod | grep e1000e
~]#
```

Output indicates that the module is already loaded into the kernel, in which case you must first unload it before proceeding. Refer to Section 23.4, “Unloading a Module” for instructions on safely unloading it.

2. Load the module and list all custom parameters after the module name. For example, if you wanted to load the Intel PRO/1000 network driver with the interrupt throttle rate set to 3000 interrupts per second for the first, second and third instances of the driver, and Energy Efficient Ethernet (EEE) turned on¹, you would run, as root:

```
~]# modprobe e1000e InterruptThrottleRate=3000,3000,3000 EEE=1
```

This example illustrates passing multiple valued to a single parameter by separating them with commas and omitting any spaces between them.

### 23.6. Persistent Module Loading

As shown in Example 23.1, “Listing information about a kernel module with lsmod”, many kernel modules are loaded automatically at boot time. You can specify additional modules to be loaded by creating a new `file_name.modules` file in the `/etc/sysconfig/modules/` directory, where `file_name` is any descriptive name of your choice. Your `file_name.modules` files are treated by the system startup scripts as shell scripts, and as such should begin with an interpreter directive (also called a “bang line”) as their first line:

```
#!/bin/sh
```

¹ Despite what the example might imply, Energy Efficient Ethernet is turned on by default in the e1000e driver.
Additionally, the `file_name.modules` file should be executable. You can make it executable by running:

```bash
modules]# chmod +x file_name.modules
```

**Example 23.5. /etc/sysconfig/modules/bluez-uinput.modules**
The following `bluez-uinput.modules` script loads the `uinput` module:

```bash
#!/bin/sh
if [ -c /dev/input/uinput ]; then
  exec /sbin/modprobe uinput >/dev/null 2>&1
fi
```

The `if`-conditional statement on the third line ensures that the `/dev/input/uinput` file does not already exist (the `!` symbol negates the condition), and, if that is the case, loads the `uinput` module by calling `exec /sbin/modprobe uinput`. Note that the `uinput` module creates the `/dev/input/uinput` file, so testing to see if that file exists serves as verification of whether the `uinput` module is loaded into the kernel.

The following `>/dev/null 2>&1` clause at the end of that line redirects any output to `/dev/null` so that the `modprobe` command remains quiet.

### 23.7. Specific Kernel Module Capabilities

This section explains how to enable specific kernel capabilities using various kernel modules.

#### 23.7.1. 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 a user-created `module_name.conf` file in the `/etc/modprobe.d/` directory.

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](http://www.redhat.com/mirrors/LDP/HOWTO/Ethernet-HOWTO.html).

#### 23.7.2. Using Channel Bonding

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. Configure a channel bonding interface as outlined in the *Fedora Networking Guide*.

2. 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 23.7.2.1, “Bonding Module Directives” for a list of available options and how to quickly determine the best ones for your bonded interface.
23.7.2.1. Bonding Module Directives

It is a good idea to test which channel bonding module parameters work best for your bonded interfaces before adding them to the BONDING_OPTS="bonding parameters" directive in your bonding interface configuration file (ifcfg-bond0 for example). Parameters to bonded interfaces can be configured without unloading (and reloading) the bonding module by manipulating files in the sysfs file system.

sysfs is a virtual file system that represents kernel objects as directories, files and symbolic links. sysfs can be used to query for information about kernel objects, and can also manipulate those objects through the use of normal file system commands. The sysfs virtual file system has a line in /etc/fstab, and is mounted under the /sys/ directory. All bonding interfaces can be configured dynamically by interacting with and manipulating files under the /sys/class/net/ directory.

In order to determine the best parameters for your bonding interface, create a channel bonding interface file such as ifcfg-bond0 by following the instructions in the Fedora Networking Guide. Insert the SLAVE=yes and MASTER=bond0 directives in the configuration files for each interface bonded to bond0. Once this is completed, you can proceed to testing the parameters.

First, bring up the bond you created by running ifconfig bond0 up as root:

```bash
-]# ifconfig bond0 up
```

If you have correctly created the ifcfg-bond0 bonding interface file, you will be able to see bond0 listed in the output of running ifconfig (without any options):

```
-]# ifconfig
bond0  Link encap:Ethernet  HWaddr 00:00:00:00:00:00
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:0 errors:0 dropped:0 overruns:0 frame:0
       TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:0
       RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
eth0   Link encap:Ethernet  HWaddr 52:54:00:26:9E:F1
       inet addr:192.168.122.251  Bcast:192.168.122.255  Mask:255.255.255.0
       inet6 addr: fe80::5054:ff:fe26:9ef1/64 Scope:Link
       inet addr: fe80::/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:207 errors:0 dropped:0 overruns:0 frame:0
       TX packets:205 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:70374 (68.7 KiB)  TX bytes:25298 (24.7 KiB)

[output truncated]
```

To view all existing bonds, even if they are not up, run:

```bash
-]# cat /sys/class/net/bonding_masters
bond0
```

You can configure each bond individually by manipulating the files located in the /sys/class/net/bond0/bonding/ directory. First, the bond you are configuring must be taken down:

```bash
-]# ifconfig bond0 down
```

As an example, to enable MII monitoring on bond0 with a 1 second interval, you could run (as root):

```bash
-]# echo 1000 > /sys/class/net/bond0/bonding/miimon
```
To configure bond0 for `balance-alb` mode, you could run either:

```
~]# echo 6 > /sys/class/net/bond0/bonding/mode
```

...or, using the name of the mode:

```
~]# echo balance-alb > /sys/class/net/bond0/bonding/mode
```

After configuring options for the bond in question, you can bring it up and test it by running `ifconfig bondW up`. If you decide to change the options, take the interface down, modify its parameters using `sysfs`, bring it back up, and re-test.

Once you have determined the best set of parameters for your bond, add those parameters as a space-separated list to the `BONDING_OPTS=` directive of the `/etc/sysconfig/network-scripts/ifcfg-bondW` file for the bonding interface you are configuring. Whenever that bond is brought up (for example, by the system during the boot sequence if the `ONBOOT=yes` directive is set), the bonding options specified in the `BONDING_OPTS` will take effect for that bond. For more information on configuring bonding interfaces (and `BONDING_OPTS`), refer to the Fedora Networking Guide.

The following list provides the names of many of the more common channel bonding parameters, along with a description of what they do. For more information, refer to the brief descriptions for each `parm` in `modinfo bonding` output, or the exhaustive descriptions in the `bonding.txt` file in the `kernel-doc` package (see Section 23.8, "Additional Resources").

**Bonding Interface Parameters**

`arp_interval=time_in_milliseconds`

Specifies (in milliseconds) how often ARP monitoring occurs.

Make sure you specify all required parameters

It is essential that both `arp_interval` and `arp_ip_target` parameters are specified, or, alternatively, the `miimon` parameter is specified. Failure to do so can cause degradation of network performance in the event that a link fails.

If using this setting while in `mode=0` or `mode=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=ip_address[,ip_address_2,...,ip_address_16]`

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=value`

Validate source/distribution of ARP probes; default is `none`. Other valid values are `active`, `backup`, and `all`.

`debug=number`

Enables debug messages. Possible values are:
- 0 — Debug messages are disabled. This is the default.

- 1 — Debug messages are enabled.

**downdelay=time_in_milliseconds**

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.

**lacp_rate=value**

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.

**miimon=time_in_milliseconds**

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.

**Make sure you specify all required parameters**

It is essential that both **arp_interval** and **arp_ip_target** parameters are specified, or, alternatively, the **miimon** parameter is specified. Failure to do so can cause degradation of network performance in the event that a link fails.

**mode=value**

Allows you to specify the bonding policy. The value can be one of:

- **balance-rr** or 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.

- **active-backup** or 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.
• **balance-xor** or 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.

• **broadcast** or 3 — Sets a broadcast policy for fault tolerance. All transmissions are sent on all slave interfaces.

• **802.3ad** or 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.

• **balance-tlb** or 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.

• **balance-alb** or 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.

`num_unsol_na=number`

- Specifies the number of unsolicited IPv6 Neighbor Advertisements to be issued after a failover event. One unsolicited NA is issued immediately after the failover.

  - The valid range is 0 - 255; the default value is 1. This parameter affects only the active-backup mode.

`primary=interface_name`

- 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.

`primary_reselect=value`

- Specifies the reselection policy for the primary slave. This affects how the primary slave is chosen to become the active slave when failure of the active slave or recovery of the primary slave occurs. This parameter is designed to prevent flip-flopping between the primary slave and other slaves.

  - Possible values are:
    - **always** or 0 (default) — The primary slave becomes the active slave whenever it comes back up.
    - **better** or 1 — The primary slave becomes the active slave when it comes back up, if the speed and duplex of the primary slave is better than the speed and duplex of the current active slave.
    - **failure** or 2 — The primary slave becomes the active slave only if the current active slave fails and the primary slave is up.

  - The `primary_reselect` setting is ignored in two cases:
    - If no slaves are active, the first slave to recover is made the active slave.
• When initially enslaved, the primary slave is always made the active slave.

Changing the `primary_reselect` policy via `sysfs` will cause an immediate selection of the best active slave according to the new policy. This may or may not result in a change of the active slave, depending upon the circumstances.

`updelay=time_in_milliseconds`

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.

`use_carrier=number`

Specifies whether or not `miimon` should use MII/ETHTOOL ioctls or `netif_carrier_ok()` to determine the link state. The `netif_carrier_ok()` function 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.

**Note**

If the bonding interface 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=value`

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 parameter uses the XOR of hardware MAC addresses to generate the hash. The formula used is:

\[
(source\_MAC\_address \text{ XOR } destination\_MAC) \text{ MODULO slave\_count}
\]

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:

\[
(((source\_port \text{ XOR } dest\_port) \text{ XOR } ((source\_IP \text{ XOR } dest\_IP) \text{ AND } 0xffff)) \text{ MODULO slave\_count}
\]
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.

- **2 or layer2+3** — Uses a combination of layer2 and layer3 protocol information to generate the hash.

  Uses XOR of hardware MAC addresses and IP addresses to generate the hash. The formula is:

  $$(((source\_IP \ XOR \ dest\_IP) \ AND \ 0xffff) \ XOR \ (source\_MAC \ XOR \ destination\_MAC)) \ MODULO \ slave\_count$$

  This algorithm will place all traffic to a particular network peer on the same slave. For non-IP traffic, the formula is the same as for the layer2 transmit hash policy.

  This policy is intended to provide a more balanced distribution of traffic than layer2 alone, especially in environments where a layer3 gateway device is required to reach most destinations.

  This algorithm is 802.3ad compliant.

### 23.8. Additional Resources

For more information on kernel modules and their utilities, refer to the following resources.

#### 23.8.1. Installed Documentation

There is a number of manual pages for various utilities related to the kernel modules:

- **man lsmod**
  The manual page for the `lsmod` command.

- **man modinfo**
  The manual page for the `modinfo` command.

- **man modprobe**
  The manual page for the `modprobe` command.

- **man rmmod**
  The manual page for the `rmmod` command.

- **man ethtool**
  The manual page for the `ethtool` command.

- **man mii-tool**
  The manual page for the `mii-tool` command.

Additionally, you can refer to the documentation provided by the `kernel-doc` package:
/usr/share/doc/kernel-doc-kernel_version/Documentation/
This directory contains information on the kernel, kernel modules, and their respective parameters. Note that before accessing the kernel documentation, you must run the following command as root:

```
yum install kernel-doc
```

23.8.2. Useful Websites

*Linux Loadable Kernel Module HOWTO*[^2]

The *Linux Loadable Kernel Module HOWTO* from the Linux Documentation Project contains further information on working with kernel modules.

[^2]: http://tldp.org/HOWTO/Module-HOWTO/
The kdump Crash Recovery Service

When the kdump crash dumping mechanism is enabled, the system is booted from the context of another kernel. This second kernel reserves a small amount of memory and its only purpose is to capture the core dump image in case the system crashes.

Being able to analyze the core dump significantly helps to determine the exact cause of the system failure, and it is therefore strongly recommended to have this feature enabled. This chapter explains how to configure, test, and use the kdump service in Fedora, and provides a brief overview of how to analyze the resulting core dump using the crash debugging utility.

24.1. Installing the kdump Service

In order use the kdump service on your system, make sure you have the kexec-tools package installed. To do so, type the following at a shell prompt as root:

```
yum install kexec-tools
```

For more information on how to install new packages in Fedora, refer to Section 5.2.4, “Installing Packages”.

24.2. Configuring the kdump Service

There are three common means of configuring the kdump service: at the first boot, using the Kernel Dump Configuration graphical utility, and doing so manually on the command line.

- **Disable IOMMU on Intel chipsets**

  A limitation in the current implementation of the Intel IOMMU driver can occasionally prevent the kdump service from capturing the core dump image. To use kdump on Intel architectures reliably, it is advised that the IOMMU support is disabled.

24.2.1. Configuring the kdump at First Boot

When the system boots for the first time, the firstboot application is launched to guide the user through the initial configuration of the freshly installed system. To configure kdump, navigate to the Kdump section and follow the instructions below.
Make sure the system has enough memory

Unless the system has enough memory, this option will not be available. For the information on minimum memory requirements, refer to the `Hardware Overview`\(^1\) section of the `Fedora 19 Release Notes`\(^2\). When the kdump crash recovery is enabled, the minimum memory requirements increase by the amount of memory reserved for it. This value is determined by the user, and defaults to 128 MB plus 64 MB for each TB of physical memory (that is, a total of 192 MB for a system with 1 TB of physical memory).

24.2.1.1. Enabling the Service

To allow the kdump daemon to start at boot time, select the **Enable kdump?** checkbox. This will enable the service and start it for the current session. Similarly, unselecting the checkbox will disable it for and stop the service immediately.

24.2.1.2. Configuring the Memory Usage

To configure the amount of memory that is reserved for the kdump kernel, click the up and down arrow buttons next to the **Kdump Memory** field to increase or decrease the value. Notice that the **Usable System Memory** field changes accordingly showing you the remaining memory that will be available to the system.

24.2.2. Using the Kernel Dump Configuration Utility

To start the **Kernel Dump Configuration** utility, select **Applications → Other → Kernel crash dumps** from the **Activities** menu, or type `system-config-kdump` at a shell prompt. You will be presented with a window as shown in `Figure 24.1, “Basic Settings”`. The utility allows you to configure kdump as well as to enable or disable starting the service at boot time. When you are done, click **Apply** to save the changes. The system reboot will be requested, and unless you are already authenticated, you will be prompted to enter the superuser password.

---


Make sure the system has enough memory

Unless the system has enough memory, this option will not be available. For the information on minimum memory requirements, refer to the *Hardware Overview*³ section of the *Fedora 19 Release Notes*⁴. When the kdump crash recovery is enabled, the minimum memory requirements increase by the amount of memory reserved for it. This value is determined by the user, and defaults to 128 MB plus 64 MB for each TB of physical memory (that is, a total of 192 MB for a system with 1 TB of physical memory).

24.2.2.1. Enabling the Service

To start the kdump daemon at boot time, click the **Enable** button on the toolbar. This will enable the service and start it for the current session. Similarly, clicking the **Disable** button will disable it and stop the service immediately.

For more information on system services and their configuration, refer to *Chapter 7, Services and Daemons*.

24.2.2.2. The Basic Settings Tab

The **Basic Settings** tab enables you to configure the amount of memory that is reserved for the kdump kernel. To do so, select the **Manual kdump memory settings** radio button, and click the up and down arrow buttons next to the **New kdump Memory** field to increase or decrease the value. Notice that the **Usable Memory** field changes accordingly showing you the remaining memory that will be available to the system.

![Figure 24.1. Basic Settings](http://docs.fedoraproject.org/en-US/Fedora/19/html/Release_Notes/sect-Release_Notes-Welcome_to_Fedora_.html#hardware_overview)


### 24.2.2.3. The Target Settings Tab

The **Target Settings** tab enables you to specify the target location for the `vmcore` dump. It can be either stored as a file in a local file system, written directly to a device, or sent over a network using the NFS (Network File System) or SSH (Secure Shell) protocol.

![Target Settings Tab](image)

**Figure 24.2. Target Settings**

To save the dump to the local file system, select the **Local filesystem** radio button. Optionally, you can customize the settings by choosing a different partition from the **Partition**, and a target directory from the **Path** pulldown lists.

To write the dump directly to a device, select the **Raw device** radio button, and choose the desired target device from the pulldown list next to it.

To store the dump to a remote machine, select the **Network** radio button. To use the NFS protocol, select the **NFS** radio button, and fill the **Server name** and **Path to directory** fields. To use the SSH protocol, select the **SSH** radio button, and fill the **Server name**, **Path to directory**, and **User name** fields with the remote server address, target directory, and a valid remote user name respectively. Refer to [Chapter 9, OpenSSH](#) for information on how to configure an SSH server, and how to set up a key-based authentication.

For a complete list of currently supported targets, see Table 24.1, “Supported kdump targets”.

#### Table 24.1. Supported kdump targets

<table>
<thead>
<tr>
<th>Type</th>
<th>Supported Targets</th>
<th>Unsupported Targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw device</td>
<td>All locally attached raw disks and partitions.</td>
<td>—</td>
</tr>
<tr>
<td>Local file system</td>
<td><code>ext2</code>, <code>ext3</code>, <code>ext4</code>, <code>minix</code> file systems on directly attached disk drives, hardware RAID logical drives, LVM devices, and mdraid arrays.</td>
<td>The eCryptfs file system.</td>
</tr>
<tr>
<td>Type</td>
<td>Supported Targets</td>
<td>Unsupported Targets</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Remote directories accessed using the iSCSI protocol over hardware initiators.</td>
<td>Remote directories accessed using the iSCSI protocol over software initiators.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Remote directories accessed using the SMB/CIFS protocol.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Remote directories accessed using the FCoE (Fibre Channel over Ethernet) protocol.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Remote directories accessed using wireless network interfaces.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Multipath-based storages.</td>
</tr>
</tbody>
</table>

24.2.2.4. The Filtering Settings Tab

The **Filtering Settings** tab enables you to select the filtering level for the `vmcore` dump.

![Filtering Settings](image)

To exclude the `zero page`, `cache page`, `cache private`, `user data`, or `free page` from the dump, select the checkbox next to the appropriate label.

24.2.2.5. The Expert Settings Tab

The **Expert Settings** tab enables you to choose which kernel and initial RAM disk to use, as well as to customize the options that are passed to the kernel and the core collector program.
24.2.3. Configuring kdump on the Command Line

24.2.3.1. Configuring the Memory Usage

To configure the amount of memory that is reserved for the kdump kernel, as root, edit the /etc/default/grub file and add the crashkernel=<size>M (or crashkernel=auto) parameter to the list of kernel options (the GRUB_CMDLINE_LINUX line). For example, to reserve 128 MB of memory, use:

```
GRUB_CMDLINE_LINUX="crashkernel=128M quiet rhgb"
```
Then update the configuration file for the GRUB 2 boot loader by typing the following at a shell prompt as root:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

**Make sure the system has enough memory**

Unless the system has enough memory, this option will not be available. For the information on minimum memory requirements, refer to the "Hardware Overview" section of the [Fedora 19 Release Notes](http://docs.fedoraproject.org/en-US/Fedora/19/html/Release_Notes/sect-Release_Notes-Welcome_to_Fedora_.html#hardware_overview). When the kdump crash recovery is enabled, the minimum memory requirements increase by the amount of memory reserved for it. This value is determined by the user, and defaults to 128 MB plus 64 MB for each TB of physical memory (that is, a total of 192 MB for a system with 1 TB of physical memory).

**Using the crashkernel=auto parameter**

In Fedora 20, the `crashkernel=auto` only reserves memory if the system has 4 GB of physical memory or more.

24.2.3.2. Configuring the Target Type

When a kernel crash is captured, the core dump can be either stored as a file in a local file system, written directly to a device, or sent over a network using the NFS (Network File System) or SSH (Secure Shell) protocol. Only one of these options can be set at the moment, and the default option is to store the `vmcore` file in the `/var/crash/` directory of the local file system. To change this, as root, open the `/etc/kdump.conf` configuration file in a text editor and edit the options as described below.

To change the local directory in which the core dump is to be saved, remove the hash sign (“#”) from the beginning of the `#path /var/crash` line, and replace the value with a desired directory path. Optionally, if you wish to write the file to a different partition, follow the same procedure with the `#ext4 /dev/sda3` line as well, and change both the file system type and the device (a device name, a file system label, and UUID are all supported) accordingly. For example:

```
ext3 /dev/sda4
path /usr/local/cores
```

To write the dump directly to a device, remove the hash sign (“#”) from the beginning of the `#raw /dev/sda5` line, and replace the value with a desired device name. For example:

```
raw /dev/sdb1
```
Chapter 24. The kdump Crash Recovery Service

To store the dump to a remote machine using the NFS protocol, remove the hash sign (“#”) from the beginning of the `#net my.server.com:/export/tmp` line, and replace the value with a valid hostname and directory path. For example:

```
net penguin.example.com:/export/cores
```

To store the dump to a remote machine using the SSH protocol, remove the hash sign (“#”) from the beginning of the `#net user@my.server.com` line, and replace the value with a valid username and hostname. For example:

```
net john@penguin.example.com
```

Refer to Chapter 9, OpenSSH for information on how to configure an SSH server, and how to set up a key-based authentication.

For a complete list of currently supported targets, see Table 24.1, “Supported kdump targets”.

### 24.2.3.3. Configuring the Core Collector

To reduce the size of the vmcore dump file, kdump allows you to specify an external application (that is, a core collector) to compress the data, and optionally leave out all irrelevant information. Currently, the only fully supported core collector is `makedumpfile`.

To enable the core collector, as root, open the `/etc/kdump.conf` configuration file in a text editor, remove the hash sign (“#”) from the beginning of the `#core_collector makedumpfile -c --message-level 1 -d 31` line, and edit the command line options as described below.

To enable the dump file compression, add the `-c` parameter. For example:

```
core_collector makedumpfile -c
```

To remove certain pages from the dump, add the `-d value` parameter, where `value` is a sum of values of pages you want to omit as described in Table 24.2, “Supported filtering levels”. For example, to remove both zero and free pages, use the following:

```
core_collector makedumpfile -d 17 -c
```

Refer to the manual page for `makedumpfile` for a complete list of available options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Zero pages</td>
</tr>
<tr>
<td>2</td>
<td>Cache pages</td>
</tr>
<tr>
<td>4</td>
<td>Cache private</td>
</tr>
<tr>
<td>8</td>
<td>User pages</td>
</tr>
<tr>
<td>16</td>
<td>Free pages</td>
</tr>
</tbody>
</table>

### 24.2.3.4. Changing the Default Action


By default, when kdump fails to create a core dump, the root file system is mounted and `/sbin/init` is run. To change this behavior, as root, open the `/etc/kdump.conf` configuration file in a text editor, remove the hash sign (`#`) from the beginning of the `#default shell` line, and replace the value with a desired action as described in Table 24.3, “Supported actions”.

Table 24.3. Supported actions

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reboot</td>
<td>Reboot the system, losing the core in the process.</td>
</tr>
<tr>
<td>halt</td>
<td>Halt the system.</td>
</tr>
<tr>
<td>poweroff</td>
<td>Power off the system.</td>
</tr>
<tr>
<td>shell</td>
<td>Run the <code>msh</code> session from within the initramfs, allowing a user to record the core manually.</td>
</tr>
</tbody>
</table>

For example:

```
default halt
```

24.2.3.5. Enabling the Service

To start the `kdump` daemon at boot time, type the following at a shell prompt as root:

```
systemctl enable kdump.service
```

Similarly, typing `systemctl disable kdump.service` will disable it. To start the service in the current session, use the following command as root:

```
systemctl start kdump.service
```

For more information on services and their configuration, refer to Chapter 7, Services and Daemons.

24.2.4. Testing the Configuration

Be careful when using these commands

The commands below will cause the kernel to crash. Use caution when following these steps, and by no means use them on a production machine.

To test the configuration, reboot the system with `kdump` enabled, and make sure that the service is running (refer to Section 7.2, “Running Services” for more information on how to run a service in Fedora):

```
systemctl is-active kdump.service
```

Then type the following commands at a shell prompt:
This will force the Linux kernel to crash, and the `address-YYYY-MM-DD-HH:MM:SS/vmcore` file will be copied to the location you have selected in the configuration (that is, to `/var/crash/` by default).

## 24.3. Analyzing the Core Dump

To determine the cause of the system crash, you can use the `crash` utility, which provides an interactive prompt very similar to the GNU Debugger (GDB). This utility allows you to interactively analyze a running Linux system as well as a core dump created by netdump, diskdump, xendump, or kdump.

### Make sure you have relevant packages installed

To analyze the `vmcore` dump file, you must have the `crash` and `kernel-debuginfo` packages installed. To install these packages, type the following at a shell prompt as `root`:

```bash
yum install crash debuginfo-install kernel
```

For more information on how to install new packages in Fedora, refer to Section 5.2.4, “Installing Packages”.

### 24.3.1. Running the crash Utility

To start the utility, type the command in the following form at a shell prompt:

```bash
crash /usr/lib/debug/lib/modules/kernell/vmlinux /var/crash/timestamp/vmcore
```

Note that the `kernel` version should be the same that was captured by kdump. To find out which kernel you are currently running, use the `uname -r` command.

### Example 24.1. Running the crash utility

```
[-]# crash /usr/lib/debug/lib/modules/2.6.32-69.el6.i686/vmlinuc \ /var/crash/127.0.0.1-2010-08-25-08:45:02/vmcore
```

```
crash 5.0.0-23.el6
Copyright (C) 2002-2010 Red Hat, Inc.
Copyright (C) 2004, 2005, 2006 IBM Corporation
Copyright (C) 1999-2006 Hewlett-Packard Co
Copyright (C) 2005, 2006 Fujitsu Limited
Copyright (C) 2006, 2007 VA Linux Systems Japan K.K.
Copyright (C) 2005 NEC Corporation
This program is free software, covered by the GNU General Public License,
and you are welcome to change it and/or distribute copies of it under
```
certain conditions. Enter "help copying" to see the conditions.
This program has absolutely no warranty. Enter "help warranty" for details.

GNU gdb (GDB) 7.0
Copyright (C) 2009 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-pc-linux-gnu"

KERNEL: /usr/lib/debug/lib/modules/2.6.32-69.el6.i686/vmlinux
DUMPFILE: /var/crash/127.0.0.1-2010-08-25-08:45:02/vmcore [PARTIAL DUMP]
CPUS: 4
UPTIME: 00:09:02
LOAD AVERAGE: 0.00, 0.01, 0.00
TASKS: 140
NODENAME: hp-dl320g5-02.lab.bos.redhat.com
RELEASE: 2.6.32-69.el6.i686
VERSION: #1 SMP Tue Aug 24 10:31:45 EDT 2010
MACHINE: i686 (2394 Mhz)
MEMORY: 8 GB
PANIC: "Oops: 0002 [#1] SMP " (check log for details)
PID: 5591
COMMAND: "bash"
OFFSET: 0
CPU: 2
STATE: TASK_RUNNING (PANIC)

24.3.2. Displaying the Message Buffer

To display the kernel message buffer, type the log command at the interactive prompt.

Example 24.2. Displaying the kernel message buffer

```
crash> log
... several lines omitted ...
EIP: 0060:[<c068124f>] EFLAGS: 00010096 CPU: 2
EIP is at sysrq_handle_crash+0xf/0x20
EAX: 00000063 EBX: 00000063 ECX: c09e1c8c EDX: 00000000
ESI: c0a09ca0 EDI: 00000286 EBP: 00000000 ESP: ef4dbf9c
DS: 007b ES: 007b FS: 00d8 GS: 00e0 SS: 0068
Process bash (pid: 5591, ti=ef4da000 task=f196d560 task.ti=ef4da000)
Stack:
c068146b c06008b1 c0068653 00000003 00000000 00000002 efade5c0 c06814d0
<0> fffffffb c061850f b7776000 f260a40 c0569e6c ef4dbf9c 00000002 b7776000
<0> efade5c0 00000002 b7776000 c0569e60 c051de50 ef4dbf9c f196d560 ef4dbf4
Call Trace:
[<c068146b>] __handle_sysrq+0xf/0x160
[<c06814d0>] write_sysrq_trigger+0x0/0x50
[<c068150f>] write_sysrq_trigger+0x3f/0x50
[<c0569e4a>] proc_reg_write+0x64/0xa0
[<c0569e60>] proc_reg_write+0x0/0xa0
[<c051de50>] vfs_write+0xa0/0x190
[<c05e8d1>] sys_write+0x41/0x70
[<c049a0ad>] syscall_call+0x7/0xb
Code: a0 c0 01 0f b6 41 03 19 d2 f7 d2 83 e2 03 83 e0 cf c1 e2 04 09 d0 88 41 03 f3 c3 90
c7 05 c8 1b 9e c0 01 00 00 00 0f ae f8 89 f6 <c6> 05 00 00 00 00 01 c3 89 f6 8d bc 27 00
00 00 00 8d 50 d0 83
EIP: [<c068124f>] sysrq_handle_crash+0xf/0x20 SS:ESP 0068:ef4dbf24
```
Chapter 24. The kdump Crash Recovery Service

24.3.3. Displaying a Backtrace

To display the kernel stack trace, type the `bt` command at the interactive prompt. You can use `bt pid` to display the backtrace of the selected process.

Example 24.3. Displaying the kernel stack trace

```
crash> bt
PID: 5591  TASK: f196d560  CPU: 2  COMMAND: "bash"
#0 [ef4dbdcc] crash_kexec at c0494922
#1 [ef4dbe20] oops_end at c080e402
#2 [ef4dbe34] no_context at c043089d
#3 [ef4dbe58] bad_area at c0430b26
#4 [ef4dbe6c] do_page_fault at c080fb9b
#5 [ef4dbeec] error_code (via page_fault) at c080d809
   EAX: 00000063  EBX: 00000063  ECX: c09e1c8c  EDX: 00000000  EBP: 00000000
   DS: 007b  ESI: c0a09ca0  ES: 007b  EDI: 0000286  GS: 00e0
   CS: 0060  EIP: c068124f  ERR: ffffffff  EFLAGS: 00010096
#6 [ef4dbf18] sysrq_handle_crash at c068124f
#7 [ef4dbf24] _handle_sysrq at c0681469
#8 [ef4dbf48] write_sysrq_trigger at c068150a
#9 [ef4dbf54] proc_reg_write at c0569ec2
#10 [ef4dbf74] vfs_write at c051de4e
#11 [ef4dbf94] sys_write at c051e8cc
#12 [ef4dbfb0] system_call at c0400ad5
   EAX: ffffffff  EBX: 00000001  ECX: b7776000  EDX: 00000002
   DS: 007b  ESI: 00000002  ES: 007b  EDI: b7776000
   SS: 007b  ESP: bfcb2088  EBP: bfcb20b4  GS: 0033
   CS: 0073  EIP: 00edc416  ERR: 00000004  EFLAGS: 00000246
```

Type `help bt` for more information on the command usage.

24.3.4. Displaying a Process Status

To display status of processes in the system, type the `ps` command at the interactive prompt. You can use `ps pid` to display the status of the selected process.

Example 24.4. Displaying status of processes in the system

```
crash> ps
    PID  PPID  CPU TASK  ST %MEM  VSZ     RSS COMM
>  0    0    0  c09dc560 RU  0.0   0  0  [swapper]
>  0    0    1  f7072030 RU  0.0   0  0  [swapper]
>  0    0    2  f70a3a90 RU  0.0   0  0  [swapper]
>  0    0    3  f70ac560 RU  0.0   0  0  [swapper]
  0    1    1  f705b6a90 IN  0.0  2828   1424  init
... several lines omitted ...  
  5566   1   1  f2592560 IN  0.0   12876   784  auditd
  5567   1   2  ef427560 IN  0.0   12876   784  auditd
  5587   5132   0  f196d030 IN  0.0   11064   3184 sshd
>  5591  5587   2  f196d560 RU  0.0   5084  1648  bash
```
Type `help ps` for more information on the command usage.

### 24.3.5. Displaying Virtual Memory Information

To display basic virtual memory information, type the `vm` command at the interactive prompt. You can use `vm pid` to display information on the selected process.

**Example 24.5. Displaying virtual memory information of the current context**

```
crash> vm
PID: 5591   TASK: f196d560  CPU: 2   COMMAND: "bash"

<table>
<thead>
<tr>
<th>MM</th>
<th>PGD</th>
<th>RSS</th>
<th>TOTAL_VM</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1bb5900</td>
<td>ef9c6000</td>
<td>1648k</td>
<td>5084k</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VMA</th>
<th>START</th>
<th>END</th>
<th>FLAGS</th>
<th>FILE</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1bb5900</td>
<td>242000</td>
<td>260000</td>
<td>8000005</td>
<td>/lib/ld-2.12.so</td>
</tr>
<tr>
<td>f26af0b8</td>
<td>260000</td>
<td>261000</td>
<td>8000007</td>
<td>/lib/libc-2.12.so</td>
</tr>
<tr>
<td>efb275c</td>
<td>261000</td>
<td>262000</td>
<td>8000007</td>
<td>/lib/libc-2.12.so</td>
</tr>
<tr>
<td>efb2a18</td>
<td>268000</td>
<td>3ed000</td>
<td>8000007</td>
<td>/lib/libc-2.12.so</td>
</tr>
<tr>
<td>efb23d8</td>
<td>3ed000</td>
<td>3ee000</td>
<td>8000007</td>
<td>/lib/libc-2.12.so</td>
</tr>
<tr>
<td>efb2888</td>
<td>3ee000</td>
<td>3f0000</td>
<td>8100007</td>
<td>/lib/libc-2.12.so</td>
</tr>
<tr>
<td>efb2cd4</td>
<td>3f0000</td>
<td>3f1000</td>
<td>8100007</td>
<td>/lib/libc-2.12.so</td>
</tr>
<tr>
<td>efb243c</td>
<td>3f1000</td>
<td>3f4000</td>
<td>100073</td>
<td></td>
</tr>
<tr>
<td>efb28ec</td>
<td>3f4000</td>
<td>3f9000</td>
<td>8000007</td>
<td>/lib/libdl-2.12.so</td>
</tr>
<tr>
<td>efb2568</td>
<td>3f9000</td>
<td>3fa000</td>
<td>8100007</td>
<td>/lib/libdl-2.12.so</td>
</tr>
<tr>
<td>efb2fc2</td>
<td>3fa000</td>
<td>3fb000</td>
<td>8100007</td>
<td>/lib/libdl-2.12.so</td>
</tr>
<tr>
<td>f26af888</td>
<td>7e6000</td>
<td>7fc000</td>
<td>8000007</td>
<td>/lib/libinfo.so.5.7</td>
</tr>
<tr>
<td>f26aff2c</td>
<td>7fc000</td>
<td>7ff000</td>
<td>8100007</td>
<td>/lib/libinfo.so.5.7</td>
</tr>
<tr>
<td>efb211c</td>
<td>d83000</td>
<td>d8f000</td>
<td>8000007</td>
<td>/lib/libnss_files-2.12.so</td>
</tr>
<tr>
<td>efb2504</td>
<td>d8f000</td>
<td>d90000</td>
<td>8100007</td>
<td>/lib/libnss_files-2.12.so</td>
</tr>
<tr>
<td>efb2950</td>
<td>d90000</td>
<td>d91000</td>
<td>8100007</td>
<td>/lib/libnss_files-2.12.so</td>
</tr>
<tr>
<td>f26afe00</td>
<td>edc000</td>
<td>edd000</td>
<td>4040005</td>
<td></td>
</tr>
<tr>
<td>f1bb018a</td>
<td>8047000</td>
<td>8118000</td>
<td>8001875</td>
<td>/bin/bash</td>
</tr>
<tr>
<td>f1bb01e4</td>
<td>8118000</td>
<td>811d000</td>
<td>8101873</td>
<td>/bin/bash</td>
</tr>
<tr>
<td>f1bb0c70</td>
<td>811d000</td>
<td>8122000</td>
<td>100073</td>
<td></td>
</tr>
<tr>
<td>f26af6ea0</td>
<td>9f9d000</td>
<td>9ff9000</td>
<td>100073</td>
<td></td>
</tr>
</tbody>
</table>
```

Type `help vm` for more information on the command usage.

### 24.3.6. Displaying Open Files

To display information about open files, type the `files` command at the interactive prompt. You can use `files pid` to display files opened by the selected process.

**Example 24.6. Displaying information about open files of the current context**

```
crash> files
PID: 5591   TASK: f196d560  CPU: 2   COMMAND: "bash"
ROOT: /      CWD: /root

<table>
<thead>
<tr>
<th>FD</th>
<th>FILE</th>
<th>DENTRY</th>
<th>INODE</th>
<th>TYPE</th>
<th>PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>f734f640</td>
<td>eedc2c6c</td>
<td>eecd6048</td>
<td>CHR</td>
<td>/pts/0</td>
</tr>
<tr>
<td>1</td>
<td>efa56c0</td>
<td>eee14000</td>
<td>f00431d4</td>
<td>REG</td>
<td>/proc/sysrq-trigger</td>
</tr>
<tr>
<td>2</td>
<td>f734f640</td>
<td>eedc2c6c</td>
<td>eecd6048</td>
<td>CHR</td>
<td>/pts/0</td>
</tr>
<tr>
<td>10</td>
<td>f734f640</td>
<td>eedc2c6c</td>
<td>eecd6048</td>
<td>CHR</td>
<td>/pts/0</td>
</tr>
<tr>
<td>255</td>
<td>f734f640</td>
<td>eedc2c6c</td>
<td>eecd6048</td>
<td>CHR</td>
<td>/pts/0</td>
</tr>
</tbody>
</table>
```

477
Chapter 24. The kdump Crash Recovery Service

Type help files for more information on the command usage.

24.3.7. Exiting the Utility
To exit the interactive prompt and terminate crash, type exit or q.

Example 24.7. Exiting the crash utility

```bash
crash> exit
~]#
```

24.4. Additional Resources

24.4.1. Installed Documentation

- **kdump.conf**(5) — a manual page for the /etc/kdump.conf configuration file containing the full documentation of available options.
- **makedumpfile**(8) — a manual page for the makedumpfile core collector.
- **kexec**(8) — a manual page for kexec.
- **crash**(8) — a manual page for the crash utility.
- **/usr/share/doc/kexec-tools-version/kexec-kdump-howto.txt** — an overview of the kdump and kexec installation and usage.

24.4.2. Useful Websites

- [https://access.redhat.com/kb/docs/DOC-6039](https://access.redhat.com/kb/docs/DOC-6039)  
  The Red Hat Knowledgebase article about the kexec and kdump configuration.
- [https://access.redhat.com/kb/docs/DOC-45183](https://access.redhat.com/kb/docs/DOC-45183)  
  The Red Hat Knowledgebase article about supported kdump targets.
- [http://people.redhat.com/anderson/](http://people.redhat.com/anderson/)  
  The crash utility homepage.
Appendix A. 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.

The RPM package format has been improved for Fedora 20. RPM packages are now compressed using the XZ lossless data compression format, which has the benefit of greater compression and less CPU usage during decompression, and support multiple strong hash algorithms, such as SHA-256, for package signing and verification.

Use Yum Instead of RPM Whenever Possible

For most package management tasks, the Yum package manager offers equal and often greater capabilities and utility than RPM. Yum also performs and tracks complicated system dependency resolution, and will complain and force system integrity checks if you use RPM as well to install and remove packages. For these reasons, it is highly recommended that you use Yum instead of RPM whenever possible to perform package management tasks. Refer to Chapter 5, Yum.

If you prefer a graphical interface, you can use the PackageKit GUI application, which uses Yum as its back end, to manage your system’s packages. Refer to Chapter 6, PackageKit for details.

Install RPM packages with the correct architecture!

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.
Running rpm commands must be performed as root

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

A.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.

A.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 A.5, "Additional Resources" for more information on RPM.
A.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 Red Hat RPM packages, 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 5, Yum for details on how to use the official Fedora package repositories.
- The active Fedora mirrors contains many installable RPMs: http://mirrors.fedoraproject.org/publiclist/.
- Unofficial, third-party repositories not affiliated with The Fedora Project also provide RPM packages.

**Third-party repositories and package compatibility**

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.

A.2.2. Installing and Upgrading

RPM packages typically have file names like `tree-1.5.3-2.fc20.x86_64.rpm`. The file name includes the package name (`tree`), version (`1.5.3`), release (`2`), operating system major version (`fc20`) and CPU architecture (`x86_64`).

You can use `rpm`'s `-U` option to:

- upgrade an existing but older package on the system to a newer version, or
- install the package even if an older version is not already installed.

That is, `rpm -U <rpm_file>` is able to perform the function of either upgrading or installing as is appropriate for the package.

Assuming the `tree-1.5.3-2.fc20.x86_64.rpm` package is in the current directory, log in as root and type the following command at a shell prompt to either upgrade or install the `tree` package as determined by `rpm`:

```
rpm -Uvh tree-1.5.3-2.fc20.x86_64.rpm
```
Use -Uvh for nicely-formatted RPM installs

The -v and -h options (which are combined with -U) cause rpm to print more verbose output and display a progress meter using hash signs.

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

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

Always use the -i (install) option to install new kernel packages!

rpm provides two different options for installing packages: the aforementioned -U option (which historically stands for upgrade), and the -i option, historically standing for install. Because the -U option subsumes both install and upgrade functions, we recommend to use rpm -Uvh with all packages except kernel packages.

You should always use the -i option to simply install a new kernel package instead of upgrading it. This is because using the -U option to upgrade a kernel package removes the previous (older) kernel package, which could render the system unable to boot if there is a problem with the new kernel. Therefore, use the rpm -i <kernel_package> command to install a new kernel without replacing any older kernel packages. For more information on installing kernel packages, refer to Chapter 22, Manually Upgrading the Kernel.

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.fc20.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.fc20.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.fc20.x86_64.rpm: Header V3 RSA/SHA1 signature: NOKEY, key ID 57bbccba
```

Refer to Section A.3, “Checking a Package's Signature” for more information on checking a package's signature.

A.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-1.5.3-2.fc20.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 -Uvh --replacepkgs tree-1.5.3-2.fc20.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.

### A.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.fc20.x86_64 conflicts with file from package bar-3.1.1.fc20.x86_64
```

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

```
rpm -Uvh --replacefiles foo-1.0-1.fc20.x86_64.rpm
```

### A.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.fc20.x86_64
```

If you are installing a package from the Fedora installation media, such as from a CD-ROM or DVD, the dependencies may be available. Find the suggested package(s) on the Fedora installation media or on one of the active Fedora mirrors and add it to the command:

```
rpm -Uvh foo-1.0-1.fc20.x86_64.rpm     bar-3.1.1.fc20.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%]
```

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.fc20.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.

### A.2.3. Configuration File Changes

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.fc20.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.fc20.x86_64.rpm
```

### A.2.4. Uninstalling

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

```
rpm -e foo
```
rpm -e and package name errors

Notice that we used the package name `foo`, not the name of the original package file, `foo-1.0-1.fc20.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.fc20.x86_64
  libgs.so.8()(64bit) is needed by (installed) foomatic-4.0.3-1.fc20.x86_64
  libijs-0.35.so()(64bit) is needed by (installed) gutenprint-5.2.4-5.fc20.x86_64
  ghostscript is needed by (installed) printer-filters-1.1-4.fc20.noarch
```

Similar to how we searched for a shared object library (i.e. a `<library_name>.so.<number>` file) in Section A.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.fc20.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.

A.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.fc20.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.
Appendix A. RPM

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.

A.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.fc20.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.

**A.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.fc20.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)
Appendix A. RPM

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.

A.3. Checking a Package's Signature

To verify that a package has not been corrupted or tampered with, examine the checksum by typing the following command at a shell prompt (where `<rpm_file>` is the file name of the RPM package):

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

If the message `<rpm_file>`: `sha1 md5 OK` (specifically the `OK` part of it) is displayed, 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 default, 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.

A.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
```

A.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 file name of the RPM package):
rpm -K <rpm_file>

If all goes well, the following message is displayed: \texttt{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 \url{http://fedoraproject.org/en/keys}.

\section*{A.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.

\begin{itemize}
  \item 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:

\begin{verbatim}
  rpm -Va
\end{verbatim}

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.

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

\begin{verbatim}
  rpm -qf /usr/bin/ghostscript
\end{verbatim}

The output would look like the following:

\begin{verbatim}
  ghostscript-8.70-1.fc20.x86_64
\end{verbatim}

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

\begin{verbatim}
  rpm -Vf /usr/bin/paste
\end{verbatim}

and the appropriate package is verified.

  \item 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:

\begin{verbatim}
  rpm -qdf /usr/bin/free
\end{verbatim}
\end{itemize}
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/TODD0
/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.fc20.noarch.rpm
```

The output would be similar to the following:

```
Name        : crontabs                     Relocations: (not relocatable)
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 crone 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.fc20.noarch.rpm
```

The output is similar to the following:

```
/etc/cron.daily
```
These are just a few examples. As you use RPM, you may find more uses for it.

## A.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.

### A.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.

### A.5.2. Useful Websites


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

### A.5.3. Related Books

*Maximum RPM* — [http://www.rpm.org/max-rpm/](http://www.rpm.org/max-rpm/)

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

---

1. [http://www.redhat.com/mailman/listinfo/rpm-list](http://www.redhat.com/mailman/listinfo/rpm-list)
Appendix B. 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, rapid-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 runs on myriad operating systems and platforms.

The X Window System uses a client-server architecture. Its main purpose is to provide network transparent window system, which runs on a wide range of computing and graphics machines. 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.

B.1. The X Server

Fedora 20 uses X server version, which includes several video drivers, EXA, and platform support enhancements over the previous release, among others. In addition, this release includes several automatic configuration features for the X server, as well as the generic input driver, evdev, that supports all input devices that the kernel knows about, including most mice and keyboards.

X11R7.1 was the first release to take specific advantage of the modularization of the X Window System. With it, X is split into logically distinct modules, which make it easier for open source developers to contribute code to the system.

In the current release, all libraries, headers, and binaries live under the /usr/ directory. The /etc/X11/ directory contains configuration files for X client and server applications. This includes configuration files for the X server itself, 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 information on configuring and adding fonts, refer to Section B.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 is able to automatically detect most of the hardware that it runs on and configure itself accordingly. Alternatively, hardware can be manually specified in configuration files.

The Fedora system installer, Anaconda, installs and configures X automatically, unless the X packages are not selected for installation. If there are any changes to the monitor, video card or other devices managed by the X server, most of the time, X detects and reconfigures these changes automatically. In rare cases, X must be reconfigured manually.

B.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 available with Fedora, from the rudimentary Tab Window Manager (twm) to the
highly developed and interactive desktop environment (such as GNOME or KDE) 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 window manager and a desktop environment.

B.2.1. Desktop Environments

A desktop environment integrates various X clients to create a common graphical user environment
and a 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.

B.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.

The Fedora repositories provide five different window managers:

- **metacity**
  
  The Metacity window manager is the default window manager for GNOME. It is a simple and
efficient window manager which supports custom themes. This window manager is automatically
  pulled in as a dependency when the GNOME desktop is installed.

- **kwin**
  
  The KWin window manager is the default window manager for KDE. It is an efficient window
  manager which supports custom themes. This window manager is automatically pulled in as a
  dependency when the KDE desktop is installed.

- **compiz**
  
  The Compiz compositing window manager is based on OpenGL and can use 3D graphics
  hardware to create fast compositing desktop effects for window management. Advanced features,
such as a cube workspace, are implemented as loadable plug-ins. To run this window manager,
you need to install the compiz package.

- **mwm**
  
  The Motif Window Manager (mwm) is a basic, stand-alone window manager. Since it is designed
to be stand-alone, it should not be used in conjunction with GNOME or KDE. To run this window
  manager, you need to install the openmotif package.
B.3. X Server Configuration Files

The X server is a single binary executable /usr/bin/Xorg; a symbolic link X pointing to this file is also provided. Associated configuration files are stored in the /etc/X11/ and /usr/share/X11/ directories.

The X Window System supports two different configuration schemes. Configuration files in the xorg.conf.d directory contain preconfigured settings from vendors and from distribution, and these files should not be edited by hand. Configuration in the xorg.conf file, on the other hand, is done completely by hand but is not necessary in most scenarios.

When do you need the xorg.conf file?

All necessary parameters for a display and peripherals are auto-detected and configured during installation. The configuration file for the X server, /etc/X11/xorg.conf, that was necessary in previous releases, is not supplied with the current release of the X Window System. It can still be useful to create the file manually to configure new hardware, to set up an environment with multiple video cards, or for debugging purposes.

The /usr/lib/xorg/modules/ (or /usr/lib64/xorg/modules/) directory 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.

When Fedora 20 is installed, the configuration files for X are created using information gathered about the system hardware during the installation process by the HAL (Hardware Abstraction Layer) configuration back end. Whenever the X server is started, it asks HAL for the list of input devices and adds each of them with their respective driver. Whenever a new input device is plugged in, or an existing input device is removed, HAL notifies the X server about the change. Because of this notification system, devices using the mouse, kbd, or vmmouse driver configured in the xorg.conf file are, by default, ignored by the X server. Refer to Section B.3.3.3, “The ServerFlags section” for further details. Additional configuration is provided in the /etc/X11/xorg.conf.d/ directory and it can override or augment any configuration that has been obtained through HAL.

B.3.1. The Structure of the Configuration

The format of the X configuration files 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. Some of these are sometimes enclosed in double quotes (").

Some options within the /etc/X11/xorg.conf file accept a boolean switch which turns the feature on or off. The acceptable values are:

- **1, on, true, or yes** — Turns the option on.
Appendix B. The X Window System

- **0, off, false, or no** — Turns the option off.

The following shows a typical configuration file for the keyboard. Lines beginning with a hash sign (#) are not read by the X server and are used for human-readable comments.

```plaintext
# This file is autogenerated by system-setup-keyboard. Any # modifications will be lost.

Section "InputClass"
  Identifier  "system-setup-keyboard"
  MatchIsKeyboard  "on"
  Option    "XkbModel"  "pc105"
  Option    "XkbLayout" "cz,us"
  Option    "XkbVariant" "(null)"
  Option    "XkbOptions"  "terminate:ctrl_alt_bksp,grp:shifts_toggle,grp_led:scroll"
EndSection
```

B.3.2. The `xorg.conf.d` Directory

The X server supports two configuration directories. The `/usr/share/X11/xorg.conf.d/` provides separate configuration files from vendors or third-party packages; changes to files in this directory may be overwritten by settings specified in the `/etc/X11/xorg.conf` file. The `/etc/X11/xorg.conf.d/` directory stores user-specific configuration. Files with the suffix `.conf` in configuration directories are parsed by the X server upon startup and are treated like part of the traditional `xorg.conf` configuration file. These files may contain one or more sections; for a description of the options in a section and the general layout of the configuration file, refer to Section B.3.3, “The `xorg.conf` File” or to the `xorg.conf(5)` man page. The X server essentially treats the collection of configuration files as one big file with entries from `xorg.conf` at the end. Users are encouraged to put custom configuration into `/etc/xorg.conf` and leave the directory for configuration snippets provided by the distribution.

B.3.3. The `xorg.conf` File

In previous releases of the X Window System, `/etc/X11/xorg.conf` file was used to store initial setup for X. When a change occurred with the monitor, video card or other device managed by the X server, the file needed to be edited manually. In Fedora, there is rarely a need to manually create and edit the `/etc/X11/xorg.conf` file. Nevertheless, it is still useful to understand various sections and optional parameters available, especially when troubleshooting or setting up unusual hardware configuration.

In the following, some important sections are described 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(5)` man page. This section is mostly intended for advanced users as most configuration options described below are not needed in typical configuration scenarios.

B.3.3.1. The `InputClass` section

`InputClass` is a new type of configuration section that does not apply to a single device but rather to a class of devices, including hot-plugged devices. An `InputClass` section's scope is limited by the matches specified; in order to apply to an input device, all matches must apply to the device as seen in the example below:

```plaintext
Section "InputClass"
```
The \texttt{xorg.conf} File

If this snippet is present in an \texttt{xorg.conf} file or an \texttt{xorg.conf.d} directory, any touchpad present in the system is assigned the \texttt{synaptics} driver.

\begin{Verbatim}
Identifier      "touchpad catchall"
MatchIsTouchpad "on"
Driver           "synaptics"
\end{Verbatim}

\textbf{Alphanumeric sorting in \texttt{xorg.conf.d}}

Note that due to alphanumeric sorting of configuration files in the \texttt{xorg.conf.d} directory, the \texttt{Driver} setting in the example above overwrites previously set driver options. The more generic the class, the earlier it should be listed.

The match options specify which devices a section may apply to. To match a device, all match options must correspond. The following options are commonly used in the \texttt{InputClass} section:

- \texttt{MatchIsPointer}, \texttt{MatchIsKeyboard}, \texttt{MatchIsTouchpad}, \texttt{MatchIsTouchscreen}, \texttt{MatchIsJoystick} — boolean options to specify a type of a device.
- \texttt{MatchProduct "product_name"} — this option matches if the \texttt{product_name} substring occurs in the product name of the device.
- \texttt{MatchVendor "vendor_name"} — this option matches if the \texttt{vendor_name} substring occurs in the vendor name of the device.
- \texttt{MatchDevicePath "/path/to/device"} — this option matches any device if its device path corresponds to the patterns given in the \texttt{"/path/to/device"} template, for example \texttt{/dev/input/event*}. Refer to the \texttt{fnmatch(3)} man page for further details.
- \texttt{MatchTag "tag_pattern"} — this option matches if at least one tag assigned by the HAL configuration back end matches the \texttt{tag_pattern} pattern.

A configuration file may have multiple \texttt{InputClass} sections. These sections are optional and are used to configure a class of input devices as they are automatically added. An input device can match more than one \texttt{InputClass} section. When arranging these sections, it is recommended to put generic matches above specific ones because each input class can override settings from a previous one if an overlap occurs.

\subsection*{B.3.3.2. The \texttt{InputDevice} section}

Each \texttt{InputDevice} section configures one input device for the X server. Previously, systems typically had at least one \texttt{InputDevice} section for the keyboard, and most mouse settings were automatically detected.

With Fedora 20, no \texttt{InputDevice} configuration is needed for most setups, and the \texttt{xorg-x11-drivers*} input driver packages provide the automatic configuration through HAL. The default driver for both keyboards and mice is \texttt{evdev}.

The following example shows a typical \texttt{InputDevice} section for a keyboard:

\begin{Verbatim}
Section "InputDevice"
\end{Verbatim}
Appendix B. The X Window System

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. If the **AutoAddDevices** option is enabled (which is the default setting), any input device section with **Driver "mouse"** or **Driver "kbd"** will be ignored. This is necessary due to conflicts between the legacy mouse and keyboard drivers and the new **evdev** generic driver. Instead, the server will use the information from the back end for any input devices. Any custom input device configuration in the **xorg.conf** should be moved to the back end. In most cases, the back end will be HAL and the configuration location will be the **/etc/X11/xorg.conf.d** directory.
- **Option** — Specifies necessary options pertaining to the device.

A mouse may also be specified to override any auto-detected values for the device. The following options are typically included when adding a mouse in the **xorg.conf** file:

- **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(5)** man page for a complete list of valid options for this section.

### B.3.3.3. The **ServerFlags** section

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 B.3.3.4, "ServerLayout"** for details).

Each entry within the **ServerFlags** section occupies a single 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.
X keyboard extension

Even if this option is enabled, the key combination still must be configured in the X Keyboard Extension (XKB) map before it can be used. One way how to add the key combination to the map is to run the following command:

```
setxkbmap -option "terminate:ctrl_alt_bksp"
```

- "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.

- "AutoAddDevices" "boolean" — When the value of <boolean> is set to false, the server will not hot plug input devices and instead rely solely on devices configured in the xorg.conf file. Refer to Section B.3.3.2, "The InputDevice section" for more information concerning input devices. This option is enabled by default and HAL (hardware abstraction layer) is used as a back end for device discovery.

B.3.3.4. ServerLayout

The ServerLayout section binds together the input and output devices controlled by the X server. At a minimum, this section must specify one input device and one output device. By default, a monitor (output device) and a keyboard (input device) are specified.

The following example shows 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 B.3.3.8, "The Screen section".
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. If the `AutoAddDevices` option is enabled, this entry needs not to be specified in the `ServerLayout` section. If the `AutoAddDevices` option is disabled, both mouse and keyboard are auto-detected with the default values.

- **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(5)` 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; as in the `Xorg -layout <layoutname>` command.

### B.3.3.5. The Files section

The `Files` section sets paths for services vital to the X server, such as the font path. This is an optional section, as these paths are normally detected automatically. This section can be used to override automatically detected values.

The following example shows a typical `Files` section:

```plaintext
Section "Files"
  RgbPath "/usr/share/X11/rgb.txt"
  FontPath "unix:/7100"
EndSection
```

The following entries are commonly used in the `Files` section:

- **ModulePath** — An optional parameter which specifies alternate directories which store X server modules.

### B.3.3.6. The Monitor section

Each `Monitor` section configures one type of monitor used by the system. This is an optional entry as most monitors are now detected automatically.

This example shows a typical `Monitor` section for a monitor:

```plaintext
Section "Monitor"
  Identifier "Monitor0"
  VendorName "Monitor Vendor"
  ModelName "DDC Probed Monitor - ViewSonic G773-2"
  DisplaySize 320 240
EndSection
```
The following entries are commonly 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(5)** 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(5)** man page.

### B.3.3.7. The Device section

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 following example shows 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 is required.

• **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(5)` 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.

### B.3.3.8. The Screen section

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 shows a typical **Screen** section:

```plaintext
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 detected automatically.
• **DefaultDepth** — Specifies the default color depth in bits. In the previous example, 16 (which provides thousands of colors) is the default. Only one **DefaultDepth** entry 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 detected automatically.

  This subsection is normally used to override auto-detected 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(5)` man page.

### B.3.3.9. The DRI section

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 is rarely used, as the DRI Group and Mode are automatically initialized to default values. If a different Group or Mode is needed, then adding this section to the `xorg.conf` file will override the default values.

The following example shows 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.freedesktop.org/wiki/](http://dri.freedesktop.org/wiki/).

### B.4. Fonts

Fedora uses **Fontconfig** subsystem to manage and display fonts under the X Window System. It 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 toolkits, or their newer versions.

The Fontconfig font subsystem allows applications to directly access fonts on the system and use the **X FreeType Interface library (Xft)** or other rendering mechanisms to render Fontconfig fonts with advanced features such as anti-aliasing. Graphical applications can use the Xft library with Fontconfig to draw text to the screen.
Font configuration

Fontconfig uses the /etc/fonts/fonts.conf configuration file, which should not be edited by hand.

Fonts group

Any system where the user expects to run remote X applications needs to have the fonts group installed. This can be done by selecting the group in the installer, and also by running the yum groupinstall fonts command after installation.

B.4.1. Adding Fonts to Fontconfig

Adding new fonts to the Fontconfig subsystem is a straightforward process:

1. To add fonts for an individual user, copy the new fonts into the .fonts/ directory in the user's home directory. 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.

2. Run the fc-cache command as root to update the font information cache:

   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/).

Interactive font installation

Individual users may also install fonts interactively, by typing fonts:/// into the Nautilus address bar, and dragging the new font files there.

B.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.

The following subsections review how X starts up in both runlevel 3 and runlevel 5.
B.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 B.5.2, “Runlevel 5” for more information about display managers.

1. 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.

2. 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.

3. After setting the above 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.

4. 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, then KDE, followed by `twm`.

When in runlevel 3, the user is returned to a text mode user session after ending an X session.

B.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.

- **GDM** (GNOME Display Manager) — The default display manager for Fedora. GNOME allows the user to configure language settings, shutdown, restart or log in to the system.

- **KDM** — KDE’s display manager which allows the user to shutdown, restart or log in to the system.

- **xdm** (X Window Display Manager) — A very basic display manager which only lets the user log in to the system.

When booting into runlevel 5, the `/etc/X11/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 use when they authenticate using the GNOME or KDE display managers by selecting it from the Sessions menu item accessed by selecting System → 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, or the xdm man page.

### B.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.

#### B.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.

- `/usr/share/doc/gdm-<version-number>/README` — contains information on how display managers control user authentication.

- `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.

#### B.6.2. Useful Websites

- `http://www.X.org/` — Home page of the X.Org Foundation, which produces major releases of the X Window System 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/ — Home of the GNOME project.

• http://www.kde.org/ — Home of the KDE desktop environment.

\footnote{1} http://www.gnome.org
\footnote{2} http://www.kde.org
Appendix C. Revision History

Revision 1-1       Thu Aug 9 2012       Jaromír Hradílek jhradilek@redhat.com
Updated Network Interfaces.

Revision 1-0       Tue May 29 2012      Jaromír Hradílek jhradilek@redhat.com
Fedora 17 release of the System Administrator's Guide.
Index

Symbols

.fetchmailrc, 213
.server options, 214
.user options, 215
.htaccess, 165, 169
(see also Apache HTTP Server)
.htpasswd, 165
(see also Apache HTTP Server)
.procmailrc, 217
/dev/oprofile/, 432
/dev/shm, 351
/run, 351
/sys/fs/cgroup, 351
/var/spool/anacron, 389
/var/spool/cron, 391

(see OProfile)

A
Access Control
configuring in SSSD, 129
SSSD rules, 129
adding
.group, 36
.user, 33
.anacron, 389
.anacron configuration file, 389
.user-defined tasks, 389
.anacrontab, 389
Apache HTTP Server
additional resources
.installed documentation, 200
.useful websites, 200
checking configuration, 162
checking status, 161
directives
<Directory>, 162
<IfDefine>, 162
<IfModule>, 163
<Location>, 163
<Proxy>, 164
<VirtualHost>, 164
.AccessFileName, 164
.Action, 165
.AddDescription, 165
.AddEncoding, 166
.AddHandler, 166
.AddIcon, 166
.AddIconByEncoding, 167
.AddIconByType, 167
.AddLanguage, 167
.AddType, 168
.Alias, 168
.Allow, 168
.AllowOverride, 169
.BrowserMatch, 169
.CacheDefaultExpire, 170
.CacheDisable, 170
.CacheEnable, 170
.CacheLastModifiedFactor, 171
.CacheMaxExpire, 171
.CacheNegotiatedDocs, 171
.CacheRoot, 171
.CustomLog, 172
.DefaultIcon, 172
.DefaultType, 172
.Deny, 173
.DirectoryIndex, 173
.DocumentRoot, 173
.ErrorDocument, 173
.ErrorLog, 174
.ExtendedStatus, 174
.Group, 174
.HeaderName, 175
.HostnameLookups, 175
.Include, 176
.IndexIgnore, 176
.IndexOptions, 176
.KeepAlive, 177
.KeepAliveTimeout, 178
.LanguagePriority, 178
.Listen, 178
.LoadModule, 179
.LogFormat, 179
.LogLevel, 180
.MaxClients, 190, 190
.MaxKeepAliveRequests, 180
.MaxSpareServers, 190
.MaxSpareThreads, 191
.MinSpareServers, 191
.MinSpareThreads, 191
.NameVirtualHost, 181
.Options, 181
.Order, 182
.PidFile, 182
.ProxyRequests, 182
.ReadmeName, 183
.Redirect, 183
.ScriptAlias, 184
.ServerAdmin, 184
.ServerName, 184
.ServerRoot, 185
.ServerSignature, 185
.ServerTokens, 186
.SetEnvIf, 189

511
D

date, 21, 22
   (see also Date and Time)

Date and Time
   system date, 21
   system time, 21
   time zone settings, 21

deleting cache files
   in SSSD, 132

desktop environments (see X)

df, 350

directory server (see OpenLDAP)

display managers (see X)

documentation
   finding installed, 489

drivers (see kernel module)

DSA keys
   generating, 148

du, 351

E

e-mail
   additional resources, 225
   installed documentation, 225
   related books, 227
   useful websites, 226

Fetchmail, 212

history of, 201

mail server
   Dovecot, 203
   Postfix, 205
   Procmail, 217

program classifications, 204

protocols, 201
   IMAP, 202
   POP, 202
   SMTP, 201

security, 223
   clients, 224
   servers, 224

Sendmail, 207

spam
   filtering out, 222

types
   Mail Delivery Agent, 205
   Mail Transport Agent, 204
   Mail User Agent, 205

expiration of password, forcing, 37

F

Fedora installation media
   installable packages, 481

feedback

contact information for this manual, xx

Fetchmail, 212
   additional resources, 225
   command options, 215
   informational, 216
   special, 216
   configuration options, 213
   global options, 214
   server options, 214
   user options, 215

file systems, 347

findmnt, 349

findsmb, 246

findsmb program, 260

free, 344

FTP, 267
   (see also vsftpd)
   active mode, 267
   command port, 267
   data port, 267
   definition of, 267
   introducing, 267
   passive mode, 267
   server software
      Red Hat Content Accelerator, 268
      vsftpd, 268

G

GNOME, 494
   (see also X)

gnome-system-log (see Log File Viewer)

gnome-system-monitor, 343, 345, 346, 352

GnuPG
   checking RPM package signatures, 488

GPG, 25
   group configuration
      adding groups, 31
      filtering list of groups, 29
      groupadd, 36
      modifying users in groups, 33
      modifying group properties, 32
      viewing list of groups, 29

groups (see group configuration)
   additional resources, 40
      installed documentation, 40

GID, 25

GRUB 2 boot loader
   configuration file, 445
Index

configuring, 445
GUI, 3

H
hardware
viewing, 353
HTTP server (see Apache HTTP Server)
httpd (see Apache HTTP Server)

I
information
about your system, 341
initial RAM disk image
verifying, 442
IBM eServer System i, 444
initial RPM repositories
installable packages, 481
insmod , 453
(see also kernel module)
installing package groups
installing package groups with PackageKit, 80
installing the kernel, 439

K
KDE, 494
(see also X)
kdump
additional resources
installed documents, 478
manual pages, 478
websites, 478
analyzing the dump (see crash)
configuring the service
default action, 470, 472
dump image compression, 470, 472
filtering level, 469, 472
initial RAM disk, 470, 470
kernel image, 470, 470
kernel options, 470, 470
memory usage, 466, 467, 470
supported targets, 468, 472
target location, 468, 471
enabling the service, 466, 467, 473
installing, 465
running the service, 473
system requirements, 465
testing the configuration, 473
kernel
downloading, 441
installing kernel packages, 439, 439
kernel packages, 439
package, 439
performing kernel upgrade, 441
RPM package, 439
upgrade kernel available, 441
Security Advisories, 441
via Fedora Update System, 441
upgrading
preparing, 440
working boot media, 440
upgrading the kernel, 439
Kernel Dump Configuration (see kdump)
kernel module
bonding module, 456
description, 456
parameters to bonded interfaces, 457
definition, 449
directories
/etc/sysconfig/modules/, 455
/lib/modules/kernel_version/kernel/drivers/, 452
Ethernet module
supporting multiple cards, 456
files
/proc/modules/, 450
listing
currently loaded modules, 449
module information, 450
loading
at the boot time, 455
for the current session, 452
module parameters
bonding module parameters, 457
supplying, 454
unloading, 453
utilities
insmod , 453
lsmod , 449
modinfo , 450
modprobe , 452, 453
rmmod , 454
kernel package
kernel
for single, multicore and multiprocessor systems, 439
kernel-devel
kernel headers and makefiles, 439
kernel-doc
documentation files, 439
kernel-headers
C header files files, 439
linux-firmware
firmware files, 439
perf
firmware files, 439
kernel upgrading
preparing, 440
keyboard configuration (see Region and Language)
   kwin , 494
   (see also X)

   L
language configuration (see Region and Language)
LDAP (see OpenLDAP)
Log File Viewer
   filtering, 383
   monitoring, 386
   searching, 383
log files, 369
   (see also Log Viewer )
additional resources
   installed documentation, 387
   useful websites, 387
description, 369
locating, 381
monitoring, 386
rotating, 381
rsyslogd daemon , 369
viewing, 383
Log Viewer
   refresh rate, 384
logrotate , 381
lsblk, 347
lscpu, 355
lsmod , 449
   (see also kernel module)
lspci, 353
lspcmcia, 354
lsusb, 353

   M
Mail Delivery Agent (see email)
Mail Transport Agent (see email) (see MTA)
Mail Transport Agent Switcher , 216
Mail User Agent, 216 (see email)
MDA (see Mail Delivery Agent)
memory usage, 344
metacity , 494
   (see also X)
modinfo , 450
   (see also kernel module)
modprobe , 452, 453
   (see also kernel module)
module (see kernel module)
module parameters (see kernel module)
MTA (see Mail Transport Agent)
   setting default, 216
   switching with Mail Transport Agent Switcher , 216
MUA, 216 (see Mail User Agent)
mwm , 494
   (see also X)

   N
net program, 261
Network Time Protocol, 22
NIC
   binding into single channel, 456
nmblookup program, 261
NSCD
   and SSSD, 135
NTP (see Network Time Protocol)

   O
opannotate (see OProfile)
opcontrol (see OProfile)
OpenLDAP
   checking status, 239
   client applications, 233
   configuration
database, 237
   global, 234
   overview, 231
directives
   olcAllows, 234
   olcConnMaxPending, 235
   olcConnMaxPendingAuth, 235
   olcDisallows, 235
   olcIdleTimeout, 236
   olcLogFile, 236
   olcReadOnly, 237
   olcReferral, 236
   olcRootDN, 237
   olcRootPW, 237
   olcSuffix, 238
   olcWriteTimeout, 236
directories
   /etc/openldap/slapd.d/, 234
   /etc/openldap/slapd.d/cn=config/
   /etc/openldap/slapd.d/cn=config/
   /etc/openldap/ldap.conf, 234
   /etc/openldap/slapd.d/cn=config.ldif, 234
   directories
   /etc/openldap/slapd.d/cn=config/
   /etc/openldap/slapd.d/cn=config/
   /etc/openldap/slapd.d/cn=config/
   olcDatabase=(1)bdb.ldif, 237
   installation, 231
   files
   migrating authentication information, 239
   packages, 231
   restarting, 239
running, 238
schema, 238
stopping, 238
terminology
  attribute, 230
  entry, 230
  LDIF, 230
utilities, 232, 233
OpenSSH, 141, 142
  (see also SSH)
additional resources, 155
client, 150
  scp, 152
  sftp, 152
  ssh, 151
DSA keys
  generating, 148
RSA keys
  generating, 148
RSA Version 1 keys
  generating, 149
server, 146
  starting, 146
  stopping, 146
ssh-add, 150
ssh-agent, 150
ssh-keygen
  DSA, 148
  RSA, 148
  RSA Version 1, 149
using key-based authentication, 147
OpenSSL
  additional resources, 155
SSL (see SSL)
TLS (see TLS)
ohelp, 424
opreport (see OProfile)
OProfile, 421
  /dev/oprofile/, 432
  additional resources, 436
  configuring, 422
  separating profiles, 425
events
  sampling rate, 424
  setting, 423
Java, 432
monitoring the kernel, 422
opannotate, 431
opcontrol, 422
  --no-vmlinux, 422
  --start, 426
  --vmlinux=, 422
ophelp, 424
opreport, 428, 430
on a single executable, 429
oprofiled, 426
  log file, 426
overview of tools, 421
reading data, 427
saving data, 427
starting, 426
SystemTap, 435
  unit mask, 425
oprofiled (see OProfile)
oprof_start, 433
OS/400 boot loader
  configuration file, 446
  configuring, 446
P
package
  kernel RPM, 439
PackageKit, 73
  adding and removing, 75
  architecture, 82
  installing and removing package groups, 80
  installing packages, 73
  managing packages, 73
PolicyKit
  authentication, 74
  uninstalling packages, 73
  updating packages, 73
  viewing packages, 73
  viewing transaction log, 81
packages
  adding and removing with PackageKit, 75
  dependencies, 483
  determining file ownership with, 489
  displaying packages
    yum info, 49
  displaying packages with Yum
    yum info, 49
  filtering with PackageKit, 76
    Development, 77
    Free, 77
    Hide Subpackages, 77
    Installed, 76
    No Filter, 76
    Only Available, 76
    Only Development, 77
    Only End User Files, 77
    Only Graphical, 77
    Only Installed, 76
    Only Native Packages, 78
    Only Newest Packages, 78
  finding deleted files from, 489
  finding RPM packages, 481
iFedora installation media, 481
initial RPM repositories, 481
installing a package group with Yum, 51
installing and removing package groups, 80
installing packages with PackageKit, 73, 78
dependencies, 79
installing RPM, 481
installing with Yum, 50
kernel
for single, multicore and multiprocessor
systems, 439
kernel-devel
kernel headers and makefiles, 439
kernel-doc
documentation files, 439
kernel-headers
C header files files, 439
linux-firmware
firmware files, 439
listing packages with Yum
Glob expressions, 47
yum grouplist, 48
yum list all, 47
yum list available, 48
yum list installed, 48
yum repolist, 49
yum search, 46
locating documentation for, 489
managing packages with PackageKit, 73
obtaining list of files, 490
packages and package groups, 46
perf
firmware files, 439
querying uninstallated, 490
removing, 484
removing package groups with Yum, 53
removing packages with PackageKit, 78
RPM, 479
already installed, 482
configuration file changes, 484
conflict, 483
failed dependencies, 483
freshening, 485
pristine sources, 480
querying, 486
removing, 484
source and binary packages, 479
tips, 489
uninstalling, 484
verifying, 487
searching for packages with Yum
yum search, 46
searching packages with Yum
yum search, 46
setting packages with PackageKit
checking interval, 74
uninstalling packages with PackageKit, 73
uninstalling packages with Yum, 52
yum remove package_name, 52
updating currently installed packages
available updates, 73
updating packages with PackageKit, 73
PolicyKit, 74
Software Update, 73
upgrading RPM, 481
viewing packages with PackageKit, 73
viewing transaction log, 81
viewing Yum repositories with PackageKit, 75
Yum instead of RPM, 479
partx, 349
password
aging, 37
expire, 37
Password, 101
passwords
shadow, 25
pdbedit program, 262
PolicyKit, 74
Postfix, 205
default installation, 206
postfix, 216
prefdm (see X)
Printer Configuration
CUPS, 280
IPP Printers, 283
LDP/LPR Printers, 284
Local Printers, 281
New Printer, 280
Print Jobs, 296
Samba Printers, 285
Settings, 291
Sharing Printers, 292
printers (see Printer Configuration)
processes, 341
Procmail, 217
additional resources, 225
configuration, 217
recipes, 218
delivering, 219
eamples, 221
flags, 219
local lockfiles, 220
non-delivering, 219
SpamAssassin, 222
special actions, 220
special conditions, 220
ps, 341
Index

R

RAM, 344
rcp, 152
Region and Language
date, time, and numeric format configuration, 16
keyboard configuration, 18
language configuration, 15
system-wide settings, 20
removing package groups
removing package groups with PackageKit, 80
rmmod, 454
(see also kernel module)
rpcclient program, 263
RPM, 479
additional resources, 491
already installed, 482
basic modes, 480
book about, 491
checking package signatures, 488
configuration file changes, 484
conf.rpmsave, 484
conflicts, 483
dependencies, 483
design goals, 480
powerful querying, 480
system verification, 480
upgradability, 480
determining file ownership with, 489
documentation with, 489
failed dependencies, 483
file conflicts
resolving, 483
file name, 481
finding deleted files with, 489
finding RPM packages, 481
freshening, 485
GnuPG, 488
installing, 481
md5sum, 488
querying, 486
querying for file list, 490
querying uninstallable packages, 490
tips, 489
uninstalling, 484
upgrading, 481
verifying, 487
website, 491
RPM Package Manager (see RPM)
RSA keys
generating, 148
RSA Version 1 keys
generating, 149
rsyslog, 369
S
Samba (see Samba)
Abilities, 243
Account Information Databases, 258
ldapsam, 258
ldapsam_compatible, 258
mysqlsam, 258
Plain Text, 258
smbpasswd, 258
tdbsam, 258
xmlsam, 258
Additional Resources, 265
installed documentation, 265
related books, 266
useful websites, 266
Backward Compatible Database Back Ends, 258
Browsing, 259
collection, 247, 247
default, 247
CUPS Printing Support, 260
CUPS smb.conf, 260
daemon, 244
nmbd, 244
overview, 244
smbd, 244
winbind, 244
encrypted passwords, 248
findsmb, 246
graphical configuration, 247
Introduction, 243
Network Browsing, 259
Domain Browsing, 259
WINS, 259
New Database Back Ends, 258
Programs, 260
findsmb, 260
net, 261
nnmlookup, 261
pdbedit, 262
rpcclient, 263
smbcacls, 263
smbclient, 263
smbcontrol, 263
smbpasswd, 263
smbextent, 264
smbstatus, 264
smbnt, 264
testparm, 264
wbinfo, 265
Reference, 243
Samba Printers, 285
Security Modes, 256
  Active Directory Security Mode, 257
  Domain Security Mode, 256
  Server Security Mode, 257
  Share-Level Security, 257
  User Level Security, 256
Server Types, 249
server types
  Domain Controller, 254
  Domain Member, 251
  Stand Alone, 249
service
  conditional restarting, 248
  reloading, 248
  restarting, 248
  starting, 248
  stopping, 248
share
  connecting to via the command line, 246
  connecting to with Nautilus, 245
  mounting, 246
smb.conf, 249
  Active Directory Member Server example, 252
  Anonymous Print Server example, 250
  Anonymous Read Only example, 249
  Anonymous Read/Write example, 250
  NT4-style Domain Member example, 253
  PDC using Active Directory, 256
  PDC using tdbssam, 254
  Secure File and Print Server example, 251
smbclient, 246
  WINS, 259
  with Windows NT 4.0, 2000, ME, and XP, 248
scp (see OpenSSH)
security plug-in (see Security)
Security-Related Packages
  updating security-related packages, 46
Sendmail, 207
  additional resources, 225
  aliases, 210
  common configuration changes, 209
  default installation, 208
  LDAP and, 212
  limitations, 208
  masquerading, 210
  purpose, 208
  spam, 211
  with UUCP, 209
sendmail, 216
  services configuration, 87
  ssystemctl, 88
  systemctl, 87
sftp (see OpenSSH)
shadow passwords
  overview of, 25
sladp (see OpenLDAP)
smbcacls program, 263
smbclient, 246
smbcontrol program, 263
smbpasswd program, 263
smbstatus program, 264
smbspool program, 264
smbtar program, 264
SpamAssassin
  using with Procmail, 222
ssh (see OpenSSH)
SSH protocol
  authentication, 144
  configuration files, 144
  system-wide configuration files, 145
  user-specific configuration files, 145
  connection sequence, 142
  features, 142
  insecure protocols, 147
layers
  channels, 144
  transport layer, 143
  port forwarding, 154
  requiring for remote login, 147
security risks, 141
version 1, 142
version 2, 142
X11 forwarding, 153
ssh-add, 150
ssh-agent, 150
SSL, 193
  (see also Apache HTTP Server)
SSL server (see Apache HTTP Server)
SSSD
  and NSCD, 135
  Kerberos authentication, 124
  LDAP domain, 117
  supported LDAP directories, 117
  Microsoft Active Directory Domain, 121
  proxy domain, 127
startx, 505 (see X)
  (see also X)
stunnel, 224
system analysis
  OProfile (see OProfile)
system information
  cpu usage, 346
  file systems, 347
  /dev/shm, 351
  /run, 351
  /sys/fs/cgroup, 351
gathering, 341
hardware, 353
memory usage, 344
processes, 341
currently running, 342
System Monitor, 343, 345, 346, 352
system-config-authentication (see Authentication Configuration Tool)
system-config-kdump (see kdump)
system-config-users (see user configuration and group configuration)
systemctl (see services configuration)

T
testparm program, 264
time, 21, 22
(time also Date and Time)
time zone, 21
(time also Date and Time)
TLS, 193
(see also Apache HTTP Server)
tool
Authentication Configuration Tool, 93
top, 342
twm, 494
(see also X)

U
updating currently installed packages
available updates, 73
updating packages with PackageKit
PolicyKit, 73, 74
User Accounts (see user configuration)
user configuration
adding users, 28, 30
changing full name, 32
changing home directory, 32
changing login shell, 32
changing password, 32
command line configuration
chage, 37
passwd, 34
useradd, 33
filtering list of users, 29
modify groups for a user, 31
modifying users, 31
password
forcing expiration of, 37
removing users, 28
viewing list of users, 26, 29
User Manager (see user configuration)
user private groups (see groups)
and shared directories, 39
useradd command
user account creation using, 33
users (see user configuration)
additional resources, 40
installed documentation, 40
introducing, 25
tools for management of
User Manager, 33
useradd, 33
UID, 25

V
virtual host (see Apache HTTP Server)
vsftpd, 268
(see also FTP)
additional resources, 279
installed documentation, 279
useful websites, 279
condrestart, 269
configuration file
/etc/vsftpd/vsftpd.conf, 270
access controls, 271
anonymous user options, 272
daemon options, 271
directory options, 274
file transfer options, 275
format of, 270
local user options, 273
logging options, 275
login options, 271
network options, 277
multihome configuration, 270
restarting, 269
RPM
files installed by, 268
security features, 268
starting, 269
starting multiple copies of, 270
status, 269
stopping, 269

W
wbinfo program, 265
web server (see Apache HTTP Server)
window managers (see X)
Windows 2000
connecting to shares using Samba, 248
Windows 98
connecting to shares using Samba, 248
Windows ME
connecting to shares using Samba, 248
Windows NT 4.0
connecting to shares using Samba, 248
Windows XP
connecting to shares using Samba, 248

X
/ etc / X11 / xorg . conf
  boolean values for, 495
  Device, 501
  DRI, 503
  Files section, 500
  InputDevice section, 497
  introducing, 496, 496
  Monitor, 500
  Screen, 502
  Section tag, 495
  ServerFlags section, 498
  ServerLayout section, 499
  structure of, 495
additional resources, 506
  installed documentation, 506
useful websites, 506
configuration directory
  / etc / X11 / xorg . conf . d, 496
configuration files
  / etc / X11 / directory, 495
  / etc / X11 / xorg . conf, 496
options within, 495
  server options, 496, 496
desktop environments
  GNOME, 494
  KDE, 494
display managers
  configuration of preferred, 505
  definition of, 505
  GNOME, 505
  KDE, 505
  prefdm script, 505
  xdm, 505
fonts
  Fontconfig, 503
  Fontconfig, adding fonts to, 504
  FreeType, 503
  introducing, 503
  Xft, 503
  introducing, 493
runlevels
  3, 505
  5, 505
runlevels and, 504
window managers
  kwin, 494
  metacity, 494
  mwm, 494
  twm, 494
  X clients, 493, 493
  desktop environments, 494
  startx command, 505
  window managers, 494
  xinit command, 505
  X server, 493
  features of, 493
  X Window System (see X)
  X.500 (see OpenLDAP)
  X.500 Lite (see OpenLDAP)
  xinit (see X)
  Xorg (see Xorg)

Y
Yum
  Additional Resources, 71
  configuring plug-ins, 66
  configuring Yum and Yum repositories, 58
  disabling plug-ins, 66
  displaying packages
    yum info, 49
    displaying packages with Yum
      yum info, 49
      enabling plug-ins, 66
      installing a package group with Yum, 51
      installing with Yum, 50
      listing packages with Yum
      Glob expressions, 47
      yum grouplist, 48
      yum list, 46
      yum list all, 47
      yum list available, 48
      yum list installed, 48
      yum repolist, 49
      packages and package groups, 46
      plug-ins
        fs-snapshot, 67
        presto, 70
        refresh-packagekit, 70
        rhnplugin, 70
        security, 70
      repository, 64, 65
      searching for packages with Yum
        yum search, 46
      searching packages with Yum
        yum search, 46
      setting [main] options, 58
      setting [repository] options, 61
      uninstalling package groups with Yum, 53
      uninstalling packages with Yum, 52
        yum remove package_name, 52
      variables, 62
      Yum plug-ins, 65
      Yum repositories
configuring Yum and Yum repositories, 58

Yum repositories
viewing Yum repositories with PackageKit, 75

Yum Updates
checking for updates, 43
updating a single package, 44
updating all packages and dependencies, 45
updating packages, 44
updating security-related packages, 46